ComSIS

Computer Science and Information Systems

Volume 23, Number 1, January 2026

Contents

Editorial
Guest Editorial: Advances in Databases, Information Systems, and Database Engineered Applications – Recent Trends, and Open Issues
Guest Editorial: Advances on Information Communication Technologies and Intelligent (Systems and their Applications)

Papers

1 Optimizing Cell-Based Software Architecture through Heuristic Community Detection Approach
Miloš Milič, Nebojša Nikolić, Dragana Makajić-Nikolić
33 Comparing Novice Programmers Performance with Block-based, Text-based, and Both Notations: A Study from Two Countries
Tomaž Kosar, Srđa Bjeladinović, Dragana Ostojić, Milica S. Škembarević, Žiga Leber, Olga A. Jejić, Matej Moravec, Filip Furtula, Miloš D. Ljubisavljević, Ivan S. Luković, Marjan Mernik, Matej Črepinšek
61 Review of Architectures in IoT Remote Patient Monitoring Systems
Lan Sovinc, Marko Bajec
75 DomiCom: Discovery of Top-k Dominant Communities in Networks with Node Attributes
Nikolaos Georgiadis, Eleftherios Tiakas, Apostolos N. Papadopoulos
113 ASD-RRT*: An Enhanced Path Planning Algorithm Based on RRT* for Multi-Obstacle Environments
Chao Wang, Wenbin Li
133 ADN-YOLO: An Improved Ship Detection Model Based on YOLOv11
Tao Li, Dezhi Han, Songyang Wu, Xiang Shen, Liqi Zhu, Wenqi Sun
165 A method for representing stock time series features based on trend and inclination angle turning points
Lei Han, Xuedong Gao
185 An ANN-WSM Hybrid Framework for Sustainable Personalized Learning in Smart Education
Christos Troussas, Akrivi Krouska, Phivos Mylonas, Cleo Sgouropoulou, Ioannis Voyiatzis
207 AI-Powered Virtual Assistants in Education: A Systematic Review of Applications and Implications (2014–2025)
Eleni Papachristou, Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou
231 Meeting Consumer Demand: Supporting Efficient Services Deployment for Cloud-Connected Devices
Iván Bernabé-Sánchez, Daniel Díaz-Sánchez, Alberto Fernández, Holger Billhardt, Sascha Ossowski
255 Weld proximity defect detection model for steel thin plates based on EP-YOLOv7
Runmei Zhang, Jingwei Fan, Zihua Chen, Zhong Chen, Bin Yuan
277 Adaptive Bandwidth Allocation via Uncertainty-Constrained Deep Reinforcement Learning
Li Wei, Wu Yong, Yan Dong
299 Robust QoS-Aware Network Scheduling for Smart Substations via Multi-Agent Adversarial Reinforcement Learning
Ping He, Dongsheng Jing, Baozhen Qi, Yu Yang, Jingsong Xue
321 Transformer Substation Network Disconnection Prediction via Semantic Reasoning with Causal Modeling
Jie Ren, Xiaojun Yao, Hong Chen
343 DICYME: Dynamic Industrial Cyber Risk Modelling Based on Evidence
Javier García-Ochoa, Jaime Rueda, Rubén R. Fernández, Alberto Fernández-Isabel, Isaac Martín de Diego, Emilio L. Cano, Romy R. Ravines, Ovidio López Espinosa, Jaume Puigbó Sanvisens
369 FNNMFF: Crop Pests and Diseases Detection Based on Fuzzy Neural Network and Multilevel Feature Fusion in Remote Sensing Images
Shoulin Yin, Hang Li, Mirjana Ivanović, Tao Chen, Lin Teng

Special section: Advances in Databases, Information Systems, and Database Engineered Applications – Recent Trends, and Open Issues

397 Mitigating Out-of-Vocabulary Challenges in Embedded devices Vulnerability Classification: An Ensemble Embedding Approach with Bidirectional Context Modeling
Aissa Ben Yahya, Hicham El Akhal, Abdelbaki El Belrhiti El Alaoui
419 Towards Enhancing Data Science Agents with Semantics
Sayed Hoseini, Maximilian Ibbels, Maximilian Knoll, Christoph Quix
443 Towards Explainable Sequential Learning
Giacomo Bergami, Emma Packer, Kirsty Scott, Silvia Del Din
475 Siren Federate: Bridging Document, Relational, and Graph Models for Exploratory Graph Analysis
Georgeta Bordea, Stéphane Campinas, Matteo Catena, Renaud Delbru

Special section: Advances on Information Communication Technologies and Intelligent (Systems and their Applications)

513 Using Genetic Programming as a Feature Selector and Classifier to Implement Bankruptcy Prediction Models
Ángel Beade, José Santos, Manuel Rodríguez
535 Enhancing Network Engineering Capabilities through LLM Fine-Tuning with Automatically Generated Datasets
Claudiu Trăistaru, Florin Pop, Costin Bădică, Cătălina Mancaș, Ionuț Murarețu
561 Cascade Systems as An Implementation of a Gray-Box Architecture: A Case Study in Traceability for Microservice Scaling
Jorge Jiménez García, Ignacio Lacalle Úbeda, Paweł Szmeja, Katarzyna Wasielewska-Michniewska, Przemysław Hołda, Maria Ganzha, Carlos E. Palau Salvador, Costin Bădică, Stefka Fidanova, Marcin Paprzycki
583 Audio Feature-based User Profiles for Personalized Music Recommendation: A Dataset-driven Evaluation
Ionuț-Dragoș Neremzoiu, Andreea Liliana Bădică
603 Trustless Exchange for Personal Data: Adapting Intellectual Property Trading Security Protocols for Data Sovereignty
Vijon Baraku, Simeon Veloudis, Iraklis Paraskakis, Poonam Yadav, Rezon Baraku
621 IoT based Personalized Online Learning Systems (Systematic Mapping Study)
Edlir Spaho, Betim Çiço
657 A Knowledge Graph based Approach for Credit Card Fraud Detection
George Konstantinos Dimou, Georgia Koloniari

ISSN: 2406-1018 (Online)

Computer Science and Information Systems

Vol 23, No 1, January 2026

ComSIS

Computer Science and Information Systems

Published by ComSIS Consortium

Volume 23, Number 1
January 2026

ComSIS is an international journal published by the ComSIS Consortium

ComSIS Consortium:
University of Belgrade:
Faculty of Organizational Science, Belgrade, Serbia
Faculty of Mathematics, Belgrade, Serbia
School of Electrical Engineering, Belgrade, Serbia
Serbian Academy of Science and Art:
Mathematical Institute, Belgrade, Serbia
Union University:
School of Computing, Belgrade, Serbia
University of Novi Sad:
Faculty of Sciences, Novi Sad, Serbia
Faculty of Technical Sciences, Novi Sad, Serbia
Technical Faculty "Mihajlo Pupin", Zrenjanin, Serbia
University of Niš:
Faculty of Electronic Engineering, Niš, Serbia
University of Montenegro:
Faculty of Economics, Podgorica, Montenegro

EDITORIAL BOARD:
Editor-in-Chief: Mirjana Ivanović, University of Novi Sad
Vice Editor-in-Chief: Boris Delibašić, University of Belgrade
Managing Editors:
Vladimir Kurbalija, University of Novi Sad
Miloš Radovanović, University of Novi Sad

Editorial Assistants:
Jovana Vidaković, University of Novi Sad
Ivan Pribela, University of Novi Sad
Davorka Radaković, University of Novi Sad
Slavica Kordić, University of Novi Sad
Srđan Škrbić, University of Novi Sad

Editorial Board:
A. Badica, *University of Craiova, Romania*
C. Badica, *University of Craiova, Romania*
M. Bajec, *University of Ljubljana, Slovenia*
L. Bellatreche, *ISAE-ENSMA, France*
I. Berković, *University of Novi Sad, Serbia*
D. Bojić, *University of Belgrade, Serbia*
Z. Bosnic, *University of Ljubljana, Slovenia*
D. Brđanin, *University of Banja Luka, Bosnia and Hercegovina*
R. Chbeir, *University Pau and Pays Adour, France*
M-Y. Chen, *National Cheng Kung University, Tainan, Taiwan*
C. Chesñevar, *Universidad Nacional del Sur, Bahía Blanca, Argentina*
W. Dai, *Fudan University Shanghai, China*
P. Delias, *International Hellenic University, Kavala University, Greece*
B. Delibašić, *University of Belgrade, Serbia*
G. Devedžić, *University of Kragujevac, Serbia*
J. Eder, *Alpen-Adria-Universität Klagenfurt, Austria*
Y. Fan, *Communication University of China*
V. Filipović, *University of Belgrade, Serbia*
T. Galinac Grbac, *Juraj Dobrila University of Pula, Croatia*
H. Gao, *Shanghai University, China*
M. Gušev, *Ss. Cyril and Methodius University Skopje, North Macedonia'*
D. Han, *Shanghai Maritime University, China*
M. Heričko, *University of Maribor, Slovenia*
M. Holbl, University of Maribor, Slovenia
L. Jain, *University of Canberra, Australia*
D. Janković, *University of Niš, Serbia*
J. Janousek, *Czech Technical University, Czech Republic*
G. Jezic, *University of Zagreb, Croatia*
G. Kardas, *Ege University International Computer Institute, Izmir, Turkey*
Lj. Kašćelan, *University of Montenegro, Montenegro*
P. Kefalas, *City College, Thessaloniki, Greece*
M-K. Khan, *King Saud University, Saudi Arabia*
S-W. Kim, *Hanyang University , Seoul, Korea*
M. Kirikova, *Riga Technical University, Latvia*
A. Klašnja Milićević, *University of Novi Sad, Serbia*
J. Kratica, *Institute of Mathematics SANU, Serbia*
K-C. Li, *Providence University, Taiwan*
M. Lujak, *University Rey Juan Carlos, Madrid, Spain*
JM. Machado, *School of Engineering, University of Minho, Portugal*
Z. Maamar, *Zayed University, UAE*
Y. Manolopoulos, *Aristotle University of Thessaloniki, Greece*
M. Mernik, *University of Maribor, Slovenia*
B. Milašinović, *University of Zagreb, Croatia*
A. Mishev, *Ss. Cyril and Methodius University Skopje, North Macedonia*
N. Mitić, *University of Belgrade, Serbia*
N-T. Nguyen, *Wroclaw University of Science and Technology, Poland*
P Novais, *University of Minho, Portugal*
B. Novikov, *St Petersburg University, Russia*
M. Paprzicky, *Polish Academy of Sciences, Poland*
P. Peris-Lopez, *University Carlos III of Madrid, Spain*
J. Protić, *University of Belgrade, Serbia*
M. Racković, *University of Novi Sad, Serbia*
M. Radovanović, *University of Novi Sad, Serbia*
P. Rajković, *University of Nis, Serbia*
O. Romero, *Universitat Politècnica de Catalunya, Barcelona, Spain*
C, Savaglio, *ICAR-CNR, Italy*
H. Shen, *Sun Yat-sen University, China*
J. Sierra, *Universidad Complutense de Madrid, Spain*
B. Stantic, *Griffith University, Australia*
H. Tian, *Griffith University, Australia*
N. Tomašev, *Google, London*
G. Trajčevski, *Northwestern University, Illinois, USA*
G. Velinov, *Ss. Cyril and Methodius University Skopje, North Macedonia*
L. Wang, *Nanyang Technological University, Singapore*
F. Xia, *Dalian University of Technology, China*
S. Xinogalos, *University of Macedonia, Thessaloniki, Greece*
S. Yin, *Software College, Shenyang Normal University, China*
K. Zdravkova, *Ss. Cyril and Methodius University Skopje, North Macedonia*
J. Zdravković, *Stockholm University, Sweden*

ComSIS Editorial Office:
University of Novi Sad, Faculty of Sciences,
Department of Mathematics and Informatics
Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
Phone: +381 21 458 888; **Fax:** +381 21 6350 458
www.comsis.org; Email: comsis@uns.ac.rs

Volume 23, Number 1, 2026
Novi Sad

Computer Science and Information Systems

ISSN: 2406-1018 (Online)

The ComSIS journal is sponsored by:

Ministry of Education, Science and Technological Development of the Republic of Serbia
http://www.mpn.gov.rs/



Computer Science and Information Systems

AIMS AND SCOPE

Computer Science and Information Systems (ComSIS) is an international refereed journal, published in Serbia. The objective of ComSIS is to communicate important research and development results in the areas of computer science, software engineering, and information systems.

We publish original papers of lasting value covering both theoretical foundations of computer science and commercial, industrial, or educational aspects that provide new insights into design and implementation of software and information systems. In addition to wide-scope regular issues, ComSIS also includes special issues covering specific topics in all areas of computer science and information systems.

ComSIS publishes invited and regular papers in English. Papers that pass a strict reviewing procedure are accepted for publishing. ComSIS is published semiannually.

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2024 two-year impact factor 1.8,
- Computer Science Bibliography, University of Trier (DBLP),
- EMBASE (Elsevier),
- Scopus (Elsevier),
- Summon (Serials Solutions),
- EBSCO bibliographic databases,
- IET bibliographic database Inspec,
- FIZ Karlsruhe bibliographic database io-port,
- Index of Information Systems Journals (Deakin University, Australia),
- Directory of Open Access Journals (DOAJ),
- Google Scholar,
- Journal Bibliometric Report of the Center for Evaluation in Education and Science (CEON/CEES) in cooperation with the National Library of Serbia, for the Serbian Ministry of Education and Science,
- Serbian Citation Index (SCIndeks),
- doiSerbia.

Information for Contributors

The Editors will be pleased to receive contributions from all parts of the world. An electronic version (LaTeX), or three hard-copies of the manuscript written in English, intended for publication and prepared as described in "Manuscript Requirements" (which may be downloaded from http://www.comsis.org), along with a cover letter containing the corresponding author's details should be sent to official journal e-mail.

Criteria for Acceptance

Criteria for acceptance will be appropriateness to the field of Journal, as described in the Aims and Scope, taking into account the merit of the content and presentation. The number of pages of submitted articles is limited to 20 (using the appropriate LaTeX template).

Manuscripts will be refereed in the manner customary with scientific journals before being accepted for publication.

Copyright and Use Agreement

All authors are requested to sign the "Transfer of Copyright" agreement before the paper may be published. The copyright transfer covers the exclusive rights to reproduce and distribute the paper, including reprints, photographic reproductions, microform, electronic form, or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce the paper or any part of it, for which copyright exists.

Computer Science and Information Systems

Volume 23, Number 1, January 2026

CONTENTS

Editorial

Guest Editorial: Advances in Databases, Information Systems, and Database Engineered Applications – Recent Trends, and Open Issues

Guest Editorial: Advances on Information Communication Technologies and Intelligent (Systems and their Applications)

Papers

1 **Optimizing Cell-Based Software Architecture through Heuristic Community Detection Approach**
Miloš Milić, Nebojša Nikolić, Dragana Makajić-Nikolić

33 **Comparing Novice Programmers Performance with Block-based, Text-based, and Both Notations: A Study from Two Countries**
Tomaž Kosar, Srđa Bjeladinović, Dragana Ostojić, Milica S. Škembarević, Žiga Leber, Olga A. Jejić, Matej Moravec, Filip Furtula, Miloš D. Ljubisavljević, Ivan S. Luković, Marjan Mernik, Matej Črepinšek

61 **Review of Architectures in IoT Remote Patient Monitoring Systems**
Lan Sovinc, Marko Bajec

75 **DomiCom: Discovery of Top-k Dominant Communities in Networks with Node Attributes**
Nikolaos Georgiadis, Eleftherios Tiakas, Apostolos N. Papadopoulos

113 **ASD-RRT*: An Enhanced Path Planning Algorithm Based on RRT* for Multi-Obstacle Environments**
Chao Wang, Wenbin Li

133 **ADN-YOLO: An Improved Ship Detection Model Based on YOLOv11**
Tao Li, Dezhi Han, Songyang Wu, Xiang Shen, Liqi Zhu, Wenqi Sun

165 **A method for representing stock time series features based on trend and inclination angle turning points**
Lei Han, Xuedong Gao

185 **An ANN-WSM Hybrid Framework for Sustainable Personalized Learning in Smart Education**
Christos Troussas, Akrivi Krouska, Phivos Mylonas, Cleo Sgouropoulou, Ioannis Voyiatzis

207 **AI-Powered Virtual Assistants in Education: A Systematic Review of Applications and Implications (2014–2025)**
Eleni Papachristou, Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou

231 **Meeting Consumer Demand: Supporting Efficient Services Deployment for Cloud-Connected Devices**
Iván Bernabé-Sánchez, Daniel Díaz-Sánchez, Alberto Fernández, Holger Billhardt, Sascha Ossowski

255 **Weld proximity defect detection model for steel thin plates based on EP-YOLOv7**
Runmei Zhang, Jingwei Fan, Zihua Chen, Zhong Chen, Bin Yuan

277 **Adaptive Bandwidth Allocation via Uncertainty-Constrained Deep Reinforcement Learning**
Li Wei, Wu Yong, Yan Dong

299 **Robust QoS-Aware Network Scheduling for Smart Substations via Multi-Agent Adversarial Reinforcement Learning**
Ping He, Dongsheng Jing, Baozhen Qi, Yu Yang, Jingsong Xue

321 **Transformer Substation Network Disconnection Prediction via Semantic Reasoning with Causal Modeling**
Jie Ren, Xiaojun Yao, Hong Chen

343 **DICYME: Dynamic Industrial Cyber Risk Modelling Based on Evidence**
Javier García-Ochoa, Jaime Rueda, Rubén R. Fernández, Alberto Fernández-Isabel, Isaac Martín de Diego, Emilio L. Cano, Romy R. Ravines, Ovidio López Espinosa, Jaume Puigbó Sanvisens

369 **FNNMFF: Crop Pests and Diseases Detection Based on Fuzzy Neural Network and Multilevel Feature Fusion in Remote Sensing Images**
Shoulin Yin, Hang Li, Mirjana Ivanović, Tao Chen, Lin Teng

Special section: Advances in Databases, Information Systems, and Database Engineered Applications – Recent Trends, and Open Issues

397 **Mitigating Out-of-Vocabulary Challenges in Embedded devices Vulnerability Classification: An Ensemble Embedding Approach with Bidirectional Context Modeling**
Aissa Ben Yahya, Hicham El Akhal, Abdelbaki El Belrhiti El Alaoui

419 **Towards Enhancing Data Science Agents with Semantics**
Sayed Hoseini, Maximilian Ibbels, Maximilian Knoll, Christoph Quix

443 **Towards Explainable Sequential Learning**
Giacomo Bergami, Emma Packer, Kirsty Scott, Silvia Del Din

475 **Siren Federate: Bridging Document, Relational, and Graph Models for Exploratory Graph Analysis**
Georgeta Bordea, Stéphane Campinas, Matteo Catena, Renaud Delbru

Special section: Advances on Information Communication Technologies and Intelligent (Systems and their Applications)

513 **Using Genetic Programming as a Feature Selector and Classifier to Implement Bankruptcy Prediction Models**
Ángel Beade, José Santos, Manuel Rodríguez

535 **Enhancing Network Engineering Capabilities through LLM Fine-Tuning with Automatically Generated Datasets**
Claudiu Trăistaru, Florin Pop, Costin Bădică, Cătălina Mancaş, Ionuţ Murareţu

561 **Cascade Systems as An Implementation of a Gray-Box Architecture: A Case Study in Traceability for Microservice Scaling**
Jorge Jiménez García, Ignacio Lacalle Úbeda, Paweł Szmeja, Katarzyna Wasielewska-Michniewska, Przemysław Hołda, Maria Ganzha, Carlos E. Palau Salvador, Costin Bădică, Stefka Fidanova, Marcin Paprzycki

583 **Audio Feature-based User Profiles for Personalized Music Recommendation: A Dataset-driven Evaluation**
Ionuţ-Dragoş Neremzoiu, Andreea Liliana Bădică

603 **Trustless Exchange for Personal Data: Adapting Intellectual Property Trading Security Protocols for Data Sovereignty**
Vijon Baraku, Simeon Veloudis, Iraklis Paraskakis, Poonam Yadav, Rezon Baraku

621 **IoT based Personalized Online Learning Systems (Systematic Mapping Study)**
Edlir Spaho, Betim Çiço

657 **A Knowledge Graph based Approach for Credit Card Fraud Detection**
George Konstantinos Dimou, Georgia Koloniari

Computer Science and Information Systems 23(1): i–iii https://doi.org/10.2298/CSIS260100iI

Editorial

Mirjana Ivanović, Miloš Radovanović, and Vladimir Kurbalija

University of Novi Sad, Faculty of Sciences
Novi Sad, Serbia
{mira,radacha,kurba}@dmi.uns.ac.rs

This, the first issue of Volume 23 of the Computer Science and Information Systems journal in 2026 gathers together 16 regular articles and two special sections, "Advances in Databases, Information Systems, and Database Engineered Applications – Recent Trends, and Open Issues" and "Advances on Information Communication Technologies and Intelligent Systems and their Applications", featuring 4 and 7 articles, respectively. As always, we are grateful for the efforts and enthusiasm of our authors, reviewers, and guest editors, without whom the current issue and the publication of the journal itself would not be possible.

The first regular article, "Optimizing Cell-Based Software Architecture Through Heuristic Community Detection Approach" by Miloš Milić et al., investigates cell-based software architectures, where a software system is viewed as a network of interconnected cells, each comprising multiple elements. Community detection is employed to identify closely connected elements, with the goal of optimizing the architecture by determining the optimal number of cells and their internal organization. A heuristic approach is developed to facilitate solving large-scale optimization problems.

In the second regular article, "Comparing Novice Programmers Performance with Block-Based, Text-Based, and Both Notations: A Study from Two Countries," Tomaž Kosar et al. explore different notation approaches for learning fundamental programming concepts, with the objective of assessing how notation choice influences beginners' performance. A controlled experiment during short-term visits aimed at promoting programming in primary schools indicated that the choice of notation did not significantly impact the performance of the participants, suggesting that educators can confidently use different notations (Python, Poliglot, or a combination). However, students' performance may improve when training sessions are extended over a longer period.

"Review of Architectures in IoT Remote Patient Monitoring Systems," by Lan Sovinc and Marko Bajec, identifies four different classification categories, each with their own strengths and weaknesses: fog-based, cloud-based, hybrid, and blockchain-enabled. The article summarized results from 11 original publications, describing the characteristics of the identified approaches by examining common and unique concepts used by the solutions in the same classification category based on prior analysis.

In their article entitled "DomiCom: Discovery of Top-k Dominant Communities in Networks with Node Attributes," Nikolaos Georgiadis et al. incorporate node attributes to construct more meaningful communities that reflect both structural connections and attribute information, by introducing the concept of dominance relationships between nodes. Experimental evaluation on real-world attributed networks demonstrated the efficiency and effectiveness of the proposed approach, with the reported communities shown to be meaningful and robust based on their significance and structural coherence.

Chao Wang and Wenbin Li, in "ASD-RRT*: An Enhanced Path Planning Algorithm Based on Rrt* for Multi-Obstacle Environments," propose a novel algorithm, Adaptive Sampling and Densification RRT* (ASD-RRT*), for path planning in multi-obstacle environments. The method extends RRT*-based sampling methods by incorporating adaptive sampling to enhance performance in complex environments. The adaptive sampling approach allows the algorithm to focus on effective regions, reducing sampling of irrelevant points and finding feasible solutions with fewer samples while maintaining the asymptotic optimality of RRT*.

"ADN-YOLO: An Improved Ship Detection Model Based on YOLOv," authored by Tao Li et al, proposes an improved You Only Look Once (YOLO) based model for ship target detection in infrared images named ADN-YOLO. The model introduces a Dynamic Upsampler (Dysample) to integrates semantic information across different layers, then a lightweight downsampling module (ADown) to reduce the parameter count, and finally a new loss function that combines the Normalized Wasserstein Distance (NWD) with the Complete Intersection over Union (CIoU) to enhance the model's ability to accurately localize small targets.

The article "A Method for Representing Stock Time Series Features Based on Trend and Inclination Angle Turning Points," by Lei Han, Xuedong Gao, proposes a stock time series feature representation method based on a combination of slope angle changes and trend turning point screening which alleviates the limitations of single methods, such as local feature overfitting, loss of global trends, and sparse long-segment features, through a multi-level collaborative mechanism of screening, filtering, and supplementation.

Christos Troussas et al., in their article "An ANN-WSM Hybrid Framework for Sustainable Personalized Learning in Smart Education," present an adaptive learning system integrated with ANNs and WSM that recommends particular learning activities to students based on their cognitive styles. The system classifies learners according to the Gregorc Mind Styles Model and, through the ANN, provides probabilities of predefined learning activities. Experimental evaluation involving 70 undergraduate students demonstrated significant improvements in terms of engagement, retention, and performance.

In "AI-Powered Virtual Assistants in Education: A Systematic Review of Applications and Implications (2014–2025)," Eleni Papachristou et al. examine and synthesize 62 empirical studies published between 2014 and 2025 that examine the role of AI-powered virtual assistants (VAs) in primary, secondary, and tertiary education, addressing four research questions: (1) the educational uses and implications of VAs, (2) the most frequently applied models and algorithms, (3) their contribution to learner motivation and self-determination, and (4) the benefits and challenges of instructional design supported by VAs.

In their article "Meeting Consumer Demand: Supporting Efficient Services Deployment for Cloud-Connected Devices," Iván Bernabé-Sánchez et al. present a service architecture that can be integrated with network service managers to identify priorities that should be taken into account when deploying virtualized network functions (VNFs) or services in infrastructures. The proposed solution collects information about the network service to be deployed and analyzes and evaluates its requirements by comparing them with operating patterns, obtaining recommendations for the deployment of each service, also providing a provisioning service which helps in automating the deployment.

The article "Weld Proximity Defect Detection Model for Steel Thin Plates Based on EP-YOLOv7," by Runmei Zhang et al. proposes an industrial detector for steel plate weld proximity defects based on EP-YOLOv7. The detector introduces efficient multi-scale channel attention (EMCA) to strengthen multi-scale feature perception, integrates it into the efficient layer aggregation network to enhance feature fusion and defect representation, and employs a partial-bottleneck decoupling predictor head (P-BD Head) to improve localization accuracy and reduce missed detections of small targets.

Wei Li et al., in "Adaptive Bandwidth Allocation via Uncertainty-Constrained Deep Reinforcement Learning," propose a novel algorithm named Uncertainty-Constrained Stability-aware Deep Reinforcement Learning (UCS-DRL) for dynamic network service bandwidth allocation, which adopts a dual-policy architecture: a task policy that learns optimal bandwidth allocation decisions, and a stability policy guided by uncertainty-aware value estimation to identify and mitigate potential risky or unstable behaviors during deployment in demanding multi-user, dynamic, and QoS-sensitive applications.

"Robust QoS-aware Network Scheduling for Smart Substations via Multi-Agent Adversarial Reinforcement Learning," authored by Ping He et al, proposes a Quality-of-Service-driven routing optimization approach based on adversarial reinforcement learning, referred to as Adversarial Critic-Cooperative Actor (ACCA). By introducing adversarial agents that model worst-case perturbations, ACCA establishes a multi-agent game framework that enhances policy robustness and adaptability in dynamic network environments.

In "Transformer Substation Network Disconnection Prediction via Semantic Reasoning with Causal Modeling," Jie Ren et al. propose a network disconnection prediction approach that integrates prompt-driven semantic reasoning with structured causal analysis. The approach constructs a causal event graph that models semantic, temporal, and topological dependencies across devices, after which an inference module combines causal path analysis, structural causal models, and counterfactual reasoning to assess the influence of events, predict emerging disconnection risks, and identify plausible root causes with coherent and interpretable justification.

"DICYME: Dynamic Industrial Cyber Risk Modelling Based on Evidence," authored by Javier García-Ochoa et al, introduces a comprehensive system that integrates diverse analytical techniques to identify patterns and characteristics which reveal emerging threat trends. DICYME operates as a pipeline that retrieves data from diverse cyber incident reports, specialized databases, and other relevant sources of cyber-related information, applies specialized techniques for victim identification, indicator computation, threat actor profiling, Common Vulnerability and Exposure (CVE) relationship mapping, and ultimately performs Cyber Risk Quantification (CRQ).

Finally, "FNNMFF: Crop Pests and Diseases Detection Based on Fuzzy Neural Network and Multilevel Feature Fusion in Remote Sensing Images," by Shoulin Yin et al, proposes a novel detection approach based on the YOLOv5 model and extracts the semantic level information of different depth features from the convolutional neural network, combining the weight aggregation module to learn the weight of each layer feature adaptively. The learned weights are then loaded to the segmentation graphs obtained by sampling on each feature layer to obtain the final segmentation results.

Computer Science and Information Systems 23(1): v–vii https://doi.org/10.2298/CSIS260100vI

Guest Editorial: Advances in Databases, Information Systems, and Database Engineered Applications – Recent Trends, and Open Issues

Mirjana Ivanović[1], Richard Chbeir[2], Johann Gamper[3], Sergio Ilarri[4], Yannis Manolopoulos[5], and Joe Tekli[6]

[1] University of Novi Sad, Serbia, mira@dmi.uns.ac.rs
[2] University of Pau and the Adour Region, Anglet, France, richard.chbeir@univ-pau.fr
[3] Free University of Bozen-Bolzano, Italy, johann.gamper@unibz.it
[4] University of Zaragoza, Spain, sillari@unizar@es
[5] University of York, Europe Campus, Thessaloniki, Greece, imanolopoulos@city.yorkeuropecampus.eu
[6] Lebanese American University, Lebanon, joe.tekli@lau.edu.lb

Nowadays, modern, technically, and technologically rapidly developing society and the progressive advancement of almost all ICT disciplines bring new intelligent and smart interconnected socio-technological systems, architectures, components and services in everyday life and activities. Two long-lasting events with high reputation: the 28th European Conference on Advances in Databases and Information Systems (ADBIS) and the 28th International Database Engineered Applications Symposium (IDEAS), were held in Bayonne, France, during August 28-31, 2024, attracting high-quality researchers and authors to present their emergent and contemporary scientific results and achievements.

ADBIS conference provides an international platform for the presentation of research on database theory, development of advanced DBMS technologies, solutions for data engineering and analytics, and their advanced applications. The main objective of ADBIS series is to provide a forum for the dissemination of research accomplishments and to promote interaction and collaboration among the database, information systems, data science, artificial intelligence and machine learning communities from European countries and the rest of the world. ADBIS'2024 program included 15 full papers carefully reviewed and selected from 43 submissions. The conference focus was on Algebra, Models and Schemata, Discovery and Data Analysis, Algorithms and Optimization, Access Methods and Query Processing, Advanced Architectures, Machine Learning, Large Language Models.

IDEAS symposium addresses engineering and application aspects of data regardless of the database models. It provides an international forum for discussion of the problems of engineering data driven systems involving not only database technology but also related areas of artificial intelligence, communication, information retrieval, human machine interface, multimedia, NLP, privacy, security and others. The goal of IDEAS is to foster closer interaction among the industrial, research and user communities and provide an opportunity for them to meet, discuss ideas, examine the current ones and develop new solutions and research directions. IDEAS'2024 program included 21 full papers carefully reviewed and selected from 34 submissions. The symposium focused on Theoretical Issues, Classification, Text and Languages, Big Data, Query Processing and Applications, Machine Learning and Rules, Recommendation, and Indexing and Event Detection.

This special section of ComSIS Journal contains revised and extended versions of papers presented at the ADBIS and IDEAS 2024 events. We have selected 10 papers from both events and invite authors to submit extended versions to be considered for publishing in ComSIS Journal. Finally, after rigorous reviewing procedure 4 papers have been accepted for publication.

The first paper entitled "Mitigating Out-of-Vocabulary Challenges in Embedded devices Vulnerability Classification: An Ensemble Embedding Approach with Bidirectional Context Modeling" by Aissa Ben Yahya, Hicham El Akhal and Abdelbaki El Belrhiti El Alaoui, contributes to the area of embedded systems and vulnerable cyberattacks. The effectiveness of commonly used pre-trained word embeddings like Word2Vec, GloVe, and FastText is limited by reliance on training corpora that lack domain specific terminology reducing classification performance. To address this limitation, the authors proposed a novel ensemble embedding technique that combines multiple pre-trained embeddings to improve vulnerability classification in embedded systems. Extensive experiments on benchmark datasets demonstrated significant potential for enhancing cybersecurity in critical infrastructure applications.

The second paper entitled "Towards Enhancing Data Science Agents with Semantics" by Sayed Hoseini, Maximilian Ibbels, Maximilian Knoll1, Christoph Quix, contributes to the area of data lakes. Contemporary approaches in this area still lack essential ML-specific features, limiting their effectiveness for end-to-end automation. However Automated Machine Learning (AutoML) and Large Language Models (LLMs) offer potentially good solutions. Based on that, the paper presents an integration of AutoML frameworks and LLMs within a data lake system. More specifically the authors proposed an innovative vision that integrates multi-agent frameworks for data science with knowledge graphs that capture historical experience from previous ML experiments. Initial version of their data lake system SEDAR for developing comprehensive, context-aware ML agents and their integration is presented.

The third paper entitled "Towards Explainable Sequential Learning" by Giacomo Bergami, Emma Packer, Kirsty Scott and Slivia Del Din, offers a hybridly explainable temporal data processing pipeline, DataFul Explainable MultivariatE coRrelatIonal Temporal Artifcial intelligence (EMeriTAte+DF) which enables human explainable results. Extending the event-based literature to design specification mining algorithms supporting concurrent constituents the authors' solutions outperform state-of-the-art algorithms for multivariate time series classification. The effectiveness of the proposed methodology premiering the extraction of explainable correlations across multivariate time series dimensions with dataful features are evaluated over four datasets.

The last paper entitled "Siren Federate: Bridging Document, Relational, and Graph Models for Exploratory Graph Analysis" by Georgeta Bordea, Stéphane Campinas, Matteo Catena, and Renaud Delbru, focuses on exploratory graph and graph models. The authors proposed Siren Federate architecture that efficiently supports exploratory graph analysis by bridging document-oriented, relational and graph models. Their contributions include distributed join algorithms, adaptive query planning, query plan folding, semantic caching, and semi-join decomposition for path query. Experimental results show that Siren Federate exhibits low latency and scales well with the amount of data, the number of users, and the number of computing nodes.

We gratefully acknowledge all the hard work and enthusiasm of authors, without whom this special section would not have been possible. In addition, we appreciate for the careful study of the selected papers and the constructive comments made by the reviewers:

Costin Badica, University of Craiova, Romania

Goran Banjac, University of Banja Luka, Bosnia and Hercegovina

Diego Ceccarelli, Bloomberg Company, UK

Zoltan Geler, University of Novi Sad, Serbia

Anastas Misev, Ss. Cyril and Methodius University, North Macedonia

Aleksandra Klasnja Milicevic, University of Novi Sad, Serbia

Vladimir Kurbalija, University of Novi Sad, Serbia

Doina Logofatu, Frankfurt University of Applied Sciences, Germany

Goran Sladic, University of Novi Sad, Serbia

Finally, we hope that readers will enjoy the content of this special section, and that it will inspire them to do high quality research in these modern and attractive areas.

Computer Science and Information Systems 23(1): ix–xi https://doi.org/10.2298/CSIS260100ixI

Guest Editorial: Advances on Information Communication Technologies and Intelligent Systems and their Applications

Mirjana Ivanović, Costin Badica, Adrian Iftene, and Yannis Manolopoulos

[1] University of Novi Sad, Serbia, mira@dmi.uns.ac.rs
[2] University of Craiova, Craiova, Romania, costin.badica@edu.ucv.ro
[3] Alexandru Ioan Cuza University Iaşi, Iasi, Romania, adiftene@info.uaic.ro
[4] University of York, Europe Campus, Thessaloniki, Greece, imanolopoulos@city.yorkeuropecampus.eu

Advances and research achievements in all domains of Computer Science and Information Communication Technologies, especially intelligent techniques and approaches, contribute significantly to quality of everyday life and functioning of society. Two interesting conferences were held in 2024, which provided a forum for researchers to disseminate wide range of their research achievements and scientific results. These two Conference are: the 10th Balkan Conference in Informatics (BCI), and the 18th International Conference on Innovations in Intelligent Systems and Applications (INISTA), both were held during 4-6 of September 2024 in Craiova, Romania.

The focus of the BCI series is to provide a forum for discussions and dissemination of research accomplishments in Computer Science and Information Technology and to promote interaction and collaboration among scientists from the Balkan countries and the rest of the world, and to encourage involvement of young researchers from the region. The BCI'2024 program included 23 full papers that were carefully reviewed and selected from 31 submissions. They covered numerous interesting topics: Data Mining and Machine Learning; Software and Systems; Languages and Text; Learning Issues; Distributed Systems; Medical and Health Issues; Web Issues and Tools; Security and Privacy.

INISTA conference aims to bring together the researchers from the entire spectrum of the multi-disciplinary fields of intelligent systems and to establish effective means of communication between them. In particular, it focuses on wide range of aspects of intelligent systems and related applications, from the points of view of both theory and practice. The INISTA'2024 program included 51 full papers carefully reviewed and selected from 74 submissions. The papers were grouped into the following areas of research and applications: Evolutionary Computing and Optimization; Artificial Intelligence in Cloud Computing; Intelligent Web and Microservices; Natural Language Processing; Biomedical Engineering and Healthcare; Smart Education and ELearning; Multi-Agent Systems; Intelligent Services for Transportation, Energy, Water and Food; IoT, Digital Twins, Robotics; Intelligence in Security, Risk, Fraud and Anomaly Detection; Computer Vision and Big Data.

This special section contains revised and extended versions of papers presented at the BCI'2024 and INISTA'2024 events. We have selected 11 papers from both events and invited authors to submit extended version to be considered for publishing in ComSIS Journal. Finally, after rigorous reviewing procedure 7 papers have been accepted for publication.

The first paper entitled "Using Genetic Programming as a Feature Selector and Classifier Implement Bankruptcy Prediction Models" by Ángel Beade, José Santos, Manuel Rodríguez, contributes to the research of Genetic Programming. The authors used genetic programming approach as a feature selector and classifier to implement bankruptcy prediction models for medium-sized companies. Two sets of input variables were used for prediction models. Also, for feature selector two strategies were defined, based on the statistical relevance of the selected features. An analysis is performed of the improvement obtained with feature selection with both genetic programming-based methods in comparison with the use of complete sets of variables and using genetic programming as a classifier. The proposed classifier is compared with other standard classifiers, and the best results are achieved with the synergy of using genetic programming as a feature selector and as a classifier.

The second paper entitled "Enhancing Network Engineering Capabilities through LLM Fine-Tuning with Automatically Generated Datasets" by Claudiu Traistaru, Florin Pop, Costin Badica, Catalina Mancas, Ionut Muraretu, presented a method for automatically generating domain-specific datasets to finetune open-source LLMs in network engineering. The authors evaluated datasets generated using several open-source LLMs, analyzed the quality of unprocessed knowledge data and the efficacy of cleaning and deduplication methods. The resulting dataset addressed various subjects related to routing, security, and network services.

The third paper entitled "How Gray-Boxed Design Can Improve Explainability in Cloud-Edge Microservice Allocation: Analysis and Case Study" by Jorge Jimeńez García, Ignacio Lacalle Úbeda, Paweł Szmeja, Katarzyna Wasielewska-Michniewska, Przemysław Hołda, Maria Ganzha, Carlos E. Palau Salvador, Costin Bădică, Stefka Fidanova, Marcin Paprzycki, is a kind of survey paper focusing on improvement of explainability in Cloud-Edge microservice allocation. The authors studied the available literature in order to propose a middle ground of mixed AI architectures that combine the performance of blackbox AI models with a more explainable overall architecture. Apart from that they presented an application case of microservice allocation decision over cloud (and cloud-like) infrastructures.

The fourth paper entitled "Audio Feature-based User Profiles for Personalized Music Recommendation: A Dataset-driven Evaluation" by Ionut-Dragos Neremzoiu, Andreea Liliana Bădică, contributes to the area of Personalized Music Recommendation. The authors proposed an experimental content-based track recommendation system, which relies on aggregated features which consider audio feature values from Spotify Data Catalog, track lyrics and popularity ratings. Additionally, they evaluated how relevant the top-5 recommended tracks are to the user profile, and in experimental evaluation they concluded that the recommended tracks are similar to the user's profile.

The fifth paper entitled "Trustless Exchange for Personal Data: Adapting Intellectual Property Trading Security Protocols for Data Sovereignty" by Vijon Baraku, Simeon Veloudis, Iraklis Paraskakis, Poonam Yadav, Rezon Baraku, brings the authors' approach for supporting data sovereignty. For that purpose, they used secure exchange in personal data stores adapting blockchain-based security protocols originally designed for intellectual property trading. They examined how their existing multi-stage verification protocol can be applied to address the requirements of personal data exchange. The proposed innovation supports maintenance of the trustless verification and secure transfer mechanisms

of the original protocol. So, they explored a complementary framework that could enhance existing Personal Data Store architectures contributing to the advancement of data sovereignty in digital ecosystems.

The sixth paper entitled "IoT based Personalized Online Learning Systems (Systematic Mapping Study)" by Edlir Spaho, Betim Çiço, brings systematic overview of IoT Personalized Learning Systems. It is evident that IoT plays an important role in the educational processes as well. The authors presented identified how IoT supports personalization in online learning processes. This mapping study involved screening of more than 6400 articles and 65 of them were identified as having implemented experimental or solution-based research involving IoT. Apart from realizing that the use of IoT in Education to implement personalized learning systems is in its early stages possible gaps in implementing such learning environments are pointed out.

The seventh paper entitled "A Knowledge Graph based Approach for Credit Card Fraud Detection" by George Konstantinos Dimou, Georgia Koloniari, is focused on Credit Card Fraud Detection issues. The authors proposed a novel graph-based approach that constructs a Knowledge Graph from transactional data to model relationships between different entities. A variety of centrality measures, both in an unweighted and in a weighted knowledge graph were calculated in order to capture structural importance. The measures are used to enrich the feature space for multiple Machine Learning models. Comprehensive experiments showcased how the addition of the centrality measures as classification features can significantly improve the performance of classification models.

We gratefully acknowledge all the hard work and enthusiasm of authors, without whom this special section would not have been possible. In addition, we appreciate for the careful study of the selected papers and the constructive comments made by the reviewers:

Veljko Aleksic, University of Kragujevac, Serbia
Reem Alhabib, University of York, UK
Bulent Bolat, Yıldız Teknik Üniversitesi, Turkey
Drazen Brdjanin, University of Banja Luka, Bosnia and Hercegovina
Sanja Candrlic, University of Rijeka, Croatia
Banu Diri, Yildiz Technical University, Turkey
Zoltan Geler, University of Novi Sad, Serbia
Dosam Hwang , Yeungnam University, Korea
Vladimir Kurbalija, University of Novi Sad, Serbia
Martin Diéguez Lodeiro, University of Angers, France
Anastas Misev, Ss. Cyril and Methodius University, North Macedonia
Doni Pracner, University of Novi Sad, Serbia
Zaharije Radivojevic, University of Belgrade, Serbia
Gordana Rakic, University of Novi Sad, Serbia
Eleftherios Tiakas, International Hellenic University, Greece
Violeta Tomasevic, University of Belgrade, Serbia

Finally, we hope that readers will enjoy the content of this special section, and that it will inspire them to do high quality research in these modern and attractive areas.

Computer Science and Information Systems 23(1):1–31 https://doi.org/10.2298/CSIS251110001M

Optimizing Cell-Based Software Architecture through Heuristic Community Detection Approach*

Miloš Milić, Nebojša Nikolić, and Dragana Makajić-Nikolić

University of Belgrade - Faculty of Organizational Sciences
Jove Ilića 154, Belgrade
milos.milic@fon.bg.ac.rs
nebojsa.nikolic@fon.bg.ac.rs
dragana.makajic-nikolic@fon.bg.ac.rs

Abstract. The goal of this research is to investigate the optimization of the Cell-based software architecture. Cell-based software architecture structures a software system into interconnected cells, each comprising multiple elements. This study focuses on optimizing the architecture by determining the optimal number of cells and their internal organization. To achieve this, the Community Detection approach, which identifies closely connected elements, was applied. To preserve cell boundaries, reduce complexity, and enhance modularity, we introduce the concept of functionality, which can be represented by one or more cells. This concept serves as the foundation for optimizing software architecture. A series of experiments were conducted to analyze the problem dimensions that can be addressed through optimization and to evaluate the robustness of the mathematical model. Given that the proposed model is unable to solve large-scale problems efficiently, we developed a heuristic approach and compared its results with those obtained from the mathematical model. The evaluation results indicate that different software architectures can be derived in terms of cell granularity, composition, and interaction. Since each cell can contain multiple elements realized in various architectural styles, the proposed model enables the integration of diverse architectures within a single software system. This flexibility enhances the system's adaptability and overall efficiency.

Keywords: software architecture, cell-based architecture, community detection, architecture optimization, destroy and repair.

1. Introduction

In today's digital age, the application of software systems spans across various domains. These systems enable seamless communication and data exchange within and across different industries. In the interconnected world, software systems can be utilized by a diverse range of clients, and it is essential to ensure they have capabilities to support them effectively.

Software architecture of a system can be defined as the set of elements needed to reason about the system [6], encompassing various software components, their relationships, as well as the properties of components and relationships [12]. Software architecture can

* The paper is an extended version of the FedCSIS 2024 conference paper.

be considered as a blueprint for further software design, based on which various components are created. These software components are used to fulfill the functional requirements of the software system [12].

In addition to functional requirements, software systems should incorporate capabilities related to non-functional requirements such as security, deployability, availability, scalability, reliability, resilience, maintainability, etc. [12]. However, achieving a high level of non-functional requirements can be a challenging task. Non-functional requirements are typically defined as quality attributes of a software system, and are closely related to software architecture [14].

Considering utilization of software systems in various domains, as well as their complexity, ensuring high-quality software becomes an important challenge. Poor architectural decisions can lead to issues such as technical debt, inappropriate resource utilization, and security vulnerabilities [11], [13]. In addition, these decisions are essential in determining the maintainability of software systems, as they establish the foundational structure that influences future software development and evolution [61]. Inappropriate architectural choices can lead to the accumulation of Architectural Technical Debt (ATD) [8], [62], which refers to design decisions that expedite short-term development but hinder long-term system evolution and maintenance [8]. This debt manifests through increased complexity and reduced system adaptability, resulting in high maintenance costs and efforts [8], [62].

Understanding the relationship between software architecture and software quality is essential for developing robust, scalable, secure, and maintainable systems. In this context, software architects and engineers should consider these aspects from the earliest stages of software development [25].

This research investigates the optimization of the Cell-based software architecture. Cell-based architecture considers the organization of a software system in the form of interconnected cells, while each cell can contain multiple elements [1], [55]. The Cell-based software architecture allows each software element within a cell to be realized using its own architectural style, enabling flexibility in design and implementation [1], [55]. By leveraging this approach, the architecture can utilize the benefits of multiple architectural paradigms within different cells. Since each cell operates autonomously and can be managed independently of others, this architecture enhances encapsulation, isolation, and the distribution of software elements [1], [55]. Some open topics in this approach relate to determining the optimal granularity, composition, and interaction of cells [52], as these factors directly impact various software quality aspects. This research examines the optimization of cell-based software architecture, specifically focusing on the number of cells and their internal organization as a community detection problem. In this context, a mathematical model for optimizing Cell-based software architecture was constructed. Additionally, given that this problem is NP-hard, we have developed a heuristic for solving large-scale problems.

The rest of the paper is organized as follows. Section 2 introduces various software architectures that can be applied in the software development process. Additionally, the Cell-based software architecture is presented, as well as the Community Detection problem and its application in different fields. Section 3 defines the problem and proposes a mathematical model for optimizing Cell-based software architecture, along with evalua-

tion and optimization results. The heuristic and numerical results are introduced in Section 4. Finally, discussion and conclusions are presented in Section 5.

2. Background

2.1. Software Architecture

When software design is concerned, various software architectures can be observed. For example, monolithic architecture represents a traditional software design approach. This architecture involves multiple modules that are executed together as a single unit at runtime, resulting in high coupling between the modules [10]. On the other hand, microservice architecture is an alternative to monolithic architecture [48]. In microservice architecture, each element is implemented as a separate microservice, which operates independently as a single unit at runtime. This approach results in low coupling between microservices [10], [48]. However, taking into account that each microservice is managed independently, microservice organization and communication must be carefully considered [24]. In addition, microservices typically require additional components for management, such as microservice orchestration and choreography [58], which can introduce additional complexity. Although monolith and microservice architecture can co-exist within the same system, researchers are exploring approaches decomposing and gradually transitioning from monolithic applications to microservices [2], [15], [42], [53]. Both monolithic and microservice architectures require infrastructure services (e.g., application server, database server, etc.), which can be either on-premises or cloud-based.

Another alternative to monolithic and microservice architectures is serverless software architecture, an approach that focuses on designing services related to specific business capabilities [56]. In this context, Functions-as-a-Service (FaaS) can be coded and deployed, while the underlying infrastructure is managed by the cloud provider [57]. Although this approach allows software engineers to focus on business functions, it results in a high degree of coupling with the infrastructure services provisioned by the cloud provider.

Based on the previous discussion, it can be stated that each software architecture has its own pros and cons that should be carefully considered during the software design process.

2.2. Cell-based Software Architecture

Cell-based architecture can be defined as a software architecture that incorporates multiple units of workload, with each unit known as a cell [55]. Each cell is independent from other cells, does not share state with other cells, and can encapsulate multiple components of different types [1], [55]. Additionally, each cell contains a cell gateway, serving as the central entry point for cell communication. In this context, intra-cell and inter-cell communication can be observed, which is realized with well-defined interfaces and protocols [1]. A specific set of functionalities or services can be incorporated within a cell, defining a cell boundary. In this context, cell-based architecture can be related with domain-driven software design [45].

Conceptual overview of the Cell-based software architecture is presented in Figure 1. The figure depicts two cells with multiple elements, with the cell boundaries outlined by

octagons. Cell A incorporates three elements (e.g., one monolith and two microservices), while Cell B also includes three elements (e.g., three microservices). Additionally, element A2 from Cell A communicates with Cell B through the cell gateway. In this way inter-cell communication is realized [1]. On the other hand, element B2 communicates with element B3. This communication is performed inside Cell B and represents intra-cell communication [1]. Each cell is autonomous and can be managed independently of other cells. As a result, better encapsulation, isolation, and distribution of software architecture elements can be achieved, addressing some of the typical challenges in software architecture design [9].

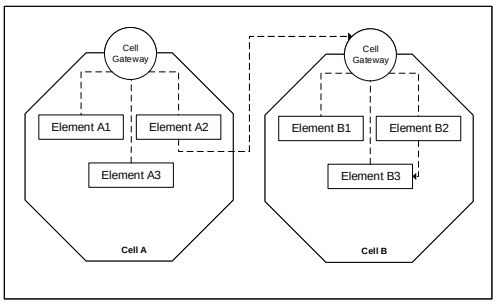


Fig. 1. Conceptual overview of the Cell-based software architecture

Considering that a cell can incorporate multiple capabilities implemented in various architectures, the cell-based approach facilitates the introduction of multi-architecture software development. In this context, the benefits of each applied architecture can be utilized, while their cons can be managed. This approach allows each cell to be independent and iterate individually, resulting in decentralized software architecture [1].

2.3. Community Detection Problem

Community detection problem belongs to the field of Complex Network Analysis. Its most common areas of application are: social networks [16], [38], neuroscience and biology [46], supply chain networks [41], [63], politics, customer segmentation, smart advertising and targeted marketing [30], etc. The community detection approach was also applied in software engineering, as both the process-oriented and object-oriented software architecture can be presented as a complex network [36] characterized by properties like those commonly observed in other complex networks [60]. Authors Pan, Jing, and Li used community detection approach for refactoring the package structures of object-oriented software in order to improve the maintenance process [49]. Software maintenance was also emphasized as the reason for using community detection in research conducted by Huang et al. [27]. Authors Hou, Yao, and Gong applied community detection approach to developer collaboration network in software ecosystem based on developer cooperation intensity [26].

In the context of software architecture, the Community Detection approach can be used to analyze service dependency graphs in microservice architectures [22]. This approach has been applied to facilitate the extraction of microservices from monolithic applications [20], [37]. Based on the defined entity types and their relationships, the Louvain algorithm for graph clustering was employed to identify potential microservice candidates [37]. The research conducted by Filippone et al. proposes microservices extraction by applying graph clustering and combinatorial optimization to maximize cohesion and minimize coupling between microservices [20].

Communities are groups of network's vertices with the common properties and/or role in the network [21]. The community detection problem is to find communities that maximize a given quality function. The solution of the problem is a set of communities such that the number of edges within the community is greater of the number of edges between the community's vertices and the rest of the network (Figure 2).

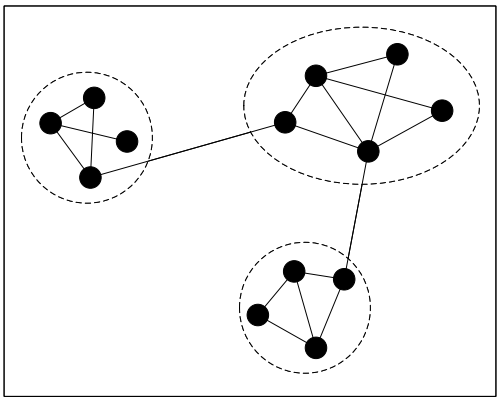


Fig. 2. A simple graph with three communities

There are several quality measures intended to evaluate the structure of communities [17]. In this paper we use the standard and most used measure of quality, called Newman–Girvan modularity [47]. One of the formulations of Newman–Girvan modularity is:

$$Q = \sum_{k \in C} [\frac{L_k}{L} - (\frac{D_k}{2L})^2] \tag{1}$$

where C represents the set of communities, L_k is the sum of the weights of the edges within community k, L is the sum of weights of all edges in the entire network, and D_k is the sum of the degree of the vertices in community k.

The function (1) is nonlinear and it can only be solved for small and medium-sized unweighted graphs [7]. Hence, several solving methods and linearizations of it can be found in the literature [54], [19]. In this paper, we use the variant of modularity function

(1) that enables further linearization, proposed by Alinezhad et al. in [5]:

$$Q = \frac{1}{L} \sum_{k \in C} (\sum_{(i,j) \in E_k} b_{ij} - \frac{1}{4L} \sum_{i,j \in V_k} d_i d_j) \tag{2}$$

where E_k represents the set of edges in the community k of a given graph $G = (E, V)$, V_k is the set of the vertices in the community k. The parameter b_{ij} is the weight of the edge (i, j), $(i, j) \in E$, and d_i is the weighted degree of the vetrex, obtained as sum of the weight of all input and output edges of vertex i:

$$d_i = \sum_{(i,j),(j,i) \in E} b_{ij}, i \in V \tag{3}$$

3. Mathematical Model for Optimizing Cell-based Software Architecture

As previously discussed in Section 2, cell-based software architecture can be depicted as a network of interconnected cells and elements that communicate with each other. Given that the solution of Community Detection problem can identify closely connected items, this section presents a mathematical model for optimizing cell-based software architecture. Optimizing the cell-based architecture can potentially lead to better resource utilization through an optimal number of cells and their internal organization. Additionally, various software quality attributes can be improved.

The model for optimizing Cell-based software architecture is related to the research presented in [44], whose core is based on the mathematical model presented in [5] by Alinezhad et al. However, in that study, we did not explain how the weights of the edges should be determined, as they were randomized in the experiments [44]. In this context, the model will be extended to define the measurement and calculation of these weights. Based on this extension, a software system will be developed, a simulation will be conducted, and the results will be analyzed. Given that this problem is NP-hard, a heuristic for solving large-scale problems will be developed, and its results will be compared with the optimization results obtained from the mathematical model.

When the software architecture whose elements should be grouped into cells based on community detection problem, the elements of the architecture are vertices of the graph $G = (E, V)$. The edges of the graph exist between the elements (vertices) which communicate, while the weight of the edge (i, j), b_{ij} represents the intensity of the communication.

To determine the intensity of communication, various software metrics can be utilized. These software metrics should be observed and collected within a specified time frame, allowing insight into application quality [51]. However, it is important to note that different software metrics may have different scales and measurement units [59], which must be carefully considered. For instance, response time is typically measured in time units (e.g. seconds, milliseconds, microseconds, nanoseconds, etc.), while payload size is typically measured in data units (e.g. bytes, kilobytes, megabytes, gigabytes, etc.). To ensure data consistency and maintain precision, it is essential to standardize measurement units

and perform necessary adjustments before further analysis [23], [50]. Based on the observed and collected metrics, we propose an approach that introduces normalization and aggregation of raw values. In this context, the Min-Max normalization can be applied:

$$m_{ij} = \frac{x_{ij} - min(X)}{max(X) - min(X)} \tag{4}$$

where x_{ij} represents the raw value of the examined software metric, X is the set of all raw values for that metric within the observed time frame, and m_{ij} is its normalized value within the range [0,1].

All relevant software metrics should contribute to the determination of the intensity of communication. However, some software metrics may have a greater impact than others. To account for this, a weighted sum approach can be utilized:

$$b_{ij} = w_1 m_{1ij} + w_2 m_{2ij} + ...w_n m_{nij} \tag{5}$$

where $m_{1ij}, m_{2ij}, ..., m_{nij}$ are normalized software metrics, $w_1, w_2, ..., w_n$ are weights that represent the importance of each metric, and b_{ij} represents the intensity of the communication.

Various software metrics can be utilized to determine the intensity of communication between software elements. In our approach, we will utilize software metrics related to effective interaction between software elements. Two software elements can interact through different communication models, protocols, and patterns [4]. For instance, in the Request-Response model, direct invocations can be calculated. On the other hand, in the Pub-Sub model, published messages received can be calculated. A high number of interactions between two software elements may indicate a high intensity of communication, as frequent exchanges suggest a strong relationship between the elements (in terms of interaction). If not properly managed, high communication can lead to performance anomalies and bottlenecks [28].

Furthermore, the time required for the submission of the request and the completion of the response should also be considered [51], [18]. Request Time refers to the duration taken to transmit the request from the sender software element to the receiver software element, including potential delays such as network latency and queueing delays. On the other hand, Response Time includes the actual time required for request processing, along with additional delays that may occur during transmission of the response (e.g., network delays, queueing delays). It is important to note that the request time and response time may differ based on the applied communication model (e.g. synchronous vs. asynchronous calls). High request time and response time may indicate network issues, inadequate resource utilization, or inefficient processing mechanisms [35], [64]. In this context, both request time and response time should be properly monitored to improve communication efficiency.

During each interaction, data is transmitted between software elements. For instance, a request may contain input data that needs processing, while a response may return output data as the result of the request. These data exchanges can be represented in various formats (e.g., plain text, JSON, XML, binary). Large transmitted data indicate substantial data transfer requirements [40]. If not properly managed, this can lead higher network utilization and increased bandwidth consumption. On the other hand, small transmitted

data suggests lightweight interactions, but has the potential to cause significant overheads [39].

The examined software metrics are commonly employed in contexts where communication between software systems is of primary concern. In addition, these metrics are supported by a variety of tools designed for the measurement and monitoring of software performance. Based on the discussed metrics, communication intensity proposed in equation (5) should be calculated as follows:

$$b_{ij} = w_1 m_{NoIntij} + w_2 m_{ReqTij} + w_3 m_{ResTij} + w_4 m_{ReqSij} + w_5 m_{ResSij} \tag{6}$$

where $m_{NoIntij}$ is normalized software metric Number of Interactions, m_{ReqTij} is normalized software metric Request Time, m_{ResTij} refers to normalized software metric Response Time, m_{ReqSij} represents normalized software metric Request Size, and m_{ResSij} is normalized software metric Response Size.

Considering different communication models, protocols, and patterns, as well as the importance of the examined software metrics, certain elements of equation (6) (i.e., weights or metric values) may vary in magnitude, including the possibility of being zero.

Since the weights inside the parentheses in equation (2) should be calculated only for the edges and vertices belonging to the same community, the set of communities C and binary variables y_{ik} are introduced:

$$y_{ik} = \begin{cases} 1, & \text{if vertice } i \text{ is in the community } k \\ 0, & \text{otherwise} \end{cases}$$

where $i \in V, k \in C$.

Equation (2) now becomes:

$$Q = \frac{1}{L} \sum_{k \in C} \Big(\sum_{(i,j) \in E_k} b_{ij} y_{ik} y_{jk} - \frac{1}{4L} \sum_{i,j \in V_k} d_i d_j y_{ik} y_{jk} \Big) \tag{7}$$

where L represents the sum of the weights of all edges of the entire network.

The nonlinearity $y_{ik} y_{jk}$ could be replaced by auxiliary binary variables z_{ijk}, where $k \in C, (i,j) \in E$:

$$z_{ijk} = \begin{cases} 1, & \text{if edge vertices } i,j \text{ are in the community } k \\ 0, & \text{otherwise} \end{cases}$$

and inequalities:

$$z_{ijk} \geq y_{ik} + y_{jk} - 1, k \in C, i,j \in V \tag{8}$$

$$z_{ijk} \leq y_{ik}, k \in C, i \in V, i,j \in V \tag{9}$$

$$z_{ijk} \leq y_{jk}, k \in C, j \in V, i,j \in V \tag{10}$$

The condition (8) ensures that variable z_{ijk} get the value 1 if both y_{ik} and y_{ik} have the value 1, i.e. the edge (i,j) is inside the community k if both vertices i and j belong to the community k. Since based on condition (8), value of z_{ijk} can be 1 if y_{ik} and/or y_{jk} are equal to zero, the conditions (9) and (10) are introduced to prevent such solutions. Furthermore, if for an edge (i,j) z_{ijk} equals zero for all $k \in C$, it indicates that edge

(i, j) does not belong to any community; instead, it represents a link between two different communities.

Additionally, the model also incorporates the concept of functionality. Functionality can be defined as a set of capabilities allowable and actionable by the software system [29]. Each functionality contains elements focused on a specific domain and should not be mixed to maintain boundaries, reduce complexity, and ensure modularity. In a cell-based software architecture, a single functionality can be represented by one or more cells, forming the foundation for optimizing the software architecture. In addition, different functionalities should not be organized in the same cell, allowing better separation of concerns between cells. As a result, each cell can be independent and managed individually [1].

In addition to the already introduced parameters and variables, notation used for the mathematical model formulation is as follows.

Sets

FC - set of functionalities

F_l - set of vertices of l-th functionality, $F_l \subset V, \bigcap_{l \in FC} F_l = \emptyset, \bigcup_{l \in FC} F_l = V$

Parameters

e – lower bound of the number of vertices in communities

Variables

$$x_k = \begin{cases} 1, & \text{if community } k \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

The final mathematical model, taken from [5] and extended with constraints related to functionalities, is listed below.

$$\max f(z) = \frac{1}{L} \sum_{k \in C} \Big(\sum_{(i,j) \in E_k} b_{ij} z_{ijk} - \frac{1}{4L} \sum_{i,j \in V_k} d_i d_j z_{ijk} \Big) \tag{11}$$

s.t.

$$z_{ijk} \geq y_{ik} + y_{jk}, k \in C, i, j \in V \tag{12}$$

$$z_{ijk} \leq y_{ik}, k \in C, i, j \in V \tag{13}$$

$$\sum_{k \in C} y_{ik} = 1, i \in V \tag{14}$$

$$y_{ik} \leq x_k, i \in V, k \in C \tag{15}$$

$$\sum_{i \in V} y_{ik} \geq e x_k, k \in C \tag{16}$$

$$y_{ik} + y_{jk} \leq 1, k \in C, i \in F_l, j \in F_p, l \neq p \tag{17}$$

$$x_k = \{0, 1\}, k \in C \tag{18}$$

$$y_{ik} = \{0, 1\}, k \in C \, i \in V \tag{19}$$

$$z_{ijk} = \{0, 1\}, k \in C \, i, j \in V \tag{20}$$

The objective function (11) represents the modularity measure linearized by replacing $y_{ik} y_{jk}$ with z_{ijk} in (7) . Since this function should be maximized, the first addend in parentheses will be as large as possible. Thus, the branches that have a greater weight

will be within the same community, that is, the software elements with more frequent communication will be in the same cell. Constraints (12) and (13) are related to linearization. Constraint (14) ensures that each vetrex is assigned exactly to one community. The constraint (15), the value 1 is set to x_k if some vertex is assigned to the k-th community. Constraint (16) is related to the minimal number of vertices assigned to the existing communities. Constraint (17) provides that vertices of different functionality cannot belong to the same community, i.e. only vertices of the same functionality can be in the same community. Constraints from (18) to (20) are related to the binary restrictions on the variables.

If necessary, additional constraints can be introduced. For example, although the parameter *e* defines the lower bound for the number of vertices in communities, an additional constraint can be added to specify a different minimum number of vertices for a particular community. This allows for fine-grained definition of cell structure in specific circumstances.

For example, if some of the software element should be isolated in a cell, a set of such element $VI \subset V$ and additional constraint can be included into mathematical model:

$$y_{ik} + y_{jk} \leq 1, k \in C, i \in VI, j \in V, i \neq j \tag{21}$$

Additionally, if some elements should be in the same cell, regardless the connections between them, the mathematical model can be extended as follows.
G - set of predefined groups of elements
V_q - set of elements predetermined to be in the same cell, $q \in G$

$$y_{ik} = y_{jk}, k \in C, i, j \in V_q, q \in G \tag{22}$$

$$y_{ik} + y_{jk} \leq 1, k \in C, i \in V_q, j \in V \setminus V_q \tag{23}$$

The constraint (22) ensures that the predetermined elements are in the same cell but allows other elements to be assigned to that cell as well. If it is necessary to assign to the same cell only the elements from $q \in G$, constraint (23) should be included into mathematical model.

3.1. Evaluation

For evaluation purposes, we have developed a software system related to a manufacturing domain (e.g., mobile phone manufacturing). The software architecture of this system is presented in Figure 3. Multiple software elements can be observed in the figure. The system incorporates five microservices (i.e., *inventory*, *product*, *order*, *notification*, and *payment*), as well as two monoliths (i.e., *legacy* and *production*). RESTful web services are used for communication between software elements, while the JSON format is used for data transfer. All software elements were executed within Docker containers, ensuring their portability across various operational environments [32]. Considering that containers are typically executed in isolation, we have created a virtual network for their communication (depicted as the *comsis-network* entity in the figure). Source code and installation instructions are provided in the Data Availability Statement section.

In order to determine the intensity of communication between these software elements, a simulation was conducted using Apache JMeter to create performance testing

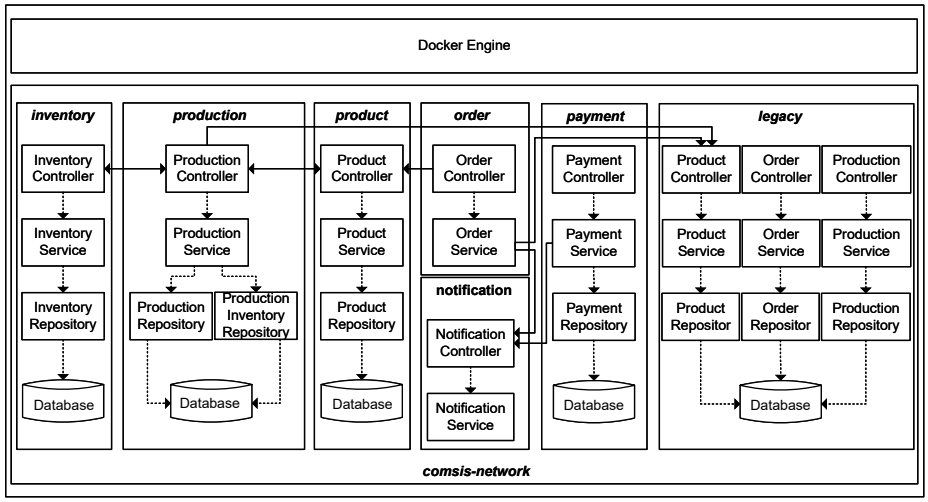


Fig. 3. Software architecture of the examined software system

plans [34]. The testing included a random number of invocations, threads, and ramp-up periods (i.e., delays) in communication, along with diverse input data sourced from CSV files containing mock data. For simulation purposes, all elements ran on a computer equipped with an Intel Core i7 (8th Generation) 2.00 GHz processor, 16 GB of RAM, and a 512 GB SSD drive. The simulation ran for 120 seconds, generating 336,858 effective interactions, which were then used to compute normalized software metrics. Table 7, located in the Appendix, presents the effective interactions between software elements, total and average values of software metrics, and their normalized counterparts. Based on the normalized metrics, the communication intensity – also shown in Table 7 – was calculated using equation (6), with each metric assigned an equal weight. The simulation test plan and results are available in the Data Availability section. The optimized structure of the solution can be graphically presented based on the results. Figure 4 presents the results of the optimization of a manufacturing software system (e.g., mobile phone manufacturing). The input includes defined Production and Purchasing functionalities. The Production functionality comprises two monolithic applications (i.e., $production$ and $legacy$) and two microservices (i.e., $inventory$ and $product$), while the Purchasing functionality consists of three microservices (i.e., $order$, $payment$, and $notification$). Additionally, the communication between these elements is specified (the weights of the edges in Figure 4).

Based on the performed optimization, the resulted solution includes three communities (named Community A, Community B, and Community C), each containing different elements (see Figure 4). In the following text these communities will be referred to as the Production Cell (Community A), Purchasing Cell (Community B), and Legacy Cell (Community C).

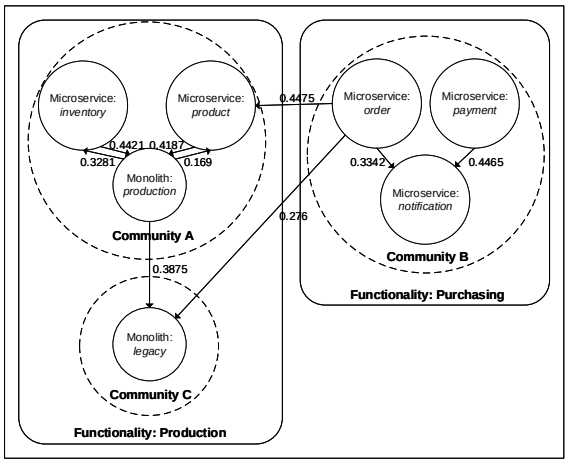


Fig. 4. Results of the optimization of a manufacturing software system

An additional observation pertains to the Legacy Community, which contains only one element (i.e., the $legacy$ monolith). A legacy software system is defined as a core system that has been functioning correctly in production for decades [33]. Considering the prevalence of legacy systems today, researchers are exploring approaches to migrate these systems to modern architectures [3], [31], [43]. In the context of cell-based software architecture, the legacy element is incorporated within a specific cell. From the optimization model perspective, this is represented as an additional constraint that restricts the particular cell structure:

$$y_{legk} + y_{jk} \leq 1, k \in C, j \in V \setminus leg \tag{24}$$

where leg is the index of the variable corresponding to the legacy monolith. This constraint ensures that no other element can be in the cell containing the legacy monolith.

Another interesting observation pertains to the Cell Gateway component. While this component is not explicitly represented in the mathematical model, it serves as the central entry point for cell communication [1]. Therefore, we have included one gateway per cell based on the optimal solution. The final software architecture is shown in Figure 5.

In order to validate the mathematical model, we conducted a series of experiments. The evaluation considered three functionalities, with the number of elements varied. Each element represents an instance of software architecture encapsulating specific features (e.g., a monolith with multiple features, a microservice containing a single feature). Considering that these elements can vary in terms of their applied software architecture and size, a software system can encompass numerous elements. Within this context, the first functionality incorporated 40 percents of the elements, while the remaining elements were equally divided between the other two functionalities. Taking into account that elements cooperate with each other, each element has at least one connection with another element within the same functionality, while up to 40 percents of elements have double connections within the same functionality. Finally, considering that all functionalities are part of

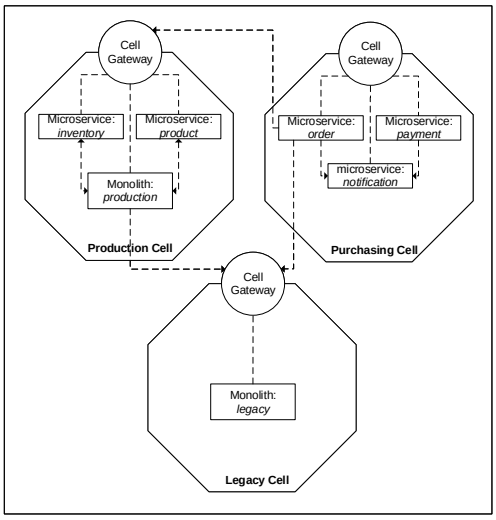


Fig. 5. Cell-based software architecture based on the optimal solution

the same software system, two connections between elements from different functionalities are also established. The elements previously discussed, such as the edges and their weights (the intensity of communication between the elements - parameter b_{ij}) are obtained using equation (6), where the metrics values are randomly generated as explained for the manufacturing software system. The lower bound of the number of vertices in communities parameter e is set to 2.

All optimizations were performed solved using GLPK software on a laptop computer equipped with 11th Gen Intel(R) Core(TM) i5 and 16 GB of RAM. The solving method used in the GLPK software was Branch and Cut, with Gomory's mixed integer cuts, MIR (mixed integer rounding) cuts, mixed cover cuts, and clique cuts options.

The first experiment was conducted to determine whether the metric weights in equation (6) affect the grouping of the elements into cells. The optimization was performed for a graph with 30 vertices and 38 edges in eight scenarios. In the first scenario, the weight of all metrics is equal (0.2). In scenarios two through six, only one metric is considered important to the decision maker. In the seventh scenario only the request and response time metrics are important (with w_2 and w_3 equal to 0.5). In the last scenario, only the request and response size metrics are important (with w_4 and w_5 equal to 0.5). The results of the optimizations are given in Table 1, where the rows represent eight scenarios, and the columns refer to three functionalities and the grouping of its elements.

Table 2 shows the results of the grouping when certain metrics are not important to the decision maker. In the first five cases, the weights w_1 to w_5 from the equation (6) are equal to zero, that is, corresponding metrics are considered irrelevant. In the sixth scenario, the request and response time are not important (with w_2 and w_3 equal to zero), while the last scenario refers to the irrelevance of the request and response size (with w_4 and w_5 equal to zero).

Table 1. Grouping into communities depending on metric weights (important metrics)

Case	f1	f2	f3
equal	{1,2,4,5,7,8,11}{3,6,9,10,12}	{13,14,15,16,17,19,20}{18,21}	{22,23,24,26,28,29,30}{25,27}
w_1	{1,5}{2,4,7,8,11}{3,6,9,10,12}	{13,16,17,20}{14,15,19}{18,21}	{22,23,26,29}{24,28,30}{25,27}
w_2	{1,2,4,5,7,8,11}{3,6,9,10,12}	{13,14,15,16,17,19,20}{18,21}	{22,23,29,30}{24,26,28}{5,27}
w_3	{1,2,4,5,7,8,11}{3,6,9,10,12}	{13,16,17,19,20}{14,15}{18,21}	{22,23,24,26,28,29,30}{25,27}
w_4	{1,5}{2,4,7,8,11}{3,6,9,10,12}	{13,14,15,16,17,19,20}{18,21}	{22,23,24,26,28,29,30}{25,27}
w_5	{1,5}{2}{3,6,9,10,12}{4,7,8,11}	{13,20}{14,15,16,17,19}{18,21}	{22,23,24,26,28,29,30}{25,27}
w_2w_3	{1,2,4,5,7,8,11}{3,6,9,10,12}	{13,14,15,16,17,19,20}{18,21}	{22,23,24,26,28,29,30}{25,27}
w_4w_5	{1,2,4,5,7,8,11}{3,6,9,10,12}	{13,14,15,16,17,19,20}{18,21}	{22,23,24,26,28,29,30}{25,27}

Table 2. Grouping into communities depending on metric weights (irrelevant metrics)

Case	f1	f2	f3
w_1	{1,2,3,4,6,8,10,11,12}{5}{7}{9}	{13,15,16,17,18,21}{14,19,20}	{22,23,24,26,28}{25,27}{29,30}
w_2	{1,2,4,8,11}{3,6,10,12}{5,9}{7}	{13,14,15,16,17}{18,21}{19,20}	{22,23,24,26,28}{25,27}{29,30}
w_3	{1,2,4,7,8,11}{3,5,6,10,12}{9}	{13,15,16,17}{14,18,19,20,21}	{22,23,24,26,28}{25,27}{29}{30}
w_4	{1,2,3,4,5,6,8,10,11,12}{7}{9}	{13,15,16,17}{14,19,20}{18,21}	{22,23,24,26,28,29}{25,27}{30}
w_5	{1,2,4,7,8,11}{3,6,10,12}{5}{9}	{13,14,15,16,17,19,20}{18,21}	{22,23,24,26,28}{25,27}{29}{30}
w_2w_3	{1,2,3,4,6,8,10,11,12}{5}{7}{9}	{13,14,15,16,17,18,19,20,21}	{22,23,24,25,26,27,28}{29}{30}
w_4w_5	{1,2,4,5,9,11}{3,6,10,12}{7,8}	{13,14,15,16,17}{18,21}{19,20}	{22,23,24,25,26,27,28}{29}{30}

Based on the clustering in Tables 1 and 2, it can be concluded that the metric weights influence the resulting clustering. This effect is especially evident when comparing the grouping results in the case of extreme importance of a given metric ($w_i = 1, i = \overline{1,5}$) and its complete irrelevance ($w_i = 0, i = \overline{1,5}$). Although there are the same groupings within individual functionalities, they differ at the level of the entire software. The only exception occurs when multiple metrics are equally important (cases "equal", w_2w_3, and w_4w_5 in Table 1).

Since no optimal solutions were obtained for the 5-minute time limit in any case, the quality of the obtained solutions was analyzed based on the gap relative to the LP relaxation of the problem. The LP relaxation of the mathematical model (11)-(20) was obtained by omitting the integer constraints (18)-(20). Figure 6 shows the modularity values (the value of the objective function (11)) for the solution obtained after 5 minutes of optimization, as well as for the LP relaxation solution. The first eight cases correspond to those in Table 1, while the remaining seven refer to the cases in Table 2.

The average gap value is 4.8%. The smallest value of the gap of 0.7% is obtained for the case when the weights w_4 and w_5 are equal to 0.5, while the smallest refers to the case when w_1 is equal to 1. It can be concluded that the obtained solutions are in most cases close to, or in some cases equal to the optimal solution. The 5-minute time limit was sufficient to obtain high-quality solutions, but not enough to finish the search of the entire solution space.

To examine the dimensions of the problem that can be solved through optimization and the robustness of the mathematical model, the next experiments were conducted for the graphs whose dimensions are given in Table 3. The columns named vertices and edges give the number of vertices and edges, respectively. In column L, the range of values of the parameter L in 10 instances is shown.

Given that the model parameters are randomly generated, the mathematical model (11)-(20) was applied to ten instances of each graph in Table 3. The execution time was

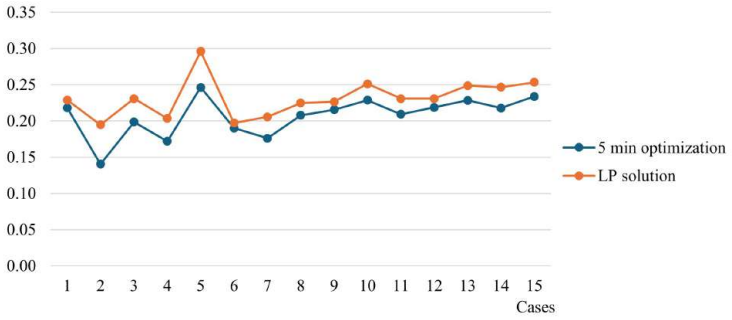


Fig. 6. The modularity obtained after 5 minutes optimization and the LP solution

Table 3. Summary of data sets used in the experiments

Case	Number of vertices	Number of edges	L
1	25	37	18.09-20.47
2	30	38	17.88-20.10
3	35	55	26.77-30.18
4	40	52	24.24-28.44
5	45	71	34.02-37.58
6	50	64	30.09-35.29
7	60	77	35.58-40.51
8	70	90	43.73-48.38
9	80	103	47.55-54.21
10	90	116	56.74-61.05
11	100	128	60.73-67.70
12	150	193	94.76-99.77
13	200	257	122.12-132.84
14	300	498	241.21-249.47

limited to 5 minutes. The average results are shown in Table 4, where the column Dim represents the number of vertices. The columns Number_of_cells and Max_cell_size represent the average number of cells and the average maximal size of the cell within 10 instances. The column Modularity shows the average modularity. The last three columns are related to the gap relative to the solution of the LP relaxation: the average gap, standard deviation and 95% confidence interval.

For the given 5-minute optimization limit time, it was possible to obtain a feasible (integer) solution only for 25-50 vertices. Figure 7 shows the value of the gap in all the instances of these six cases.

The largest gap was obtained in the second instance of the 45-vertices case, while the smallest gap was obtained in the eighth instance of the 50-vertices case. However, regardless of relatively large gaps, the mathematical model showed robustness when looking at the grouping of gaps within the same case (Figure 7).

Since an integer solution could not be obtained in five minutes for cases of 60 or more vertices, the next step is to develop heuristics to solve larger-dimensional problems.

Table 4. Average results of the optimization with limited time

Dim	Number of cells	Max cell size	Modularity	Gap		
				average(%)	SD	CI(%)
25	3.40	5.9	0.6596	26.37	0.0101	[25.65,27.10]
30	6.10	7	0.7932	13.2	0.0063	[12.73,13.64]
35	6.00	9.7	0.6368	30.9	0.0270	[28.93,32.79]
40	7.3	7	0.8377	10.5	0.0091	[9.85,11.15]
45	8.3	10.9	0.6041	35.7	0.0542	[31.78,39.54]
50	10.7	7	0.8359	11.9	0.0395	[9.07,14.73]

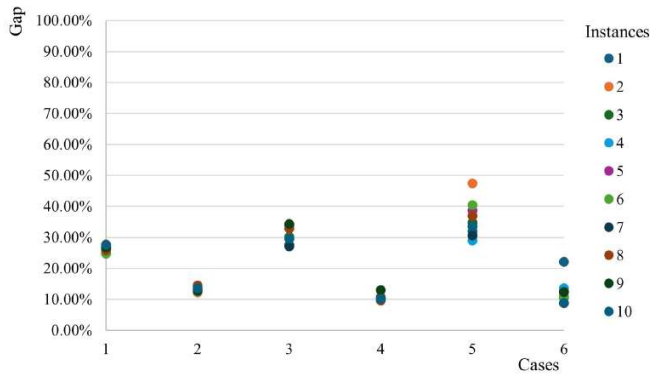


Fig. 7. The gap between the solution obtained after 5 minutes and the LP solution

4. Heuristic Approach to Problem Solving

As discussed in Section 3, functionalities serve as the foundation for optimizing the cell-based software architecture. Given that distinct functionalities should not be grouped within the same cell and that each functionality can span multiple cells, the following solution approach focuses on optimizing each functionality individually. By addressing each functionality separately, complexity is reduced while preserving clear boundaries and enforcing the separation of concerns between functionalities and cells [1]. Consequently, the following algorithms should be applied iteratively for each functionality, with the results (i.e., modularities and identified communities) analyzed collectively to derive the optimal architecture for the entire software system.

First, we will explain the auxiliary *Greedy algorithm*. The main idea of our *Greedy algorithm* is shown in Figure 8. Using the *Greedy algorithm*, we increase the current modularity by merging (uniting) selected communities. Let C' be a set of communities, and $k, l \in C', k \neq l$. The new set of communities obtained by merging the communities k and l, while leaving the other communities unchanged is denoted as C_{kl}. Let Q_{kl} represent the modularity of C_{kl}.

Below is the pseudocode of our *Greedy algorithm*. First, for all pairs $k, l \in C'$ we determine C_{kl} with modularity Q_{kl} (line 2-3) and determine the largest among them Q_{rs} (line 4). If the obtained value is greater than the current modularity Q then the current community C with modularity Q becomes C_{rs} with modularity Q_{rs} (line 5-6). The de-

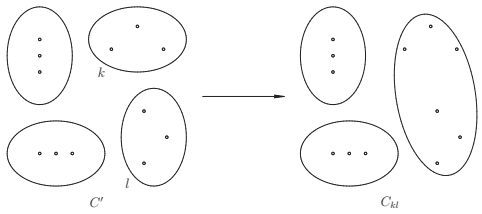


Fig. 8. Cell merging process in the *Greedy algorithm*

scribed procedure is repeated as long as it is possible to improve the modularity Q (loop 1-7).

$Greedy\ algorithm$

Input: Graph G and set of communities C with modularity Q.
Output: New set of communities C' with modularity Q'.
Initialization $C' = C$ and $Q' = Q$;
1: **repeat**
2: **for all** $k, l \in C', k \neq l$ **do**
3: Determine C_{kl} with modularity Q_{kl};
4: Determine $Q_{rs} = \max_{k,l \in C, k \neq l} Q_{kl}$;
5: **if** $Q_{rs} > Q'$ **then**
6: $Q' = Q_{rs}$ and $C' = C_{rs}$;
7: **until** $Q' \neq Q_{rs}$;
Print out C' with modularity Q'.

Now, with the *Destroy and repair algorithm*, we increase current modularity by alternately disassembling and merging cells. The pseudocode of the algorithm is given below. The initial set of communities C^*, with modularity Q^*, is obtained by applying the *Greedy algorithm* on the trivial set of communities $C = \{\{i\} : i \in V\}$ (line 1). We randomly select two communities k and l (line 3), which we then decompose into individual vertices, while the other cells remain unchanged (Figure 9). Let $\overline{C_{kl}}$ denote the set of communities obtained in this manner, with modularity $\overline{Q_{kl}}$ (line 4.).

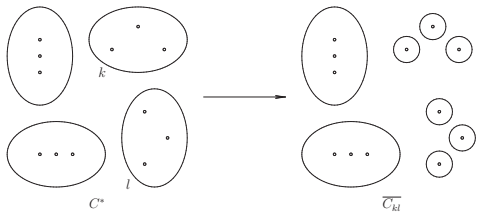


Fig. 9. Cell decomposition process in the *Destroy and repair algorithm*

By applying the *Greedy algorithm*, the communities are reconnected, resulting in a new set of communities C' with modularity Q' (line 5). If the new modularity Q' exceeds the current best modularity Q^*, then the destroying and repair process was successful and the set of communities C', with modularity Q', becomes the new best (line 6-7). Otherwise, the counter of unsuccessful attempts to destroy and repair is increased (line 8-9). Since the process of destroying is stochastic (line 3), the described process of destroying and repairing is repeated until the maximum number of consecutive unsuccessful attempts $Count_{max}$ is reached (loop 2-10).

Destroy and repair algorithm

Input: Graph G, $Count_{max}$.
Output: Set of communities C^* with modularity Q^*.
Initialization: $C = \{\{i\} : i \in V\}$ and $Q = Counter = 0$.
1: $(C^*, Q^*) = Greedy(G, C, Q)$;
2: **repeat**
3: Randomly choose $k, l \in C^*, k \neq l$;
4: Determine $\overline{C_{kl}}$ with modularity $\overline{Q_{kl}}$;
5: $(C', Q') =$ ***Greedy*** $(G, \overline{C_{kl}}, \overline{Q_{kl}})$;
6: **if** $Q' > Q^*$ **then**
7: $(C^*, Q^*) = (C', Q')$ and $Counter = 0$;
8: **else**
9: $Counter = Counter + 1$;
10: **until** $Counter = Count_{max}$;
Print out C^* with modularity Q^*.

So, with the *Destroy and repair algorithm*, we alternately disassemble and assemble the community structure in order to obtain a structure with as much modularity as possible.

4.1. Evaluation of the heuristic

The heuristic algorithm was evaluated on the graphs given in Table 3. The average results are shown in Tables 5 and 6. Table 5 contains the same elements as Table 4: the number of vertices, the average number of cells, the average maximal cell size, the average modularity, the average gap, standard deviation, and 95% confidence interval. Data on heuristic execution times are given in Table 6: average time in seconds, standard deviations, and 95% confidence interval.

The detailed data for all instances of 14 cases are given in Figures 10 and 11. Figure 10 shows the gap values, while Figure 11 shows the time to reach the algorithm's stopping criterion.

Based on Figures 10 and 11, it can be concluded that the heuristic is robust. Changes in parameters do not affect the quality of the solution or the time required to obtain it when the heuristic is applied to graphs of the same dimensions.

Finally, heuristic was used to examine the impact of metric weighting schemes on community structures. For that purpose, we have generated a new graph consisting of

Table 5. Average results of the heuristic

Dim	Number of cells	Max cell size	Modularity	Gap		
				average(%)	SD	CI(%)
25	3.40	9.4	0.6596	26.37	0.0101	[25.65,27.1]
30	6.10	7	0.7932	13.18	0.0063	[12.73,13.64]
35	4.10	12.1	0.6730	26.93	0.0112	[26.13,27.74]
40	7.1	7	0.8399	10.26	0.0032	[10.04,10.49]
45	4.8	13.8	0.6840	27.13	0.0139	[26.15,28.13]
50	9	7	0.8311	12.40	0.0024	[12.24,12.58]
60	10	7	0.8916	6.84	0.0016	[6.73,6.96]
70	10	7	0.8988	6.66	0.0007	[6.61,6.72]
80	13.0	7	0.9179	5.19	0.0011	[5.11,5.27]
90	14.0	7	0.9148	5.84	0.0018	[5.72,5.98]
100	16.0	7	0.9313	4.39	0.0006	[4.35,4.44]
150	23.0	7	0.9545	2.89	0.0003	[2.87,2.92]
200	30.0	7	0.9656	2.18	0.0002	[2.17,2.2]
300	16	20	0.9354	5.56	0.0003	[5.55,5.59]

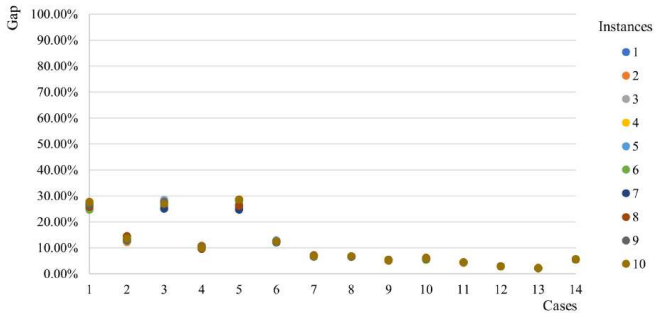


Fig. 10. The gap between the heuristic solution and the LP solution

30 vertices and 38 edges. Using the *Destroy and repair algorithm*, we have solved 15 scenarios. Table 8, incorporated in the Appendix, shows first eight scenarios: 1. the weight of all metrics is equal (0.2), in scenarios 2-6. only one of metric w_1-w_5 respectively have the weight equal to 1, while the other metrics have the weight 0, in scenario 7 the weights of w_2 and w_3 are equal to 0.5, and in scenario 8 the weights of w_4 and w_5 are equal to 0.5. Table 9, located in the Appendix, shows the last seven scenarios: in the first five columns only one metric is not important and have the weight 0, and in the last two columns, the weighst w_2,w_3, and w_4,w_5 are equal to 0, respectively. In all seven cases the weights of the remaining metrics are equal.

The community structures for given scenarios as well their modularity are given in sub-figures in Figure 12 and their captions. The missing scenarios have the same community structure as scenario w equal. However, their modularity differs from the modularity of the w equal scenario due to the different weights of the edges. The modularity of the missing scenarios: $w_5 = 1$, $w_4 = w_5 = 0.5$, $w_1 = 0$, $w_2 = 0$, $w_3 = 0$, $w_5 = 0$, and $w_2 = w_3 = 0$ are: 0.717, 0.69, 0.663, 0.668, 0.671, 0.648, and 0.68, respectively.

Table 6. Heuristic execution times

Dim	average(sec)	SD	CI(sec)
25	0.21	0.0170	[0.21,0.23]
30	0.22	0.0095	[0.22,0.23]
35	0.23	0.0261	[0.21,0.25]
40	0.24	0.0079	[0.23,0.25]
45	0.32	0.0301	[0.3,0.35]
50	0.31	0.0242	[0.3,0.33]
60	0.35	0.0262	[0.33,0.37]
70	0.43	0.0279	[0.42,0.46]
80	0.53	0.0581	[0.49,0.58]
90	0.79	0.0817	[0.73,0.85]
100	0.88	0.1056	[0.81,0.96]
150	2.48	0.1658	[2.36,2.6]
200	6.42	0.2716	[6.23,6.62]
300	59.88	3.8869	[57.11,62.67]

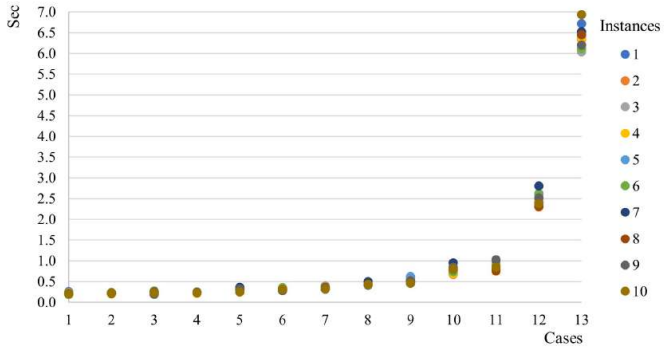


Fig. 11. Heuristic solution finding times

It can be concluded that different metric weights influence the detected community structures, as well as the values of modularity, even when the same community structure is obtained. In the examined case, however, these differences are relatively small. The standard deviation of modularity across all 15 scenarios is 0.022 (corresponding to 3.34% of the average value), and the 95% confidence interval is [0.653, 0.677]. These modest variations can be partly explained by the fact that the differences between the metric values—and thus the resulting edge weights—were not substantial. In cases where the edge weights differ more strongly, the variations in modularity may become more pronounced and potentially more relevant for decision-making.

5. Discussion and Conclusions

In this paper, the problem of grouping software elements into cells is addressed as a community detection problem – an optimization problem whose goal is to find a set of

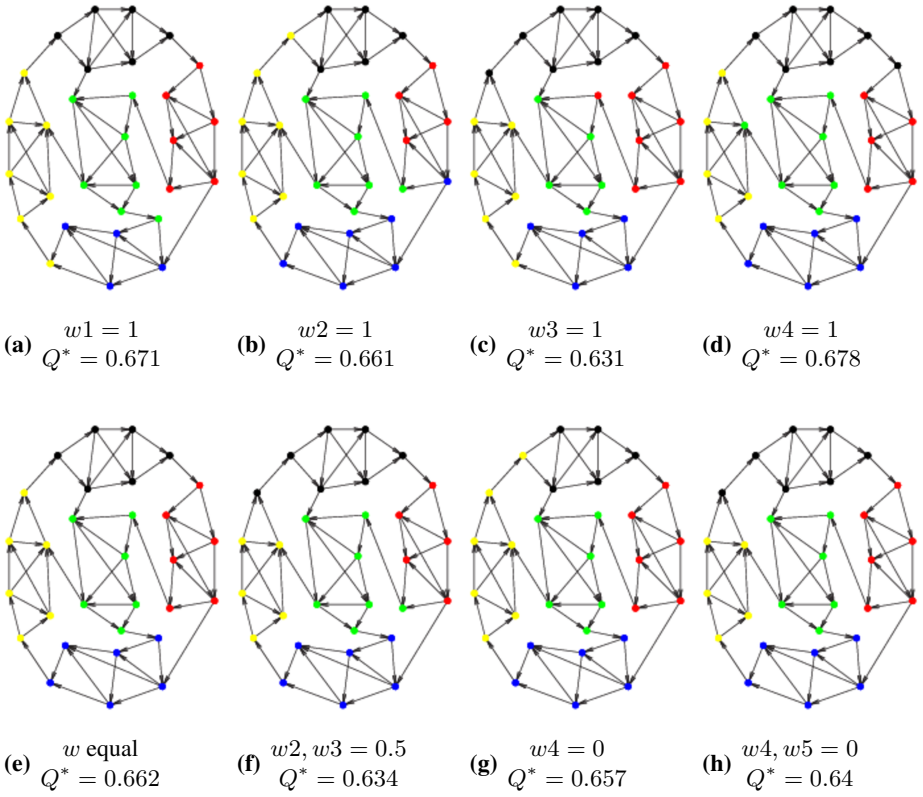


Fig. 12. The community structures for given scenarios

communities such that the connections between nodes within the community are stronger than the connections with other nodes in the graph. The elements of software are represented by vertices, communication between them by edges, and cells by communities. Following this analogy, a corresponding mathematical model was formulated.

Our model leverages a diverse set of software metrics and their normalization to accurately determine the intensity of communication, represented by the b_{ij} parameter. By employing a weighted sum approach, the model allows software architects to adjust metric priorities based on the specific characteristics of the software system being designed. For instance, in time-critical systems, higher weights can be assigned to Request Time and Response Time, ensuring that performance-related factors take precedence. Conversely, for data-intensive applications, greater emphasis can be placed on Request Size and Response Size to reflect the importance of data exchange. Additionally, the model is flexible, enabling the incorporation of additional software metrics and the customization of their weights. This adaptability influences the granularity and composition of cells, allowing for dynamic adjustments that align with evolving system requirements. As a continuous process, this approach promotes optimal resource utilization while maintaining architectural efficiency. The mathematical model formulated in this paper, in addition to determining the clusters, enables experimentation with the importance of individual metrics.

The objective of the mathematical model is the modularity function, which is originally nonlinear. Therefore, the linearization of this function was applied to ensure the possibility of obtaining solutions to problems of larger dimensions. However, despite linearization, integer (non-optimal) solutions were only obtained for graphs with up to 50 nodes within the 5-minute time limit. That is why heuristics were developed which have proven effective in solving the problem of grouping software elements into cells. Figure 13 shows a comparison of the modularity values obtained through 5-minute limited optimization and the heuristic, while Figure 14 gives the gaps for these two groups of the results.

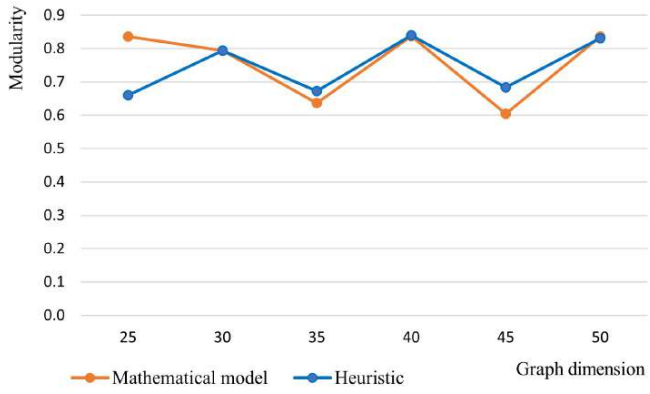


Fig. 13. Modularity function of 5-munites optimization and heuristc

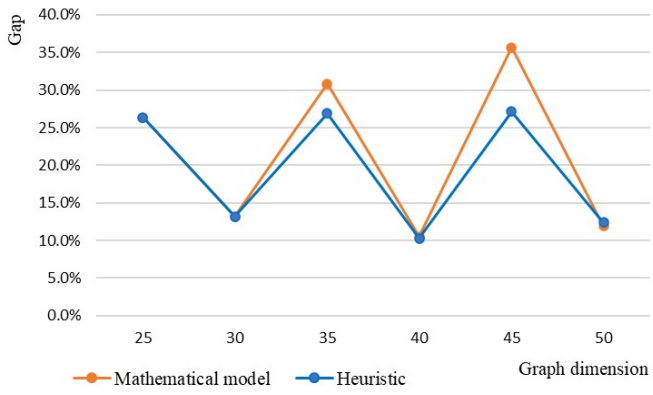


Fig. 14. Gaps in 5-munites optimization and heuristic

The modularity obtained by the heuristic is the same or better than the modularity obtained by the optimization limited to 5 minutes. Except for dimensions 25 and 30, where optimization and heuristics give the same results, the gap with respect to the LP model solution is much smaller for heuristics than for 5-minute optimization for all other dimensions. It is important to emphasize that the time required to obtain a solution using heuristics is, in all cases, less than one second, which is significantly shorter than the optimization time. Of course, it is possible to use a different (larger) counter value. It is also possible to select more than two communities in the destroying process. These changes would improve the quality of the obtained solutions, while increasing the time, that would still remain shorter than the optimization time.

The main goal of this research was to investigate the validity of modeling the cell-based software architecture as a community detection problem. The research has certain limitations, which open up space for further improvements. It is important to acknowledge the limitations of using the normalization. Min–max normalization is sensitive to extreme values, which can stretch the scale and compress most data into a narrower interval, potentially reducing variability that may be important for detecting subtle differences in edge strength. Furthermore, the final result depends on the choice of weights; If the weights are not well justified empirically or theoretically, the detected communities may reflect the assumptions of the researcher rather than the intrinsic structure of the network. In short, normalization and weighting together enhance methodological transparency and balance, but interpretation of the resulting communities should always take into account the assumptions underlying these preprocessing steps. In addition, destroying only two communities in the *Destroy and repair algorithm* gave good results in a very short time for graphs up to 200 nodes. However, for a graph of 300 nodes, the execution time increased by an order of magnitude (from 6.42 to 59.88 seconds). Subsequently, problems with 500 and 1000 nodes were solved to examine how the execution time further increases. The execution time of the heuristic for the graph of 500 nodes was 169.98 seconds, while for the graph with 1000 nodes it was about 40 minutes, which may indicate a computational problem for larger graphs.

Further research could examine system workload in the context of additional non-functional attributes such as scalability, availability, and reliability. Additionally, different linearization of the modularity function can be examined as well as different quality measures. In order to speed up the heuristic, the possibility of using other neighborhood structures than destroying only two communities should be examined. In addition, the efficiency of some other heuristics in solving the observed problem will be tested, primarily Variable neighborhood search.

References

1. Abeysinghe, A.: Cell-based architecture: A decentralized reference architecture for cloud-native applications. Available online: https://github.com/wso2/reference-architecture/blob/master/reference-architecture-cell-based.md (2024), accessed: 2024-05-17
2. Abgaz, Y., McCarren, A., Elger, P., Solan, D., Lapuz, N., Bivol, M., Jackson, G., Yilmaz, M., Buckley, J., Clarke, P.: Decomposition of monolith applications into microservices architectures: A systematic review. IEEE Transactions on Software Engineering 49(8), 4213–4242 (2023)

3. Ahmad, A., Babar, M.A.: A framework for architecture-driven migration of legacy systems to cloud-enabled software. In: Proceedings of the WICSA 2014 Companion Volume, pp. 1–8 (2014)
4. Aksakalli, I.K., Çelik, T., Can, A.B., Tekinerdoğan, B.: Deployment and communication patterns in microservice architectures: A systematic literature review. Journal of Systems and Software 180, 111014 (2021)
5. Alinezhad, E., Teimourpour, B., Sepehri, M.M., Kargari, M.: Community detection in attributed networks considering both structural and attribute similarities: two mathematical programming approaches. Neural Computing and Applications 32, 3203–3220 (2020)
6. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley Professional, 4th edn. (2021)
7. Bennett, L., Liu, S., Papageorgiou, L.G., Tsoka, S.: A mathematical programming approach to community structure detection in complex networks. In: Computer Aided Chemical Engineering, vol. 30, pp. 1387–1391. Elsevier (2012)
8. Besker, T., Martini, A., Bosch, J.: Managing architectural technical debt: A unified model and systematic literature review. Journal of Systems and Software 135, 1–16 (2018)
9. Bierska, A., Buhnova, B., Bangui, H.: An integrated checklist for architecture design of critical software systems. In: FedCSIS (Position Papers). pp. 133–140 (2022)
10. Blinowski, G., Ojdowska, A., Przybyłek, A.: Monolithic vs. microservice architecture: A performance and scalability evaluation. IEEE access 10, 20357–20374 (2022)
11. Bollinadi, M., Damera, V.K.: Cloud computing: security issues and research challenges. Journal of Network Communications and Emerging Technologies (JNCET) 7(11) (2017)
12. Bourque, P., Fairley, R.E.E.: Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press (2014)
13. Brenner, R.: Balancing resources and load: Eleven nontechnical phenomena that contribute to formation or persistence of technical debt. In: 2019 IEEE/ACM International Conference on Technical Debt (TechDebt). pp. 38–47. IEEE (2019)
14. Chandrasekar, A., Rajesh, S., Rajesh, P.: A research study on software quality attributes. International Journal of Scientific and Research Publications 4(1), 14–19 (2014)
15. Chen, R., Li, S., Li, Z.: From monolith to microservices: A dataflow-driven approach. In: 2017 24th Asia-Pacific Software Engineering Conference (APSEC). pp. 466–475. IEEE (2017)
16. Costa, A.R., Ralha, C.G.: Ac2cd: An actor–critic architecture for community detection in dynamic social networks. Knowledge-Based Systems 261, 110202 (2023)
17. Dao, V.L., Bothorel, C., Lenca, P.: Community structure: A comparative evaluation of community detection methods. Network Science 8(1), 1–41 (2020)
18. Eisty, N.U., Thiruvathukal, G.K., Carver, J.C.: A survey of software metric use in research software development. In: 2018 IEEE 14th international conference on e-Science (e-Science). pp. 212–222. IEEE (2018)
19. Ferdowsi, A., Chenary, M.D.: Toward an optimal solution to the network partitioning problem. In: 2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS). pp. 111–117. IEEE (2023)
20. Filippone, G., Mehmood, N.Q., Autili, M., Rossi, F., Tivoli, M.: From monolithic to microservice architecture: an automated approach based on graph clustering and combinatorial optimization. In: 2023 IEEE 20th International Conference on Software Architecture (ICSA). pp. 47–57. IEEE (2023)
21. Fortunato, S.: Community detection in graphs. Physics reports 486(3-5), 75–174 (2010)
22. Gaidels, E., Kirikova, M.: Service dependency graph analysis in microservice architecture. In: Perspectives in Business Informatics Research: 19th International Conference on Business Informatics Research, BIR 2020, Vienna, Austria, September 21–23, 2020, Proceedings 19. pp. 128–139. Springer (2020)

23. Gudivada, V., Apon, A., Ding, J.: Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. International Journal on Advances in Software 10(1), 1–20 (2017)
24. Hassan, S., Bahsoon, R.: Microservices and their design trade-offs: A self-adaptive roadmap. In: 2016 IEEE International Conference on Services Computing (SCC). pp. 813–818. IEEE (2016)
25. Hasselbring, W.: Software architecture: Past, present, future. The essence of software engineering pp. 169–184 (2018)
26. Hou, T., Yao, X., Gong, D.: Community detection in software ecosystem by comprehensively evaluating developer cooperation intensity. Information and Software Technology 130, 106451 (2021)
27. Huang, G., Zhang, P., Zhang, B., Yin, T., Ren, J.: The optimal community detection of software based on complex networks. International Journal of Modern Physics C 27(08), 1650085 (2016)
28. Ibidunmoye, O., Hernández-Rodriguez, F., Elmroth, E.: Performance anomaly detection and bottleneck identification. ACM Computing Surveys (CSUR) 48(1), 1–35 (2015)
29. ISO/IEC/IEEE: 24765:2017 systems and software engineering — vocabulary. Available online: https://www.iso.org (2017), accessed: 2024-05-27
30. Karataş, A., Şahin, S.: Application areas of community detection: A review. In: 2018 International congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT). pp. 65–70. IEEE (2018)
31. Kazanavičius, J., Mažeika, D.: Migrating legacy software to microservices architecture. In: 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream). pp. 1–5. IEEE (2019)
32. Keller Tesser, R., Borin, E.: Containers in hpc: a survey. The Journal of Supercomputing 79(5), 5759–5827 (2023)
33. Khadka, R., Batlajery, B.V., Saeidi, A.M., Jansen, S., Hage, J.: How do professionals perceive legacy systems and software modernization? In: Proceedings of the 36th International Conference on Software Engineering. pp. 36–47 (2014)
34. Kiran, S., Mohapatra, A., Swamy, R.: Experiences in performance testing of web applications with unified authentication platform using jmeter. In: 2015 International Symposium on Technology Management and Emerging Technologies (ISTMET). pp. 74–78. IEEE (aug 2015)
35. Kraft, S., Pacheco-Sanchez, S., Casale, G., Dawson, S.: Estimating service resource consumption from response time measurements. In: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools. pp. 1–10 (2009)
36. Li, D., Han, Y., Hu, J.: Complex network thinking in software engineering. In: 2008 International Conference on Computer Science and Software Engineering. vol. 1, pp. 264–268. IEEE (2008)
37. Li, Z., Shang, C., Wu, J., Li, Y.: Microservice extraction based on knowledge graph from monolithic applications. Information and Software Technology 150, 106992 (2022)
38. Lin, C.C., Kang, J.R., Chen, J.Y.: An integer programming approach and visual analysis for detecting hierarchical community structures in social networks. Information Sciences 299, 296–311 (2015)
39. Lin, S., Yu, J., Jiang, X.: Signalling overhead analysis of small data transmission for machine type communication. In: 2018 IEEE 18th International Conference on Communication Technology (ICCT). pp. 664–668. IEEE (2018)
40. Liu, S., Jung, E.S., Kettimuthu, R., Sun, X.H., Papka, M.: Towards optimizing large-scale data transfers with end-to-end integrity verification. In: 2016 IEEE International Conference on Big Data (Big Data). pp. 3002–3007. IEEE (2016)
41. Lu, Z., Dong, Z.: A gravitation-based hierarchical community detection algorithm for structuring supply chain network. International Journal of Computational Intelligence Systems 16(1), 110 (2023)

42. Mazlami, G., Cito, J., Leitner, P.: Extraction of microservices from monolithic software architectures. In: 2017 IEEE International Conference on Web Services (ICWS). pp. 524–531. IEEE (2017)
43. Menychtas, A., Santzaridou, C., Kousiouris, G., Varvarigou, T., Orue-Echevarria, L., Alonso, J., Gorronogoitia, J., Bruneliere, H., Strauss, O., Senkova, T., Pellens, B., Stuer, P.: Artist methodology and framework: A novel approach for the migration of legacy software on the cloud. In: 2013 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. pp. 424–431. IEEE (2013)
44. Milić, M., Makajić-Nikolić, D.: Optimization of the cell-based software architecture by applying the community detection approach. In: 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS). pp. 149–156. IEEE (2024)
45. Millett, S., Tune, N.: Patterns, principles, and practices of domain-driven design. John Wiley & Sons (2015)
46. Mohamed, E.M., Agouti, T., Tikniouine, A., El Adnani, M.: A comprehensive literature review on community detection: Approaches and applications. Procedia Computer Science 151, 295–302 (2019)
47. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Physical review E 69(26113), 1–16 (2004)
48. Nikolić, L., Dimitrieski, V., Čeliković, M.: An approach for supporting transparent acid transactions over heterogeneous data stores in microservice architectures. Computer Science and Information Systems 21(1), 167–202 (2024)
49. Pan, W.F., Jiang, B., Li, B.: Refactoring software packages via community detection in complex software networks. International Journal of Automation and Computing 10(2), 157–166 (2013)
50. Rabinovich, S.G.: Measurement errors and uncertainties: theory and practice. Springer Science & Business Media (2005)
51. Rashid, J., Mahmood, T., Nisar, M.W.: A study on software metrics and its impact on software quality. arXiv preprint arXiv:1905.12922 (2019)
52. Roa, Y.N.: Taking advantage of cell-based architectures to build resilient and fault-tolerant systems. In: Gancarz, R. (ed.) Cell-Based Architectures: How to Build Scalable and Resilient Systems. No. 113, InfoQ eMag (2024), https://www.infoq.com/minibooks/cell-based-architecture-2024/, accessed: March 26, 2025
53. Sellami, K., Saied, M.A., Ouni, A., Abdalkareem, R.: Combining static and dynamic analysis to decompose monolithic application into microservices. In: International Conference on Service-Oriented Computing. pp. 203–218. Springer (2022)
54. Serrano, B., Vidal, T.: Community detection in the stochastic block model by mixed integer programming. Pattern Recognition 152, 110487 (2024)
55. Services, A.W.: Reducing the scope of impact with cell-based architecture: Aws well-architected. Available online: https://docs.aws.amazon.com/wellarchitected/latest/reducing-scope-of-impact-with-cell-based-architecture/reducing-scope-of-impact-with-cell-based-architecture.html (2024), accessed: 2024-07-12
56. Sewak, M., Singh, S.: Winning in the era of serverless computing and function as a service. In: 2018 3rd International Conference for Convergence in Technology (I2CT). pp. 1–5. IEEE (2018)
57. Shahrad, M., Fonseca, R., Goiri, I., Chaudhry, G., Batum, P., Cooke, J., Laureano, E., Tresness, C., Russinovich, M., Bianchini, R.: Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. In: 2020 USENIX annual technical conference (USENIX ATC 20). pp. 205–218 (2020)
58. Singhal, N., Sakthivel, U., Raj, P.: Selection mechanism of micro-services orchestration vs. choreography. International Journal of Web & Semantic Technology (IJWesT) 10(1), 25 (2019)
59. Srinivasan, K., Devi, T.: Software metrics validation methodologies in software engineering. International Journal of Software Engineering & Applications 5(6), 87 (2014)

60. Šubelj, L., Bajec, M.: Community structure of complex software systems: Analysis and applications. Physica A: Statistical Mechanics and its Applications 390(16), 2968–2975 (2011)
61. Venters, C.C., Capilla, R., Betz, S., Penzenstadler, B., Crick, T., Crouch, S., Nakagawa, E.Y., Becker, C., Carrillo, C.: Software sustainability: Research and practice from a software architecture viewpoint. Journal of Systems and Software 138, 174–188 (2018)
62. Verdecchia, R., Kruchten, P., Lago, P., Malavolta, I.: Building and evaluating a theory of architectural technical debt in software-intensive systems. Journal of Systems and Software 176, 110925 (2021)
63. Viljoen, N.M., Joubert, J.W.: Supply chain micro-communities in urban areas. Journal of Transport Geography 74, 211–222 (2019)
64. Wang, Q., Kanemasa, Y., Li, J., Jayasinghe, D., Kawaba, M., Pu, C.: Response time reliability in cloud environments: an empirical study of n-tier applications at high resource utilization. In: 2012 IEEE 31st Symposium on Reliable Distributed Systems. pp. 378–383. IEEE (2012)

Appendix

Data Availability Statement. All relevant artifacts, including source code, heuristics, input data, and optimization results, are available at the following GitHub repository: https://github.com/milicm/optimizing-cell-based-software-architecture-community-detection.

Table 7. Effective interactions between software elements and software metric values after simulation

Metrics	Interactions								
	order -> legacy	payment -> notification	inventory -> production	product -> production	order -> notification	production -> product	production -> legacy	order -> product	production -> inventory
NoInt	15372	11322	67469	64425	7356	27886	33123	62122	47783
Total_ReqT	527979	1030516	616175	612443	694224	392293	511689	699353	596878
Total_ResT	22342	26597	19455	18254	12284	10053	12714	23151	14619
Total_ReqS	5053032	3902588	11267323	10758975	2462557	4835305	5862807	10523413	8362025
Total_ResS	5818935	3446692	11654157	11128212	1366186	8212586	50771373	18301326	13596598
Avg_ReqT	34.3467	91.0189	9.1327	9.5062	94.3752	14.0677	15.4481	11.2577	12.4914
Avg_ResT	1.45342	2.34914	0.28835	0.28334	1.66993	0.36050	0.38384	0.37267	0.30594
Norm_NoInt	0.13334	0.06597	1	0.94936	0	0.34152	0.42864	0.911054	0.67252
Norm_Avg_ReqT	0.29579	0.96063	0	0.00438	1	0.05789	0.07409	0.02493	0.039402
Norm_Avg_ResT	0.56640	1	0.00243	0	0.67121	0.03735	0.04865	0.04324	0.01094
Norm_Total_ReqS	0.29421	0.16355	1	0.94226	0	0.26948	0.38618	0.91551	0.67003
Norm_Total_ResS	0.09013	0.04211	0.20824	0.19759	0	0.13858	1	0.34278	0.24755
Communication Intensity (b_{ij})	0.276	0.4465	0.4421	0.4187	0.3342	0.169	0.3875	0.4475	0.3281

Table 8. First eight scenarios for the examination of the influence of weighting schemes on the community structure

Edges	equal	w1=1	w2=1	w3=1	w4=1	w5=1	w2,w3=0.5	w4,w5=0.5
(5, 7)	0.554	0.625	0.675	0.580	0.755	0.133	0.628	0.444
(7, 8)	0.415	1.000	0.116	0.167	0.099	0.695	0.141	0.397
(8, 4)	0.389	0.060	0.107	0.942	0.722	0.115	0.525	0.418
(4, 11)	0.741	0.999	0.681	0.946	0.600	0.477	0.814	0.539
(11, 2)	0.464	0.794	0.385	0.274	0.500	0.368	0.330	0.434
(2, 1)	0.558	0.714	0.555	0.933	0.196	0.393	0.744	0.295
(1, 5)	0.425	0.938	0.351	0.134	0.090	0.612	0.242	0.351
(8, 7)	0.704	0.251	0.807	0.980	0.754	0.729	0.893	0.742
(5, 1)	0.663	0.898	0.972	0.938	0.100	0.407	0.955	0.254
(6, 9)	0.487	0.205	0.775	0.830	0.415	0.210	0.803	0.313
(9, 10)	0.639	0.524	0.404	0.411	0.934	0.920	0.408	0.927
(10, 3)	0.503	0.693	0.181	0.427	0.575	0.638	0.304	0.606
(3, 12)	0.564	0.949	0.398	0.972	0.080	0.424	0.685	0.252
(12, 6)	0.649	0.759	0.680	0.543	0.378	0.884	0.611	0.631
(10, 9)	0.358	0.183	0.151	0.058	0.516	0.884	0.104	0.700
(9, 6)	0.611	0.812	0.815	0.306	0.189	0.933	0.561	0.561
(15, 16)	0.294	0.153	0.605	0.345	0.121	0.247	0.475	0.184
(16, 17)	0.280	0.175	0.266	0.288	0.376	0.297	0.277	0.336
(17, 13)	0.554	0.066	0.567	0.988	0.883	0.265	0.778	0.574
(13, 20)	0.361	0.153	0.518	0.478	0.154	0.504	0.498	0.329
(20, 19)	0.535	0.473	0.973	0.948	0.091	0.190	0.960	0.140
(19, 14)	0.494	0.346	0.626	0.713	0.655	0.128	0.670	0.392
(14, 15)	0.587	0.925	0.398	0.946	0.531	0.136	0.672	0.334
(20, 13)	0.761	0.918	0.691	0.795	0.969	0.431	0.743	0.700
(15, 14)	0.525	0.634	0.365	0.572	0.673	0.379	0.469	0.526
(18, 21)	0.641	0.106	0.987	0.830	0.902	0.379	0.908	0.641
(21, 18)	0.485	0.488	0.035	0.517	0.414	0.973	0.276	0.694
(23, 26)	0.385	0.778	0.205	0.005	0.253	0.683	0.105	0.468
(26, 28)	0.638	0.546	0.973	0.271	0.818	0.583	0.622	0.701
(28, 24)	0.767	0.916	0.591	0.924	0.584	0.819	0.757	0.702
(24, 30)	0.273	0.049	0.513	0.142	0.074	0.587	0.327	0.331
(30, 29)	0.272	0.238	0.469	0.095	0.238	0.323	0.282	0.280
(29, 22)	0.750	0.756	0.584	0.762	0.833	0.812	0.673	0.823
(22, 23)	0.617	0.673	0.103	0.759	0.797	0.754	0.431	0.775
(30, 24)	0.525	0.989	0.558	0.346	0.404	0.326	0.452	0.365
(22, 29)	0.463	0.759	0.235	0.289	0.541	0.491	0.262	0.516
(25, 27)	0.696	0.407	0.740	0.587	0.940	0.805	0.663	0.872
(27, 25)	0.539	0.599	0.318	0.782	0.003	0.996	0.550	0.500

Table 9. Last seven scenarios for the examination of the influence of weighting schemes on the community structure

Edges	w1=0	w2=0	w3=0	w4=0	w5=0	w2,w3=0	w4,w5=0
(5, 7)	0.373	0.537	0.441	0.238	0.598	0.501	0.606
(7, 8)	0.418	0.615	0.333	0.736	0.528	0.464	0.707
(8, 4)	0.403	0.482	0.555	0.615	0.703	0.631	0.452
(4, 11)	0.407	0.452	0.565	0.393	0.492	0.434	0.700
(11, 2)	0.431	0.548	0.596	0.531	0.859	0.726	0.331
(2, 1)	0.809	0.662	0.769	0.285	0.576	0.539	0.301
(1, 5)	0.705	0.686	0.370	0.492	0.471	0.343	0.574
(8, 7)	0.501	0.557	0.356	0.542	0.541	0.563	0.331
(5, 1)	0.368	0.542	0.494	0.707	0.166	0.291	0.590
(6, 9)	0.547	0.488	0.715	0.791	0.242	0.641	0.489
(9, 10)	0.260	0.474	0.340	0.668	0.458	0.432	0.487
(10, 3)	0.406	0.428	0.422	0.236	0.238	0.220	0.478
(3, 12)	0.256	0.478	0.367	0.466	0.348	0.814	0.732
(12, 6)	0.522	0.588	0.364	0.526	0.448	0.539	0.416
(10, 9)	0.323	0.752	0.499	0.393	0.346	0.633	0.509
(9, 6)	0.727	0.452	0.485	0.605	0.839	0.462	0.598
(15, 16)	0.473	0.512	0.450	0.306	0.771	0.452	0.906
(16, 17)	0.431	0.466	0.420	0.396	0.364	0.680	0.549
(17, 13)	0.423	0.410	0.543	0.343	0.738	0.400	0.531
(13, 20)	0.179	0.805	0.475	0.546	0.535	0.347	0.237
(20, 19)	0.207	0.121	0.812	0.349	0.652	0.491	0.686
(19, 14)	0.504	0.384	0.356	0.632	0.548	0.745	0.376
(14, 15)	0.663	0.699	0.508	0.315	0.470	0.715	0.423
(20, 13)	0.328	0.396	0.381	0.361	0.610	0.342	0.655
(15, 14)	0.472	0.417	0.504	0.700	0.409	0.461	0.435
(18, 21)	0.393	0.530	0.548	0.676	0.310	0.641	0.469
(21, 18)	0.405	0.440	0.617	0.406	0.630	0.593	0.617
(23, 26)	0.719	0.590	0.477	0.537	0.765	0.264	0.566
(26, 28)	0.601	0.747	0.535	0.237	0.481	0.481	0.777
(28, 24)	0.659	0.569	0.313	0.594	0.461	0.483	0.465
(24, 30)	0.352	0.346	0.641	0.484	0.584	0.880	0.689
(30, 29)	0.631	0.555	0.500	0.455	0.528	0.507	0.497
(29, 22)	0.513	0.584	0.177	0.324	0.340	0.134	0.670
(22, 23)	0.520	0.627	0.266	0.347	0.686	0.553	0.802
(30, 24)	0.475	0.444	0.639	0.587	0.329	0.597	0.694
(22, 29)	0.617	0.641	0.434	0.457	0.384	0.091	0.587
(25, 27)	0.304	0.550	0.482	0.446	0.759	0.605	0.563
(27, 25)	0.380	0.429	0.344	0.674	0.374	0.891	0.665

Miloš Milić is an Associate Professor at the University of Belgrade, Faculty of Organizational Sciences, Department of Software Engineering. His research and professional interests include software quality, software design, software architectures, design patterns, software engineering technologies, and software engineering education. He teaches undergraduate and graduate courses in his field. Miloš Milić serves as the President of the Committee for Standards and Related Documents in the fields of Software Engineering, IT in Education, and the Internet at the Institute for Standardization of Serbia.

Nebojša Nikolić is an Associate Professor at the University of Belgrade, Faculty of Organizational Sciences, Department of Mathematics. His research interests include applied mathematics, graph theory, combinatorial optimization, and metaheuristics. He teaches undergraduate courses in mathematics and PhD-level courses in combinatorial optimization and combinatorial algorithms. Nebojša Nikolić is the head of the Department of Mathematics and a member of the Mathematical Society of Serbia.

Dragana Makajić-Nikolić is a Full Professor at the University of Belgrade, Faculty of Organizational Sciences, Department of Operations Research and Statistics. Her research interests are related to mathematical modelling, optimization methods, business analytics, reliability, and risk assessment. She is the author or co-author of over 150 papers in international and national journals and conferences, several chapters in international monographs and encyclopaedias, and three books in the OR field. Dragana Makajić-Nikolić is the head of DOPIS – the Serbian Operational Research Society, and a member of UETS – the Union of Engineers and Technicians of Serbia, and the International Association of Engineers (IAENG).

Received: March 30, 2025; Accepted: November 10, 2025.

Computer Science and Information Systems 23(1):33–59 https://doi.org/10.2298/CSIS250401002K

Comparing Novice Programmers Performance with Block-based, Text-based, and Both Notations: A Study from Two Countries*

Tomaž Kosar[1], Srđa Bjeladinović[2], Dragana Ostojić[1], Milica S. Škembarević[2], Žiga Leber[1], Olga A. Jejić[2], Matej Moravec[1], Filip Furtula[2], Miloš D. Ljubisavljević[2], Ivan S. Luković[2], Marjan Mernik[1] and Matej Črepinšek[1]

[1] University of Maribor, Faculty of Electrical Engineering and Computer Science
Koroška cesta 46, Maribor, 2000 Maribor, Slovenia
{tomaz.kosar, dragana.ostojic, ziga.leber, matej.moravec, marjan.mernik, matej.crepinsek}@um.si
[2] University of Belgrade, Faculty of Organizational Sciences
Jove Ilića 154, Beograd, 11000, Belgrade, Serbia
{srdja.bjeladinovic, milica.skembarevic, olga.jejic, filip.furtula, milos.ljubisavljevic, ivan.lukovic}@fon.bg.ac.rs

Abstract. Teaching programming presents numerous challenges, one of which is selecting the most effective notation to introduce programming concepts to beginners. This study explores different notation approaches for learning fundamental programming concepts, with the objective of assessing how notation choice influences beginners' performance. To investigate this, we conducted a controlled experiment during short-term visits aimed at promoting programming in primary schools. Our multinational study divided participants into three groups: one using block-based notation (Poligot), one using text-based notation (Python), and one using both simultaneously (Poliglot). After completing a training session, the participants engaged in practical programming tasks to assess their performance in Python or the multiple-representation environment Poliglot. The results indicate that the choice of notation did not impact the performance of the participants significantly, as no statistically significant differences were found between the three groups. These findings were consistent across two different countries. Our study suggests that educators can use different notations (Python, Poliglot, or a combination) confidently when introducing programming to beginners. However, the performance results may improve when training sessions are extended over a longer period.

Keywords: Programming Education, Block-based Programming, Text-based Programming, Multiple-representation Environments, Programming Engagement, Novice Programmers.

1. Introduction

Introducing programming to children in primary school is crucial in today's digital world. Early exposure to programming not only equips children with valuable technical skills, but also enhances their complex problem-solving abilities [1, 2], creativity, communication, and teamwork skills. Despite its importance, informatics or programming subjects

* An extended version of The 19th Conference on Computer Science and Intelligence Systems/FedCSIS 2024 paper

are still not part of the primary school curriculum in many countries [3, 4]. This gap in primary school education can lead to an essential void in future job markets. By integrating programming into the curriculum [5], we can ensure that all children have the opportunity to develop these essential skills, preparing them for a future where technology, as we all agree, will play an important role in different aspects.

To address this deficit in primary school education we occasionally visit schools, spending time with the children demonstrating software development. During these visits, our aim is to engage the children by showcasing the development of simple applications and potential career opportunities in the field of Computer Science. We believe that, as guest lecturers in primary schools, we have several significant impacts. Guest lecturers can bring enthusiasm to the subject, which is different from the usual ones, making it more engaging for the children. The additional importance of guest lecturers is that they can pave the way for a creative approach to the adoption of basic programming by teachers in primary schools. Moreover, guest lecturers can introduce diverse perspectives and knowledge that might not be readily available through the standard curriculum, enriching the children's learning experience and highlighting the importance and relevance of programming skills in today's digital world.

However, teaching programming to beginners is a complex task that requires careful consideration of various factors. One of the key challenges is the limited time available to teach children programming during visits [6]. Another critical decision involves selecting the appropriate notation to introduce programming concepts [7, 8]. Using block-based notation (e.g., Scratch [9, 10], App Inventor [11]) allows the creation of visually appealing games and applications [12, 13]. Conversely, text-based notation enables focused teaching of fundamental programming concepts [14]. Each notation offers distinct advantages. We believe that block-based notation is the most inspiring way to introduce programming concepts, particularly for beginners. However, text-based notation is better suited for long-term learning, as it prepares students for professional programming, which relies primarily on textual coding. Early exposure to text-based syntax helps beginners build confidence and familiarity with industry-standard programming languages.

On the other hand, the transition from block-based programming to text-based programming is often highlighted as problematic; starting with block-based notation can lead novice programmers to be reluctant to switch to text-based notation [15]. This is a common pitfall that programming educators face. To address this problem, educational tools have emerged that enable educators to teach novice programmers to use both notations simultaneously [16, 17]. The most significant advantage of a multi-representational environment [18], which allows programming in two distinct notations, is that the transition from block-based to text-based notation occurs very naturally. For example, consider how time-consuming it is to write math equations using blocks; in contrast, text-based notation allows expressing math equations more naturally and efficiently. The strategy of presenting both notations simultaneously helps students integrate seamlessly and understand both forms of programming, easing the learning curve, and enhancing their overall comprehension. Using both notations simultaneously facilitates a smooth transition between the visual and textual programming, allowing students to leverage the strengths of each approach. This is especially beneficial for those who are already familiar with one notation but need to transition to the other. Although we developed one such multi-representational environment called Poliglot [18] that enables programming with both

notations simultaneously, we are still determining the advantages of using such an educational programming environment.

Regardless of the strengths of using each notation mentioned above, our motivation for this study arose from a practical need to evaluate whether the choice of programming notation has a meaningful impact on beginners' performance during short, introductory programming sessions. Specifically, our objective was to investigate whether using block-based, text-based, or both simultaneously (block and text) leads to statistically significant differences in participants' performance on basic programming tasks. To ensure a fair comparison, all the groups received the same amount of instructional and training time, although the participants using both notations were introduced to two notations simultaneously, potentially sacrificing performance. Furthermore, to examine how different educational contexts and prior programming exposure could influence the results, we conducted the study in two different countries as part of a broader multinational investigation. The following research questions guided our empirical study:

- RQ1: Does the type of programming notation (block-based, text-based, or using both simultaneously) influence programming performance significantly?
- RQ2: How does programming performance vary between two different national contexts?
- RQ3: How does the duration of programming training sessions affect performance outcomes, regardless of the notation used?

To test these research questions, we designed a controlled experiment [19–21], conducted in classrooms during short-term visits (2 hours), including training and a brief test at the end of the sessions. Additionally, in one country, we conducted sessions in 4 hours to answer the third research question. This multinational study was conducted in two different countries, providing a broad perspective on the performance of these notations. Our multinational study divided the participants into three groups: one using block-based notation, another using text-based notation, and the third using both. After training, the participants were assigned practical programming tasks.

This paper is an extended version of our previously published conference paper [22]. Although the core findings remain consistent, we have introduced substantial new content, including a clearer articulation of the motivation and explicitly defined research questions. Our multiple-representation environment, Poliglot, is now introduced more thoroughly, with additional usage examples to illustrate its capabilities. The experimental design and procedure are newly presented. In addition, a background study was added to the results, and the experiment has been repeated to obtain a larger sample size of participants.

The paper is structured as follows. The second section provides crucial insights into the background of the three different approaches used to teach beginners to program: block-based, text-based, and both simultaneously. The following section introduces the multi-representational environment Poliglot briefly, developed by one of the universities participating in this study. The fourth section reviews related studies. The fifth section illuminates the experimental design, goals, and data collection instruments. The sixth section presents the comparative results of the experimental study. The seventh section outlines and discusses the essential findings from the empirical investigation. Subsequently, the following section exposes critical threats to the validity of the results in this study. Finally, in the last section, conclusions are drawn regarding the research outcomes.

2. Background

As stated earlier, our experiment included three approaches to teaching programming. In this section, we introduce these alternatives and explain their dynamics and the potential of their use as an introduction to the programming world. Each of the tools presented was analyzed from the aspect of comprehensiveness to novice programmers, i.e., children who had never encountered programming concepts before.

Block-based notation

Block programming allows early-age students to become familiar with basic programming concepts [23] while strengthening programming logic by applying visual components. Dedicated editors provide visualization of programming constructs and learning through play. Pure block-based notation is still often used for children who encounter programming at the earliest age. Block-based notation eliminates certain logical or semantic types of errors that occur in text editors, since blocks can be combined according to clearly defined rules in advance. By adopting the rules of combining blocks, the students learn to eliminate inevitable mistakes spontaneously, contributing to faster acquisition of text-based notation.

Scratch [24] is an editor that allows one to learn block programming notation through play. It is based on a multi-panel, single-window setup, which provides transparency and clear visibility at all times. Every change is noticeable immediately, giving the impression that the program is "alive". Scratch supports hands-on, one of the main approaches to practicing coding [25], through the ease of testing each block and learning through changes and play. Although based on a block, Scratch can represent a reasonable basis for adopting a bottom-up approach to programming, but also for introducing students to extremely fine-grained programming [26].

Alice is another tool that enables active engagement during the process of learning programming skills [27]. As the authors stated, one of the major challenges for novice students in programming is "putting the pieces together". Alice provides 3-D visualization for solving different programming problems. The animated programming environment in 3-D enables the students to research further and develop algorithms for animating objects in an even more intuitive and interesting way than 2-D environments provide. Creating methods for objects and testing them in the dynamic 3-D environment can enhance the adoption of object-oriented programming using text-based notation.

To approach young generations and activate them, not only when they are at the computer in the classrooms, other systems can also be used, such as App Inventor [11]. This tool is available on mobile devices. By using mobile devices, the students can practice programming spontaneously and intuitively, even in moments of leisure. With this, through the game and constant availability, interest in programming can be accelerated less formally [26]. Using such systems, it is feasible to induce students to practice programming spontaneously in their spare time [28].

Text-based notation

Python is a renowned text-based language, a multi-purpose programming language that can be utilized on various platforms [29]. The simple and minimalistic text-based syntax

makes Python a convenient programming language for beginners, whereas various specialized modules that can be imported contribute to the versatility of this programming language. According to the data in [30], based on the number of Google searches for the tutorials, Python surpassed Java in 2018, and has been the most popular programming language ever since. Python enforces indentation as a way of separating nested blocks of code, which leads to a more visually intuitive way of reading and understanding the code (since indentation is considered a part of the syntax, and not just a recommendation in coding style, e.g., in Java). The role of parentheses is relatively reduced compared to Java or other object-oriented languages, where parentheses have the role of code separators and proprietorship indicators. Python is a high-level programming language with low-level machine instructions hidden from the developers, thus increasing comprehension and softening the learning curve. Another notable characteristic of Python is the dynamic assignment of variable type based on the given value of the variable, and there is no need for preemptive type declaration. Another benefit of the Python programming language is the possibility of functionality extensions that can be achieved with the addition to the program of predefined packages (modules). The separation of functionalities into modules contributes to the code's overall simplicity and reduces imports of unnecessary functionalities. In this way, the user does not need to be familiar with all functionalities at once, but can instead study module by module, depending on their needs.

Text-based is the default and established approach for learning programming, and because of that, we strive to compare the learning effects and outcomes of this notation to block-based and both notations' usage. In our research, we chose Python as a representative of text-based notation.

Both notations

Block-based languages are a popular way to introduce programming and create educational programming environments. However, users eventually need to transition to text-based notation to develop more complex programs. Significant efforts have been made to aid this transition through various methods, including presenting translated versions, dual-mode, multiple-representation, and hybrid environments. Examples of these environments include tools such as Tiled Grace [16], BlockPy [31], Pencil code [17], Droplet [32], Greenfoot [33], and Poliglot [18], all designed to address this challenge.

Let us introduce some of these environments briefly. Tiled Grace [16] is a tile-based editor for the Grace programming language. Using tiles enables visualization of the code, and there is also support for text editing. Textual editing of code expands the tiles' visualization, enabling a seamless change of the work environment and an easy transition between source code and visual representation of the code [16]. Droplet [32] is a library designed to create dual-mode environments. It translates code by inserting tags into the textual code, to indicate which parts will be represented as blocks. These blocks are then displayed by extracting the code between the tags and presenting it within the block structure. Tags are added using an external parser and precedence is handled by a custom JavaScript function that inserts parentheses into the blocks. The transition back to text-based notation is done by removing the tags while keeping the parentheses intact. On the other hand, BlockPy [31] is a web-based environment with open access targeting primarily novice programmers in Data Science. It uses Python for its text-based notation,

facilitating a smooth transition from block-based to text-based programming for beginners.

Poliglot [18], the tool used in this study, uses both notations simultaneously. For our research, it is essential to compare the benefits and limitations of using text-based, block-based, and both notations simultaneously to introduce programming to beginners. A detailed description of the Poliglot tool, including its design and functionalities, is provided in the following section.

3. Poliglot

An example of multiple-representation environments is also Poliglot[3] [18], developed by the Slovenian partners in this paper. Poliglot is an educational programming environment designed for beginners who are taking their first steps in programming, or who are already familiar with block-based programming and want to move forward and learn text programming. Our experiences teaching programming as guest lecturers in primary schools inspired the development. Previously, we used block-based languages such as Scratch and App Inventor, which engage children in programming effectively. These tools allowed users to create functional games, mobile applications, or the motion of a robot more quickly. However, when the capabilities of block-based languages were exhausted, a transition to text-based languages became necessary.

This transition posed a significant challenge for novice programmers. They had to start with the basics again, taking much longer to reach the level of complexity they had achieved with block-based languages. We frequently observed that this shift led to a loss of enthusiasm among novices. Poliglot addresses this issue by introducing both notations simultaneously from the outset. It helps beginners connect each block to its textual representation, easing the transition. By presenting both notations together, Poliglot blurs the boundaries between block-based and text-based programming. As beginners gain experience, they often find that expressing themselves in text-based notation becomes easier than using blocks. This transition occurs naturally within the Poliglot environment.

An example of the usage of the Poliglot system is shown in Figure 1. In this example, a user inputs a number and the system calculates the corresponding multiplication table. The children can choose to work in either block-based or text-based notation. When using the block-based interface, the equivalent text-based code is displayed simultaneously in the tool's top-right corner. Any changes made in the block-based environment are reflected instantly in the text-based code, and vice versa.

As discussed earlier in this paper, arithmetic or logical equations often encourage beginners to switch to text-based notation naturally within Poliglot's multiple-representation environment [18]. To facilitate this transition, we often design tasks that are inconvenient and time-consuming to perform using block-based notation. Consider the block-based program in Figure 2, where the task is to calculate the number of days that have passed in the current year. In such cases, the students typically prefer text-based notation over block-based notation, as writing the code is faster than dragging and dropping the necessary arithmetic equation blocks onto the workspace.

Regarding error reporting in Poliglot, the block-based view prevents most syntax errors; however, users may still write incomplete or incorrect code, which must be reported.

[3] https://poliglot.um.si/

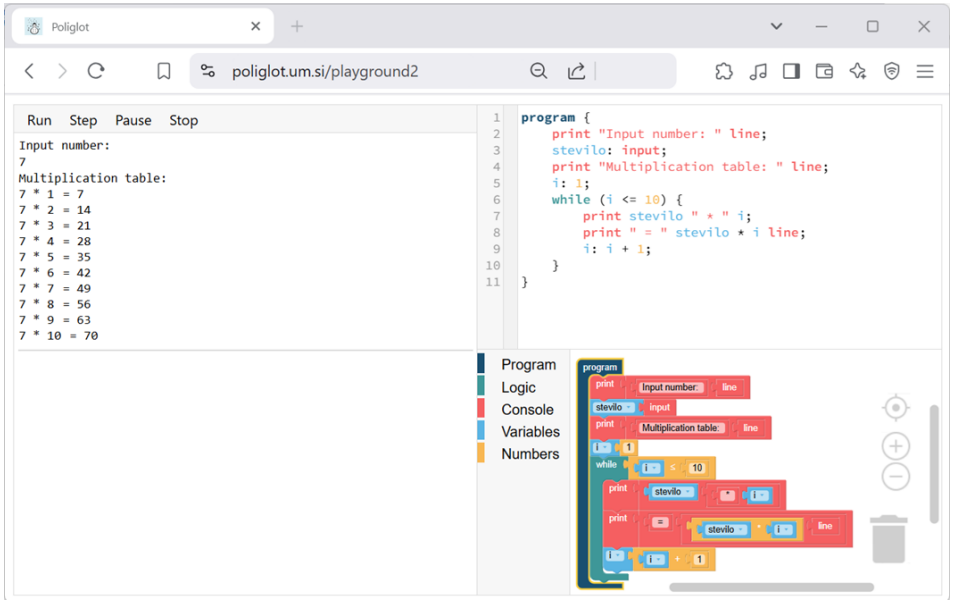


Fig. 1. Multiple-representation environment Poliglot

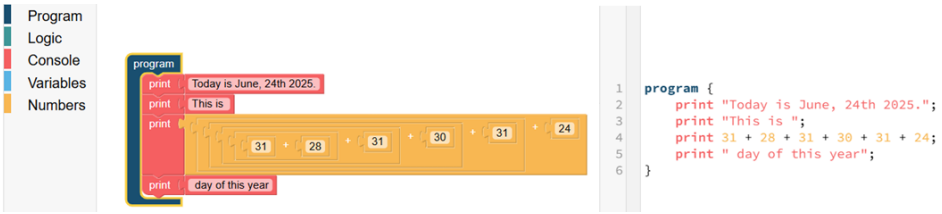


Fig. 2. Comparing arithmetic equations in block-based and text-based environments

When an error occurs within a block (such as a type mismatch in which a string is entered instead of an expected integer), the background of the text is highlighted. If the user attempts to confirm the incorrect value by pressing enter, the content reverts automatically to the last known valid state. In contrast, the textual view allows users to make any kind of error, just as in a standard text editor. Errors are detected and marked in the usual way, typically at the beginning of each row. A line containing an error will not be displayed in block-based notation.

However, once the line of code is corrected, it will appear immediately in the block-based notation, and vice versa when changing a block-based program in the text-based notation. In Poliglot simultaneous multiple representations are achieved using pretty-printing abstract syntax trees (AST), a standard task in language workbenches as described by Fowler in [34], and utilized in MPS [35]. In these programming environments, the end-user is not editing the code directly, but rather the AST, which is the model underlying the code. Programs can be understood as trees, a hierarchy of constructs that form the language behind the code. Each editor is merely one projection of the same model, and a projectional editor can have multiple projections, or representations, of the same code.

In this context, Poliglot offers two projections: a block-based editor and a text-based editor [18].

We noticed that after using both notations for a certain period, Poliglot users tend to prefer the text-based notation and recognize the advantages of text-based programming. However, we have always been interested in whether using both notations simultaneously improves their understanding of programming concepts and how it affects their programming performance.

Note that we do not favor Poliglot as a multi-representation environment. Instead, we encourage other researchers to conduct similar experiments using comparable tools, such as Grace [16], BlockPy [31], Pencil Code [17], Droplet [32], Greenfoot [33], etc.

4. Related work

The authors in [36] performed a quasi-experimental study investigating how modality (block-based and text-based environment) impacts high school Computer Science students, by conducting two classes at the same school through the same curriculum and the same teacher using either the block-based or text-based programming environment (the Pencil.cc environment was used, which supports both modalities, but the students were able to use only one modality). The outcome of this study [36] shows that the students' conceptual knowledge had been improved in both groups. However, the students using a block-based environment showed significant learning gains, as well as a higher attitude toward future programming courses. On the other hand, no difference was found in both groups with respect to confidence and enjoyment. The study [36] was extended in [8] by a third group using a hybrid block/text environment, with the main goal of how modality (block-based, text-based, hybrid block/text) influences programming practices (e.g., the number of runs, patterns in novice's help-seeking behavior). While the authors didn't find hybrid block/text modality superior, they did find some new programming practices. However, cognitive and attitudinal outcomes have not been measured or discussed. This work was later extended in [37], where the authors checked the hypothesis that gains in attitudinal and conceptual learning using a block-based environment would transfer to a conventional text-based programming language (Java). The study showed that, whilst the students had a greater conceptual learning gain using a block-based environment, this was not transferred to the environment using the professional Java programming language. Furthermore, no difference in programming practices or attitudinal shifts was found between both groups. As such, this study [37], is important, to show the limitations of block-based programming. Our study extends this one [36], and brings additional evidence in this field.

In Portugal and Serbia, a software tool for visualizing program units called C Tutor was used in introductory programming courses, and the study showed little initial association between the tool used and the test results [19]. However, the research has shown a correlation between the repeated usage of C Tutor and the achieved results via controlled practice outside the course, especially among high-achieving students. As a result, the authors have confirmed the hypothesis that "C Tutor has some positive impact on student progress in programming" [19]. The study [38] tried to answer the difference between block-based and text-based environments on novice Computer Science students' cognitive (knowledge, comprehension, application, analysis, synthesis, evaluation) and

attitudinal (satisfaction, confidence, motivation, appreciation, enthusiasm) outcomes by performing a meta-analysis, which showed that block-based environments had a small effect on cognitive outcomes, and only a trivial effect on attitudinal outcomes.

Students' difficulties in the transition from block-based to a text-based environment have been discussed in [39], where it was shown that the students struggled to solve a new coding challenge in a text-based environment due to difficulties of one or more of the following aspects: readability, memorization of commands, memorization of syntax, native language of programming, typing/spelling and writing expressions.

The authors in [40] presented a Systematic Literature Review (SLR) [41] on the characteristics of block-based environments, and how block-based environments support beginners in the transition to text-based programming, where the following distinct approaches were identified: Blocks-only, One-way Transition, Dual-modality, and Hybrid. Among those, dual-modality programming environments are the most effective for supporting students' transitioning to text-based programming.

5. Experiment design

The primary objective of this study is to examine the outcomes of assessments when beginners engage in programming using three different types of programming: block-based, text-based, and both simultaneously. By comparing the results of these approaches, the study seeks to elucidate the learning outcomes associated with teaching beginners using different programming notations.

According to the research questions defined in the introduction, the following central hypothesis is defined:

- $H1_{\text{null}}$: There are no significant differences in short-term programming performance between participants using different programming notations (block-based, text-based, or both simultaneously).
- $H1_{\text{alt}}$: There is a significant difference in short-term programming performance between participants using different programming notations (block-based, text-based, or both simultaneously).

This hypothesis will be examined from two additional perspectives that may influence the primary outcome:

- *National context*: a comparison of results between two countries: Slovenia and Serbia, and
- *Duration of programming instruction*: specifically, comparing two hours versus four hours of training.

5.1. Experiment procedure

The experiment was carried out following a clearly defined procedure (see Figure 3), consisting of several key phases, presented schematically to facilitate a better understanding of the overall process. The schematic representation illustrates the sequence of activities

within the experiment clearly, including the administration of an initial participants' background survey, the implementation of programming training with different notations, and the final assessment phase.

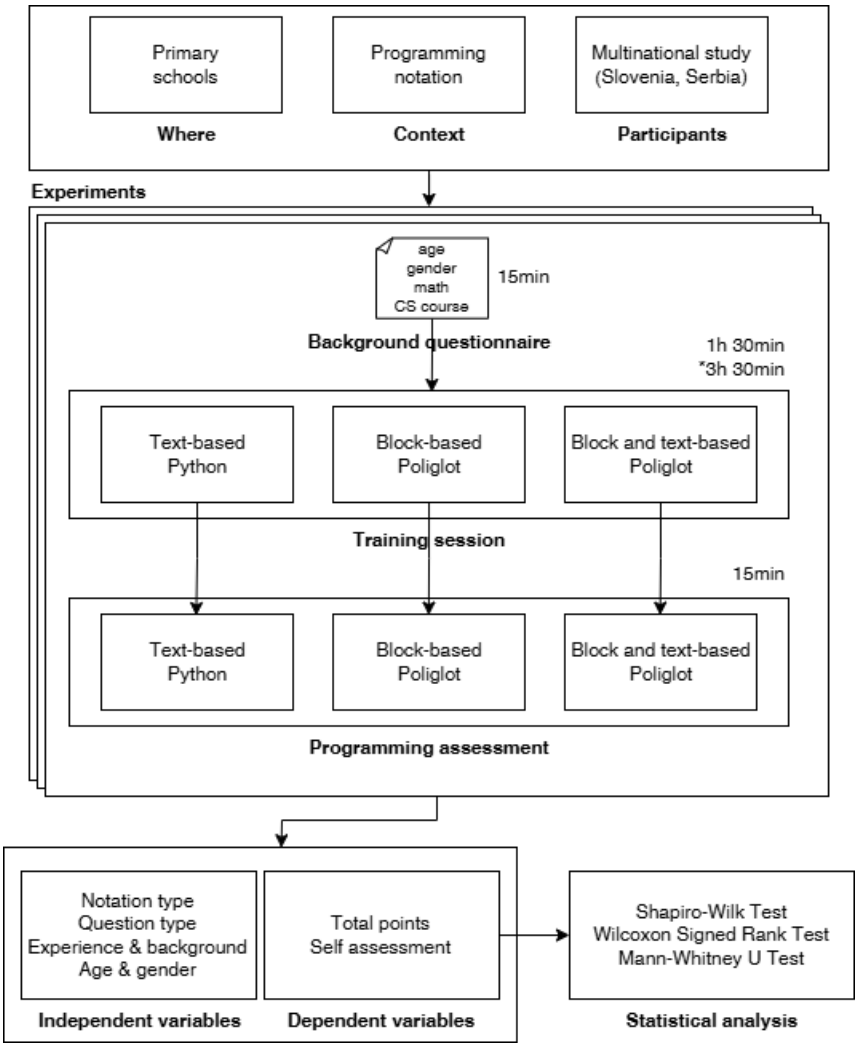


Fig. 3. Experiment procedure

The experiment was conducted in person in primary schools in Serbia and Slovenia. The participants were primary school students divided into three groups based on the notation used during the training. The first group worked with block-based notation in Poliglot, the second group used text-based notation in the Python programming language, while the third group was taught using a combination of both notations simultaneously in

Poliglot. This approach enabled a comparative analysis of the differences in understanding fundamental programming concepts among groups exposed to different instructional methods.

To ensure the consistency and validity of the results, each phase of the experiment was clearly time-defined. At the beginning of the experiment, the participants completed an initial questionnaire [4] lasting 15 minutes, which collected data on their prior knowledge and experience with computers, as well as their inclination toward solving logical and mathematical tasks. Following this, the main phase of the experiment (programming training) took place. This phase lasted 90 minutes for the participants in Serbia, while, in Slovenia, it lasted 90 minutes for one group and 210 minutes for two groups. During this phase, the participants were introduced to the theoretical foundations of programming concepts, then worked with the instructor on solving practical tasks, and, finally, solved tasks independently on computers to reinforce their acquired knowledge further.

The experiment ended with a final test lasting 15 minutes, which contained identical tasks adapted to the type of notation each group had practiced. This test evaluated the participants' programming performance. Figure 3 provides a clear representation of the sequence of these phases, offering insight into the process of conducting the experiment and the structure of activities used to investigate the impact of different types of notations on participants' success in understanding fundamental programming concepts.

5.2. Participants

The experiment was multinational and multi-institutional, involving participants from two different countries: 236 from Serbia and 155 from Slovenia. The participants were primary school children aged between 11 and 14 years. The participants were sixth, seventh, and eighth-grade primary school students. No prior participant selection was conducted for this study, resulting in the inclusion of participants with diverse backgrounds, varying levels of knowledge and experience in programming, and differing levels of interest in the subject. The assessment of previous knowledge and experience was self-assessed, supplemented by their grades the participants had at the end of a previous school year.

Multiple iterations of the study were conducted in both countries. For each new iteration, a different programming notation was used: block-based, text-based, and, finally, a combination of both notations. During the training sessions, all the educators utilized the same PowerPoint slides, which included explanations of the concepts, tasks, and correct program examples. This approach ensured consistency in the training provided by different educators. Prior to the main experiment, pilot studies were conducted, to refine the background questionnaire, training materials, and the final test. Each question in the test offered five potential answers, with only one being correct.

5.3. Data collection instruments

The tests that were handed out after the lectures consisted of the same set of questions written in the corresponding notation based on the materials that were presented during the lectures (block-based, text-based, or both simultaneously). The questions appeared in ascending order according to their difficulty. There were five question types:

[4] https://github.com/tomazkosar/PoliglotStudyOnNotations

1. Prediction of the given code execution: The participants were presented with a few lines of code with options on what the result of the execution of that code would be.
2. Finding a redundant piece of code: Based on the given code block, the participants were asked to identify a redundant line of code that did not influence the program's execution.
3. Code insertion: A block of code was presented with one line missing; the participants were expected to select the line that would complete the block of code and provide a logical solution to the problem.
4. Identification of logical errors in the code: The block of code was shown with a notice stating that there was a logical error in the code that needed to be identified.
5. Code modification: The last task required the participants to change a line of code, thus changing the result of code execution to match the description of the desired code behavior.

Even though the tests were written in different notations, the logic behind the question remained the same, without any changes to the formulation of the question except the syntax. Figure 4 shows a question from a block-based test. In this question, the participants were asked about the result of running the block-based program. This question is an example of the "Prediction of the given code execution" question type. Note that half of the questions on the test for both notations were written in the text-based notation, and the other half of the questions were written using the block-based notation.

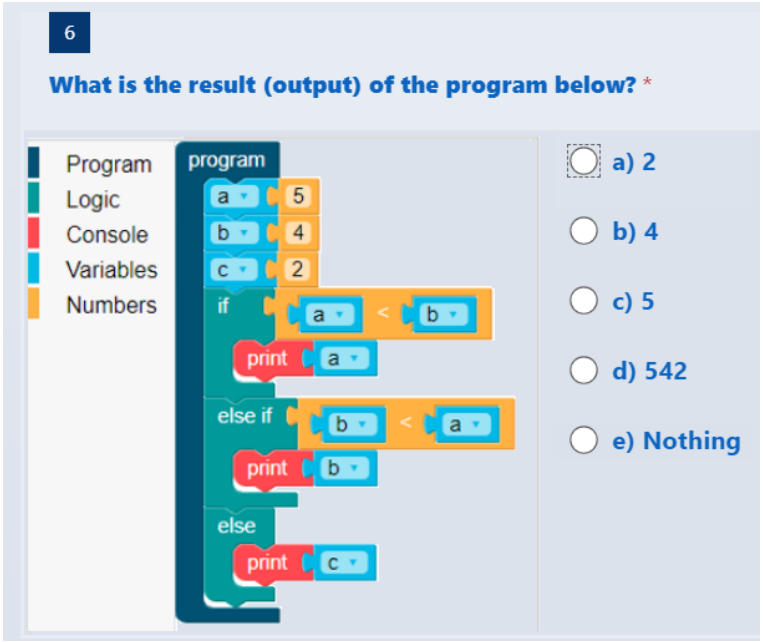


Fig. 4. Question from the block-based test

The tests were prepared in two languages, Slovenian and Serbian, allowing the participants to solve tasks in their native language. The points awarded to each question were also the same (each answer was worth 1 point) to make the questions comparable.

6. Results

This section compares the participants' performance from two different countries (Serbia is referred to as country 1 and Slovenia is referred to as country 2) with three different alternatives to teach programming. It also presents the results of the background questionnaires. All the observations were tested statistically with a α = .05 as a threshold for judging the significance.

6.1. Background study

To minimize potential bias arising from differences among participants, we compared their backgrounds across different executions, separated by country and the notation used. This comparison included previous programming experience (0 - no experience, 1 - experiences in programming), whether they had taken a mandatory or elective informatics course (0 - no informatics subject so far, 1 - had a subject in Informatics), mathematics and informatics grade (grades are between 1 and 5).

In Table 1, we present the results of a single execution, specifically from country 1 (Serbia), where we compared block-based and text-based outcomes (see Subsection 6.2).

Since the experiment followed a between-subjects design and the data did not meet the normality assumption (as determined by the Shapiro-Wilk normality test), we used the Mann-Whitney U test to compare the two groups. The results, shown in Table 1, do not indicate significant differences between the block-based and text-based groups in terms of programming experience, selection of the informatics course, or grades in math and informatics (the p-value is above 0.05 in the last column of Table 1). These findings suggest that mathematical background, programming experience, or prior informatics education are unlikely to have influenced the comparison of block-based and text-based results in Serbia.

For brevity, the results from other executions are omitted, but statistical analyses yielded similar findings. Further discussion on background results is presented in Section 7.

Table 1. Participants' background: block-based vs. text-based (Mann-Whitney U Test) – country 1

	Group	Mean	N	Std. Dev.	Median	Mean Rank	Z	p-value
Programming experience	Group I (block-based)	0.73	52	0.448	1.0	57.75	-0.477	0.634
	Group II (text-based)	0.77	65	0.425	1.0	60.00		
Mandatory Informatics	Group I (block-based)	1.00	52	0.000	1.0	59.00	-0.000	1.000
	Group II (text-based)	1.00	65	0.000	1.0	59.00		
Math grade	Group I (block-based)	3.75	52	1.118	4.0	61.32	-0.686	0.493
	Group II (text-based)	3.62	65	1.041	4.0	57.15		
Informatics grade	Group I (block-based)	4.94	52	0.308	5.0	60.69	-1.104	0.270
	Group II (text-based)	4.91	65	0.292	5.0	57.65		

6.2. Comparative analysis: two-hour executions

The following subsection presents the results of two-hour executions comparing different notations. To minimize potential biases related to cultural differences, environmental factors, and variations in primary school curricula, the results are divided by country.

Block-based vs. text-based notation results: Country 1

The first comparison examined the test results of the Serbian participants (country 1) who attended lectures and took the tests in block-based notation (Poliglot), versus those who did so in text-based notation (Python). After data cleansing, there were 52 valid responses for the block-based Poliglot test and 65 valid responses for the text-based Python test. Some tests could not be paired with their initial counterparts, and were therefore discarded. Additionally, instances of double submissions by the same participant reduced the number of tests considered for further analysis.

The Shapiro-Wilk test was used to check for normal distribution. Since the data deviated from a normal distribution, the non-parametric Mann-Whitney U test was employed to compare the two independent samples. The slight difference in mean scores (see Table 2) between the two groups was not statistically significant (p-value = 0.621). Therefore, no conclusive differences in performance outcomes could be established between the block-based and text-based groups.

Table 2. Performance results: block-based vs. text-based (Mann-Whitney U Test) – country 1

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	p-value
Group I (block-based)	45.67	52	22.40	50.00	57.29	-0.495	0.621
Group II (text-based)	48.08	65	23.72	50.00	60.37		

Block-based vs. text-based notation results: Country 2

To verify the consistency of the results obtained in Serbia a similar between-subjects study was conducted in Slovenia (country 2). Table 3 presents the performance results, measured as the percentage of correct responses, to assess the programming performance after training. Both Group I (block-based) and Group II (text-based) completed an equal number of tasks with identical question types and complexity. An examination of Table 3 reveals that the block-based group outperformed the text-based group, as indicated by the mean scores (34.19% vs. 32.92%).

Table 3. Performance results: block-based vs. text-based (Mann-Whitney U Test) – country 2

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	p-value
Group I (block-based)	34.19	39	24.47	33.33	40.73	-0.287	0.774
Group II (text-based)	32.92	40	21.84	33.33	39.29		

Once again, the data deviated from a normal distribution, necessitating the non-parametric Mann-Whitney U test to compare the two independent samples. Despite the observed difference in mean scores between the two groups, this difference was not statistically significant (p-value = 0.774). These results are consistent with our findings from the initial study conducted in Serbia – we could not confirm any statistical differences between these two groups in Slovenia.

A comparison of the results in Table 2 and Table 3 reveals that the Serbian participants performed better. This is likely attributed to differences in the participants' backgrounds, particularly the mandatory informatics curriculum in country 1. However, a direct comparison between countries is outside the scope of this study.

These results are aligned with those obtained in the study by Weintrop et al. [8] as discussed in the Related Work section.

Both notations vs. text-based notation results: Country 1

The second comparison in this research examined the use of both notations simultaneously (Poliglot) versus a text-based notation (Python). A total of 39 participants took the test using Poliglot (see Table 4). Among them, some participants did not answer any questions correctly and none achieved the maximum score. Data for the text-based environment (a total of 65 participants) were the same as shown in Table 2, but were then compared statistically with the data collected from the participants who experienced both notations simultaneously side by side during their training.

The average score on the test with both notations was the highest of all three groups, at 49.04%. The Standard Deviation for this test was with a value of 23.71%. The variance matched that of the text-based test, which was 23.72%. Once again, the difference between the groups was not statistically significant (p-value = 0.775), as determined by the Mann-Whitney U statistical test.

Table 4. Performance results: Both notations vs. text-based (Mann-Whitney U Test) – country 1

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	p-value
Group I (both notations)	49.04	39	23.71	50.00	53.83	-0.286	0.775
Group II (text-based)	48.08	65	23.72	50.00	51.85		

6.3. Comparative analysis: four-hour executions

One potential limitation that may have influenced the results of previous executions is the duration of the training sessions. To address this, in country 2 (Slovenia), we extended the training sessions to four hours for each treatment. This extension allowed the participants more time for practice, reinforcing key concepts through repetition, and enabling them to develop their solutions independently without the pressure of time constraints.

Block-based vs. text-based notation results: Country 2

Table 5 presents the performance results from the block-based and text-based notations. Both Group I (using block-based notation) and Group II (using text-based notation) demonstrated improved performance compared to the two-hour executions (e.g., see the Mean Column in Table 3). When comparing the block-based notation approach with the text-based approach, the participants achieved almost identical scores (67.26% vs 67.13%). Therefore, it was obvious that the performance difference is not statistically significant (p-value = 0.949).

Table 5. Performance results: block-based vs. text-based (Mann-Whitney U Test) – country 2

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	p-value
Group I (block-based)	67.26	21	22.53	75.00	24.64	-0.064	0.949
Group II (text-based)	67.13	27	17.73	75.00	24.39		

Both notations vs. block-based notation results: Country 2

In this experiment, we compared the performance outcomes between a group using both notations (block-based and text-based) and a group using only block-based notation. It is important to note that both treatments in this comparison were performed using the Poliglot system [18], which supports programming instruction in block-based notation alone, or in both notations simultaneously.

Table 6 presents the comparison of the performance results between the use of both notations simultaneously and the block-based notation. Both groups completed the same set of tasks. As shown in Table 6, the block-based group outperformed both notations groups, as reflected in the mean scores (67.26% vs. 60.27%). However, this difference was not statistically significant.

Table 6. Performance results: Both notations vs. block-based (Mann-Whitney U Test) – country 2

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	p-value
Group I (both notations)	60.27	28	19.56	62.50	23.16	-1.061	0.289
Group II (block-based)	67.26	21	22.53	75.00	27.45		

When comparing the block-based notation results in Tables 6 and 3, a noticeable difference can be observed in the mean scores. Since both executions were conducted under identical conditions – using the same instructors, teaching materials, and tests – we can conclude that the extended training session was the primary factor contributing to the significantly improved results in this execution. The additional time allowed for more practice and repetitions likely improved the performance of the participants.

6.4. Discovering learning patterns through association rules

In order to test the hypothesis that there are differences in performance and learning outcomes when different notations are used during the learning process, the Apriori algorithm was applied to the data retrieved from the entry tests as well as the exit test results. The Apriori algorithm is a technique for discovering hidden patterns during the data mining process. A definition of the association rule is given in [42]: association rule implies hidden association relationships that can be found among a set of objects. The algorithm has two steps: finding frequent item sets (the threshold for statistically relevant frequency is a predefined parameter - support) and generating an association rule based on the most common itemsets. The rule is derived based on the probability of occurrence of one item in the itemset if another item is present.

In this case, the algorithm was implemented in Python, leveraging the pandas library for preprocessing data, the mlxtend library for item set extraction, and matplotlib and seaborn for result visualization (heatmaps).

Before importing data to the dataframes, the data were cleaned of invalid responses, textual answers were converted to numerical values, and answers with a wide range of numbers were converted to discrete intervals, as the algorithm requires categories between which it searches for the correlations.

The first aim was to test if there was a correlation between the test type and the total number of points. The maximum number of points on the test was 8, and the minimum was 0. Based on the number of points, the rows were divided into three categories: low (0-2), medium (3-5), and high (6-8). The support was set to 0.1, as well as the minimum threshold. For country 1 (Serbia), the heatmap showed that there is no significant correlation between the score and the test type (the correlation coefficient between the category for low score and the block-based, text-based and both notations test were 0.46, 0.48 and 0.49 respectively) for the two-hour training sessions.

With the same preconditions (algorithm parameters and the duration of the training session), the data for country 2 (Slovenia) showed a strong correlation between medium performance and the use of both notations simultaneously (0.73), and a lower correlation was observed between high performance and the tests in block-based notation (0.53).

In country 2 (Slovenia), there was also an additional experiment setup where the training classes lasted for four hours instead of two. By applying the same algorithm, the results showed a strong correlation between the medium performance and the use of the both notations simultaneously test (0.73), alongside a high performance and block-based notation (0.58).

The next step was designed to test whether there is a correlation between test type and the score on the questions related to loops. The support and threshold parameters remained the same, but the category for points was reduced to low and high. For country 1 (Serbia), for classes that had a training session of two hours, there was an almost identical percentage for each test type and category (0.44, 0.57, and 0.51 for high scores in relation to block-based, text-based, and the use of both notations simultaneously tests, respectively) which leads to the conclusion that neither of the notations is better suited for learning about loops in programming.

Similarly to the first step, the same preconditions were applied during the testing in country 2 (Slovenia). The data were given as input to the algorithm with the same parameters, and the heatmap showed a strong correlation with the block-based notation and a

high score (0.71), and the use of both notations simultaneously tests and high score (0.64). Interestingly, the same results were obtained for the training sessions that lasted 4 hours.

Based on the results of the Apriori algorithm, the final conclusion for country 1 (Serbia), derived from the data collected during the two-hour training sessions, is that there is no evidence to support the existence of a correlation between the performance on the tests and the notation taught to participants. However, the results from country 2 (Slovenia) show that the participants performed better on tests that are given in both notations simultaneously and have both block and text elements.

7. Discussion

To understand the outcomes of our controlled experiment better, we present the results of a study utilizing a background questionnaire in subsection 6.1, focusing particularly on the Informatics and Mathematical backgrounds of the participants involved.

Based on the data analyzed in sections 6.1 and 6.2. two groups of between-subject tests were conducted in country 1 (Serbia). The first one was block-based vs. text-based. We cannot confirm the differences in the results statistically (see the results in Table 2, again). The fact that each participant achieved at least 1 point speaks in favor of the block-based notation, which was not the case with the text-based notation, because there were two participants with 0 points in the sample.

In favor of the uniformity of the useful prior knowledge of the participants in the two tested groups, the almost insignificant differences in the average grade in Informatics that the participants of both tested groups had (4.94 participants who did the block-based notation, 4.91 participants who did text-based notation as presented in Table 1), as well as the successfulness of resolving three logical tasks given in the input survey (the participants who did the block-based notation averaged 1.96 points from 3, while the participants who did the text-based notation had 1.97 on average). A slightly higher average grade in Mathematics was present in the participants who did the block-based notation notation (3.75) compared to the participants who did only the text-based notation (3.62). However, the average score was on the side of text-based notation, so it can be concluded that prior knowledge of Mathematics and Informatics was not crucial for the achieved result in programming in the tested groups.

The second group of between-subject tests conducted in country 1 (Serbia) was both notations simultaneously vs. text-based. Identical values of Standard Deviation and variance (see Table 4, again) in both approaches suggest the uniformity of participants within the sample, although the number of participants involved in the execution with both notations was the smallest of all three tested groups. Furthermore, the obtained results suggested a slight advantage of both notations shown simultaneously compared to the textual one, which was represented by a slightly higher average, but also by the values of the first and third quartiles, while the fourth belonged to the textual notation, because there was no maximum number of points in the execution with both notations shown simultaneously. Based on the sample, it can be concluded that the participants using block-based notation achieved better performance more easily, with lower and medium performance on the test. In contrast, for maximum performance, textual notation still had an edge. All of the above leads to the conclusion that Poliglot achieves an advantage by using both notations, and improves the average performance compared to exclusively block-based notation. Similar

to the previously analyzed group of tests (block-based vs. text-based), when comparing the groups of participants who did both notations vs. textual notation, it can be concluded that the average marks in Informatics (an average of 4.87 for the participants with both notations vs 4.94 for the participants with textual notation) and Mathematics (3.64 was the average grade for the participants learning both notations vs 3.67 for the participants learning textual notation) were quite uniform. A slight difference in the average number of points on the logical tasks of the input survey was evident in favor of the participants who did text-based notation (average 1.97 from 3 points vs average 1.77 from 3 points). Despite this, the participants with both notations achieved the best results of average points on the output of all three analyzed groups in Serbia, which can lead to the conclusion that, by applying both notations, there is scope for achieving better performance, even for those participants who are closer to block-based notation (e.g. who have experience in using Scratch or some other environments), but also among participants who are closer to textual notation, precisely because of the possibility of choosing a notation that is more convenient for them.

We evaluated our hypothesis and the results suggest that the type of programming notation (block-based, text-based, or both simultaneously) does not influence primary school participants' performance significantly in short-term introductory programming lessons. This supports the idea that educators are flexible in the decision of a programming environment to use for beginners, and this does not affect the immediate learning outcomes. The results are consistent with other similar studies [8]. Notably, even in the use of both notations simultaneously, where the participants had to use both block and text-based notation, the performance remained statistically comparable to the other groups. This implies that introducing two notations simultaneously does not affect short-term understanding, although it may impose a higher cognitive load.

Another key result highlights the positive effect of extended instructional time, regardless of the notation used. The participants who received four hours of training outperformed those who had only two hours drastically, suggesting that time investment plays a more crucial role than the choice of a notation in shaping beginner programming outcomes.

The study results were received from two participating countries, and this multinational aspect strengthens the generalizability of these results. They indicate that foundational programming skills can be taught effectively using any of the three notation types, across different cultural or educational contexts, provided that sufficient instructional time is allocated.

8. Threats to validity

This section discusses the construct, internal, and external validity threats [43] associated with our experiment.

8.1. Construct validity

In our experiment, our objective was to measure the effect of notation on test results. The participants were assessed with multiple-choice questions after two or four hours of training in specific notations: block-based, text-based, and both simultaneously. The use

of multiple-choice questions may have influenced the results. Different outcomes might have emerged if we had used code implementation instead.

The test for both notations included half of the questions in block and half in text-based notation. This may have introduced a threat to construct validity. We do not know the outcome of the test if each question had contained programs in both notations. As a result, the test may not measure the participants' conceptual knowledge accurately across both notations equally, potentially skewing the assessment results.

Another potential threat to the construct validity is the complexity of the questions. The test results in two-hour executions are generally low, with a performance average 50% or less. The outcomes of our experiments might differ if the question complexity were reduced.

The training sessions were limited to two hours, as requested by the primary schools. This constraint might have influenced our results significantly. To investigate this, we conducted executions, during which the training duration was extended to four hours. This extended training included additional functionality and repetitions of mastering the same programming concept. Consequently, we observed higher performance results (see Table 5). However, these improvements were observed in both groups, and the differences in the results were not statistically significant.

We used Poliglot for block-based and multiple-representation environments, although alternative tools exist for using both notations simultaneously (e.g., BlockPy [31]). Our experiments did not include the usage of these tools, nor did our study cover other textual languages beyond Python. Consequently, our findings are specific to the particular combination of Python and Poliglot and should not be interpreted as generalized findings, valid for all block and text-based notations. Different text-based programming languages (e.g., Java, Smalltalk, Rust, or C++) and block-based environments (e.g., Scratch, BlockPy) may yield different performance outcomes.

8.2. Internal validity

One potential threat to internal validity is the quality of instruction provided to novice programmers during our experiment. Although we standardized training materials (presentations, code snippets, etc.), the use of different lecturers may have influenced the outcomes of our experiment.

The sample size may have influenced the results of the study. To enhance the reliability and validity of these findings, it is crucial to perform multiple repetitions of the experiment with a larger sample size. This will help mitigate potential biases and provide a more robust understanding of the observed effects. Despite a relatively small sample size, the extended training period within that study demonstrated a positive influence on the results clearly.

Although all the groups received the same amount of instructional and training time, the participants using both notations were required to learn and apply two notations simultaneously in the same time slot as other groups using only the block or text-based notation. This likely imposed a higher cognitive load compared to the single-notation groups. As a result, the differences in performance outcomes between the groups may be attributed to unequal cognitive demands rather than the type of notation itself, which threatens the internal validity of the study.

The limitation of association rules analysis stems from the fact that the dataset has fewer than 400 records, and the number of participants who have taken each test is not equal. There is also a variation in the duration of the classes and the time that was dedicated to the participants practicing programming in the environment, which adds to the variation in results.

8.3. External validity

The specific context and settings of our experiment might influence our study's external validity. The results could vary with different demographics of the participants, educational environments, or levels of previous programming experience among the participants. Our findings were derived from a small set of schools in two countries. To generalize these findings, further research is needed, involving more institutions and conducting multi-institutional and multinational studies in diverse settings.

We intentionally avoided a direct comparison of student performance between the two countries. We believe that the differences in primary school curricula are too substantial to allow for a valid cross-national comparison. As previously noted in this paper, Informatics is a mandatory subject in Serbian primary schools, while, in Slovenia, it is offered as an optional subject. In Serbia, this ensures that all the students receive a foundational education in Computer Science, digital literacy, and basic programming skills. In contrast, while motivated students in Slovenia may be exposed to similar content, the coverage is less uniform and varies significantly, depending on individual school offerings and student choices. This disparity likely contributes to the substantial difference observed in the mean performance results shown in Table 2 and Table 3. An overview of receiving introductory programming in primary schools in different countries is discussed in greater detail in [4, 5].

9. Conclusion

By evaluating the impact of programming notation on learning outcomes, this paper aims to provide insights into the use of three distinct approaches for teaching programming to beginners.

In conclusion, our study examines the choice of notation, whether text-based, block-based, or both notations used simultaneously, for teaching programming concepts to novice programmers during short-term visits to primary schools. The results of our study, conducted in two countries, show that the choice of notation (whether block-based, text-based, or both simultaneously) did not affect the short-term programming performance of primary school children significantly. The results of data mining with the Apriori algorithm suggest that, for country 1 (Serbia), there is no significant correlation between performance and notation types during the two-hour sessions. The results of country 2 (Slovenia) indicate that the performance was slightly better for tests in multiple-representation environments. Even when the training duration was extended from two to four hours, the performance differences between the notation groups remained statistically insignificant. These findings suggest that educators have the flexibility to choose the programming notation for beginner sessions without impacting immediate learning outcomes negatively in terms of performance. However, the performance improved with a longer training time,

highlighting the importance of instructional duration over notation type in beginner programming education. The potential sacrifice of lower performance results for teaching both notations simultaneously in the same time slot as other notations was rejected by the results of this study. This finding suggests that introducing both notations simultaneously is a viable strategy even within time-constrained settings, as it does not compromise beginners' performance on basic concepts of programming.

For future work, longitudinal studies and follow-up research are essential, to explore the potential effects of teaching various notations in greater depth. We are planning additional experiment repetitions [44] with the same and similar settings, to validate our current findings, ensuring greater accuracy and reliability. Additionally, future research could explore whether different programming languages (e.g. C++) and tools (e.g. BlockPy [31]) offer different cognitive or performance outcomes compared to Poliglot. Future research should investigate whether such alternatives provide outcomes consistent with our results. In this study, we demonstrated how performance results were correlated with the duration of training (two hours vs. four hours). We intend to test our findings with future experiments. With a larger sample size, we can also analyze the participants' results with similar backgrounds, the same age, and the same conditions, thereby isolating the variables to be evaluated. Extending a multinational approach and considering diverse experiment settings is essential for a comprehensive understanding of using different programming notations and environments for teaching novice programmers.

Acknowledgments. This work is sponsored by the bilateral project "Programming Environments with Simultaneous Multiple Representations in Support of Early Programming Education" between Slovenia and Serbia. The Slovenian authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0041). This research has been supported by the Faculty of Organizational Sciences of the University of Belgrade.

References

1. Bubnič, B., Mernik, M., Kosar, T.: Exploring the predictive potential of complex problem-solving in computing education: A case study in the introductory programming course. Mathematics 12(11) (2024), https://www.mdpi.com/2227-7390/12/11/1655
2. Flannery, L.P., Silverman, B., Kazakoff, E.R., Bers, M.U., Bontá, P., Resnick, M.: Designing ScratchJr: Support for early childhood learning through computer programming. In: Proceedings of the 12th international conference on interaction design and children. pp. 1–10 (2013)
3. Dagienė, V., Jevsikova, T., Stupurienė, G., Juškevičienė, A.: Teaching computational thinking in primary schools: Worldwide trends and teachers' attitudes. Computer Science and Information Systems 19(1), 1–24 (2022)
4. Oda, M., Noborimoto, Y., Horita, T.: International trends in k-12 computer science curricula through comparative analysis: Implications for the primary curricula. International Journal of Computer Science Education in Schools 4(4), 24–58 (2021)
5. Szabo, C., Sheard, J., Luxton-Reilly, A., Simon, Becker, B.A., Ott, L.: Fifteen years of introductory programming in schools: a global overview of k-12 initiatives. In: Proceedings of the 19th Koli Calling International Conference on Computing Education Research. pp. 1–9 (2019)
6. Funke, A., Geldreich, K., Hubwieser, P.: Analysis of scratch projects of an introductory programming course for primary school students. In: 2017 IEEE global engineering education conference (EDUCON). pp. 1229–1236. IEEE (2017)

7. Mladenović, M., Žanko, Ž., Zaharija, G.: From blocks to text: Bridging programming misconceptions. Journal of educational computing research 62(5), 1302–1326 (2024)
8. Weintrop, D., Wilensky, U.: How block-based, text-based, and hybrid block/text modalities shape novice programming practices. International Journal of Child-Computer Interaction 17, 83–92 (2018), https://www.sciencedirect.com/science/article/pii/S2212868917300314
9. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al.: Scratch: programming for all. Communications of the ACM 52(11), 60–67 (2009)
10. Videnovik, M., Vlahu-Gjorgievska, E., Trajkovik, V.: To code or not to code: Introducing coding in primary schools. Computer Applications in Engineering Education 29(5), 1132–1145 (2021)
11. Wolber, D., Abelson, H., Spertus, E., Looney, L.: App inventor. O'Reilly Media, Inc. (2011)
12. Wilson, A., Hainey, T., Connolly, T.: Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. In: European Conference on Games Based Learning. p. 549. Academic Conferences InternationalLimited (2012)
13. Sáez-López, J.M., Román-González, M., Vázquez-Cano, E.: Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools. Computers & Education 97, 129–141 (2016), https://www.sciencedirect.com/science/article/pii/S0360131516300549
14. Tsukamoto, H., Takemura, Y., Nagumo, H., Ikeda, I., Monden, A., Matsumoto, K.: Programming education for primary school children using a textual programming language. In: 2015 IEEE Frontiers in Education Conference (FIE). pp. 1–7 (2015)
15. Moors, L., Luxton-Reilly, A., Denny, P.: Transitioning from block-based to text-based programming languages. In: 2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE). pp. 57–64. IEEE (2018)
16. Homer, M., Noble, J.: A tile-based editor for a textual programming language. In: IEEE Working Conference on Software Visualisation (VISSOFT). pp. 1–4 (2013)
17. Bau, D., Bau, D.A., Dawson, M., Pickens, C.S.: Pencil code: block code for a text world. In: Proceedings of the 14th International Conference on Interaction Design and Children. pp. 445–448 (2015)
18. Leber, Ž., Črepinek, M., Kosar, T.: Simultaneous multiple representation editing environment for primary school education. In: 2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). pp. 175–179. IEEE (2019)
19. Alves, L., Gajić, D., Rangel Henriques, P., Ivančević, V., Ivković, V., Lalić, M., Luković, I., Varanda Pereira, M.J., Popov, S., Correia Tavares, P.: C tutor usage in relation to student achievement and progress: A study of introductory programming courses in Portugal and Serbia. Computer Applications in Engineering Education 28, 1058–1071 (2020)
20. Kosar, T., Mernik, M., Carver, J.C.: Program comprehension of domain-specific and general-purpose languages: comparison using a family of experiments. Empirical software engineering 17, 276–304 (2012)
21. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering. Springer Science & Business Media (2012)
22. Kosar, T., Bjeladinović, S., Ostojić, D., Škembarević, M.S., Leber, Ž., Jejić, O.A., Furtula, F., Ljubisavljević, M.D., Luković, I.S., Mernik, M.: Teaching beginners to program: should we start with block-based, text-based, or both notations? In: 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS). pp. 395–403. IEEE (2024)
23. Zaharija, G., Mladenović, S., Boljat, I.: Introducing basic programming concepts to elementary school children. Procedia-social and behavioral sciences 106, 1576–1584 (2013)
24. Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E.: The Scratch programming language and environment. ACM Transactions on Computing Education (TOCE) 10(4), 138–144 (2010)

25. Handur, V., Kalwad, P.D., Patil, M.S., Garagad, V.G., Yeligar, Nagaratna andPattar, P., Mehta, D., Baligar, P., H., J.: Integrating class and laboratory with hands-on programming: Its benefits and challenges. In: IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE). pp. 163–168 (2016)
26. Salant, O.M., Armoni, M., Ben-Ari, M.: Habits of programming in Scratch. In: ITiCSE '11: Proceedings of the 16th annual joint conference on Innovation and technology in computer science education. pp. 168–172 (2011)
27. Cooper, S., Dann, W., Pausch, R.: Alice: a 3-D tool for introductory programming concepts. Journal of Computing Sciences in Colleges 15(5), 107–116 (2000)
28. Maloney, J.H., Peppler, K., Kafai, Y., Resnick, M., Rusk, N.: Programming by choice: urban youth learning programming with scratch. In: Proceedings of the 39th SIGCSE technical symposium on Computer science education. pp. 367–371 (2008)
29. Martelli, A., Ravenscroft, A.M., Holden, S., McGuire, P.: Python in a Nutshell. " O'Reilly Media, Inc." (2023)
30. PYPL: PYPL - popularity of programming language. https://pypl.github.io/PYPL.html, accessed: 22.05.2024.
31. Bart, A.C., Tibau, J., Tilevich, E., Shaffer, C.A., Kafura, D.: BlockPy: An open access data-science environment for introductory programmers. Computer 50(5), 18–26 (2017)
32. Bau, D.: Droplet, a blocks-based editor for text code. Journal of Computing Sciences in Colleges 30(6), 138–144 (2015)
33. Kölling, M.: The Greenfoot Programming Environment. ACM Transactions on Computing Education (TOCE) 10(4), 1–21 (2010)
34. Fowler, M.: Language Workbenches: The Killer-App for Domain Specific Languages? http://www.martinfowler.com (2005)
35. Voelter, M., Pech, V.: Language modularity with the MPS language workbench. In: 2012 34th International Conference on Software Engineering (ICSE). pp. 1449–1450. IEEE (2012)
36. Weintrop, D., Wilensky, U.: Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education (TOCE) 18(1) (oct 2017), https://doi.org/10.1145/3089799
37. Weintrop, D., Wilensky, U.: Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. Computers & Education 142, 103646 (2019)
38. Zhen Xu, Albert D. Ritzhaupt, F.T., Umapathy, K.: Block-based versus text-based programming environments on novice student learning outcomes: a meta-analysis study. Computer Science Education 29(2-3), 177–204 (2019), https://doi.org/10.1080/08993408.2019.1565233
39. Espinal, A., Vieira, C., Guerrero-Bequis, V.: Student ability and difficulties with transfer from a block-based programming language into other programming languages: a case study in Colombia. Computer Science Education 33(4), 567–599 (2023), https://doi.org/10.1080/08993408.2022.2079867
40. Lin, Y., Weintrop, D.: The landscape of block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. Journal of Computer Languages 67, 101075 (2021), https://www.sciencedirect.com/science/article/pii/S259011842100054X
41. Kosar, T., Bohra, S., Mernik, M.: A Systematic Mapping Study driven by the margin of error. Journal of Systems and Software 144, 439–449 (2018)
42. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advances in knowledge discovery and data mining. American Association for Artificial Intelligence (1996)
43. Feldt, R., Magazinius, A.: Validity threats in empirical software engineering research - an initial survey. In: 22nd International Conference on Software Engineering & Knowledge Engineering (SEKE'2010), Redwood City, San Francisco Bay, CA, USA, July 1 - July 3, 2010. pp. 374–379. Knowledge Systems Institute Graduate School (2010)

44. Kosar, T., Gaberc, S., Carver, J.C., Mernik, M.: Program comprehension of domain-specific and general-purpose languages: replication of a family of experiments using integrated development environments. Empirical Software Engineering 23, 2734–2763 (2018)

Tomaž Kosar is an Associate Professor at the Faculty of Electrical Engineering and Computer Science, University of Maribor. He received his M.Sc. and Ph.D. degrees in Computer Science from the University of Maribor. His research interests include programming languages, domain-specific languages, model-driven development, program comprehension, and empirical software engineering. He is a member of ACM, a reviewer for several international research journals, and a program committee member for various international conferences. He currently serves as the head of the Computer Science and Information Technologies study programmes at the University of Maribor.

Srđa Bjeladinović is an Associate Professor at Faculty of Organizational Sciences, University of Belgrade. He received his M.Sc. and Ph.D. degrees in Information Systems from University of Belgrade. His research interests are databases, information systems development methodologies, integrated software solutions and ERPs. In recent years he has been researching NoSQL and hybrid databases.

Dragana Ostojić works as a teaching assistant at the Faculty of Electrical Engineering and Computer Science, University of Maribor. She completed her BSc and MSc degrees in computer science at the University of Maribor in 2019 and 2021. Since 2019, she has been part of the Programming Methodologies Laboratory.

Milica Škembarević is a teaching assistant at the Faculty of Organizational Sciences, University of Belgrade. She completed her master's thesis on consistency in distributed information systems in 2021 at the University of Belgrade. Her research interests include databases, information systems development methodologies, data mapping, and interoperability, with a particular focus in recent years on distributed systems.

Žiga Leber is a researcher at the University of Maribor, Faculty of Electrical Engineering and Computer Science. He received his MSc degree in Computer Science from the University of Maribor. His research interests include programming languages, domain specific languages, multiple representation environments and parsing.

Olga A. Jejić is a teaching assistant at the Faculty of Organizational Sciences, University of Belgrade. She earned her master's degree in 2021 at the same university, focusing her thesis on defining software architecture modalities based on the event-sourcing architectural pattern. Her research interests include databases and methodologies for developing information systems.

Matej Moravec received his BSc and MSc degrees in computer science from the University of Maribor, Maribor, Slovenia, in 2017 and 2019, respectively. He is currently pursuing a PhD in computer science and works as a teaching assistant at the Faculty of Electrical Engineering and Computer Science, University of Maribor. Since 2019, he has

been a member of the Programming Methodologies Laboratory. His research interests include evolutionary computation and single- and multi-objective dynamic optimization.

Filip Furtula is a Teaching Assistant at the Faculty of Organizational Sciences, University of Belgrade. He holds an M.Sc. in Information Systems from the University of Belgrade. His professional interests span software engineering, programming languages, and modern application architecture. He is particularly focused on domain-driven and model-driven approaches, as well as contemporary web technologies. In the past few years, his work has been oriented toward distributed systems, exploring microservices, event-driven design, and the actor-based computational model.

Miloš Ljubisavljević is a Teaching Assistant at Faculty of Organizational Sciences, University of Belgrade. He holds an M.Sc. degree in Information Systems from the University of Belgrade. His main interests include programming languages, domain-driven design, model-driven design and web technologies. In recent years, he has been researching topics around distributed systems such as microservice architectures, actor model of computing and event-driven design.

Ivan Luković received his diploma degree (5 years) in Informatics from the Faculty of Military and Technical Sciences in Zagreb in 1990. He completed his M.Sc. (former Mr, 2 years) degree at the University of Belgrade, School of Electrical Engineering in 1993, and his Ph.D. at the University of Novi Sad, Faculty of Technical Sciences in 1996. Currently, he works as a Full Professor at the Faculty of Organizational Sciences of the University of Belgrade, where he lectures in several Computer Science and Informatics courses. His research interests are related to Database Systems, Information Systems, Business Intelligence, Software Engineering, and Data Science. He is the author or co-author of over 200 papers, 4 books, and 30 industry projects and software solutions in the area. He supervised 13 completed Ph.D. theses. He created a new set of B.Sc. and M.Sc. study programs in Information Engineering, i.e. Data Science, at the Faculty of Technical Sciences. The programs were accredited for the first time in 2015. Currently, he is a chair of Managing Board of the Computer Science and Information Systems (ComSIS) journal. He is a member of Serbian AI Society.

Marjan Mernik received the MSc and PhD degrees in Computer Science from the University of Maribor in 1994 and 1998, respectively. He is currently a professor at the University of Maribor, Faculty of Electrical Engineering and Computer Science. From 2007 to 2017, he was a visiting professor at the University of Alabama at Birmingham, Department of Computer and Information Sciences. His research interests include programming languages, domain-specific (modelling) languages, grammar and semantic inference, and evolutionary computations. He is the Editor-in-Chief of the Journal of Computer Languages, as well as an Associate Editor of the Applied Soft Computing Journal and Swarm and Evolutionary Computation Journal. He has been named a Highly Cited Researcher for the years 2017 and 2018. More information about his work is available at https://lpm.feri.um.si/en/members/mernik/.

Matej Črepinšek received his B.Sc. in 1999 and his Ph.D. in 2007, both in Computer Science from the University of Maribor, Slovenia. He is currently an Associate Professor

at the Faculty of Electrical Engineering and Computer Science, University of Maribor. His research interests include game development, mobile development, grammar inference, evolutionary computation, single- and multi-objective optimization, and computer science education.

Received: April 01, 2025; Accepted: November 30, 2025.

Computer Science and Information Systems 23(1):61–74 https://doi.org/10.2298/CSIS250208003S

Review of Architectures in IoT Remote Patient Monitoring Systems

Lan Sovinc and Marko Bajec

University of Ljubljana,
Večna pot 113, 1000 Ljubljana, Slovenia
ls7312@student.uni-lj.si
marko.bajec@fri.uni-lj.si

Abstract. This review article examines different architectural approaches for IoT-based Remote Patient Monitoring Systems. Four different classification categories have been identified, each with their own strengths and weaknesses.

Fog-based architectures reduce latency and enhance privacy through localized data processing. Cloud-based solutions offer robust storage and analytics capabilities, but struggle with latency and centralization. Hybrid systems combine edge and cloud computing to improve scalability and efficiency, while blockchain-enabled architectures focus on secure, decentralized data management.

The findings from 11 original publications were summarized. This paper describes the characteristics of the identified approaches by examining common and unique concepts used by the solutions in the same classification category based on prior analysis.

Keywords: Remote Patient Monitoring, RPM, Internet of Things, IoT, IoT Healthcare Systems, Fog Computing, Cloud Computing, Blockchain, Scalable RPM Systems, Wearable Devices, Real-time Health Monitoring.

1. Introduction (Architecture approaches for IoT RPM)

Remote Patient Monitoring (RPM) systems are being increasingly discussed in modern healthcare [7]. RPMs relieve patients of manual health data collection and provide more representative measurements as they are being taken in their home environment. On the other hand, they benefit the medical institutions and their staff by reducing the number of physical visits, better monitoring of a higher number of patients and extending the range of medical services to a greater geographical area.

To implement a modern RPM system, different technologies can be chosen. One promising example is the Internet of Things (IoT). IoT and its healthcare-specific subset, IoMT (Internet of Medical Things), are being increasingly recognized for their potential to address medical requirements efficiently, especially in terms of scalability, security, and real-time monitoring [1]. The widespread use of this technology for home automation suggests growing maturity and feasibility of exploring this area for use in environments where the reliability and accuracy of systems and their measurements are critical. However, implementation of such solutions using IoT brings several system design challenges that should be addressed to improve effectiveness [11].

This review focuses on analyzing and classifying the architectural paradigms employed in IoT-based RPM systems examined and how they ensure high availability, patient privacy and scalability. Four distinct classes of IOT RPM system were identified: cloud-based, fog-based, hybrid, and blockchain-enabled. By classifying solutions based on data processing and system functionality, the review examines innovations and trade-offs in RPM design. This work aims to contribute to understanding the strengths and limitations of architectural approaches, providing valuable insight to system designers and researchers to improve IoT-based RPM systems.

The remainder of this paper is structured as follows. Section 2 (Search strategy) describes the literature search and selection process. Section 3 (Publication Classification) presents the classification framework used to group the identified RPM architectures. Sections 4–7 then discuss fog-based, cloud-based, hybrid, and blockchain-enabled architectures in detail. Finally, the Conclusion summarizes key findings and outlines directions for future research on IoT-based RPM system design.

2. Search strategy

Studies published before 1st February 2024 were retrieved from The IEEE Xplore digital library database. The following search term was used: IoT AND (”Remote Patient Monitoring” OR ”RPM”) AND (”Non-Invasive Technologies” OR ”Digital Health Technologies” OR ”Wearable Devices”). The search period was limited to publications published from 2018 onward in order to restrict the search window to the last five years, calculated from the time we started writing the article. Our search yielded 35 results. The abstracts and full texts of each publication were screened and assessed for inclusion. Out of 35 initial publications, 23 were excluded for several reasons:

- 16 did not discuss the topic of our review article directly
- 6 were reviews without original data
- 1 was a survey
- 1 was a physical-only resource that we did not have access to

Finally, 11 original publications were included in this review.

3. Publication Classification

We analyzed the selected publications in order to develop a system for classifying their discoveries into different categories. In this way, we could later compare the details of the solutions that used the same approaches. By looking only at the specifics of the architecture within a particular paradigm, we sought to enable more direct comparisons.

We decided to classify the solutions according to where most of the data is processed, in relation to the role of the device in the system. With this approach, we formed 4 categories. “Fog-based architecture”, where the processing power is partially transferred from the central server to the periphery, as opposed to the more traditional “cloud-based architecture”, where the powerful processing takes place on a centralized remote server or group of servers. Then we observed some approaches where simple operations are performed on the edge devices, but more complex analytics are moved to the cloud, which

we call “hybrid architecture”. Finally, we observed a interest in the use of blockchain paradigms to fulfill the requirements of an IoT RPM system and decided to include a chapter discussing the pros and cons of integrating “blockchain technology” in the system architecture.

All examined solutions, as represented in the supplementary material, are provided for reference.

4. Fog-based Architecture

In the early stages of the development of RPM IoT system architectures, there were two main approaches: Edge-based and cloud-based approaches. Both have their own advantages and disadvantages, which we will discuss in later chapters. For some years now, however, there has been a middle ground between these two approaches, namely the use of the so-called fog computing paradigm. The term was invented to refer to “cloud computing” by comparing how fog appears in nature in relation to clouds. In this case, some of the computational capabilities are spread from a central cloud closer to the IoT devices (the fog). Such devices are usually responsible for communicating with multiple IoT devices on the same local network. A commonly cited advantage of fog nodes is their relative computing power compared to the IoT sensors. This can reduce central server load through local filtering and processing, may enhance the privacy aspect of such systems as less data is sent outside the patient’s home network, and can support more advanced emergency response capabilities.

The first example of a fog-based infrastructure is the one in [5]. The goal of the RPM system was a low-latency approach to detect abnormalities of patients’ heart rhythm by ECG (electrocardiogram) using IoT wearable devices and deep learning. The system is represented by the sensor, fog and cloud layers. Wearable IoT devices are part of the sensor layer, which is capable of transmitting data to the fog layer and responding to commands from the fog layer when medical emergencies are detected. The devices in this layer are the simplest and therefore cannot perform data analysis. The fog layer consists of a cluster of fog devices, called gateways, which act as intermediaries between the sensors and the cloud. Each fog node has its own local storage for the temporary storage of measurement data. If the node cannot process the signal locally due to a high computational load, the data is transferred from the local storage to the cloud for processing. In a scenario without overload, the signal is first filtered and then processed on the node. As they are computationally more powerful than the sensor level devices, but still far less powerful than the cloud, the fog nodes perform the signal classification using the inference model downloaded from the cloud, but do not implement the deep learning process. If an abnormal value is detected, the alert is first sent to the sensor layer to notify the emergency services before being sent to the cloud layer for model training. The cloud layer is responsible for the global storage of the measured values, the constant training process of the deep learning model and the transmission of the latest inference model to the fog layer. The system architecture was tested together with a one-dimensional convolutional neural network as a classification model. The authors reported a 25% faster response time compared to the conventional cloud-based approach.

The second fog-based computing architecture [6] attempts to optimize patient privacy, data processing, system response time and patient-service interaction. The system uses

IoT sensors in combination with a surveillance camera feed and speech recognition devices to improve interaction. The fog node is the main component of the framework. The sensor data is transmitted to the fog device, which processes it and obtains the patient's consent to connect via the voice recognition module if an abnormal value is detected. The medical center is only given permission to view the camera feed if the patient has given consent or if there is an emergency (if the patient is asked for consent but does not respond). In this way, the fog node fulfills three tasks. It provides local data filtering that ensures patient privacy. By checking for anomalies and speech recognition in the fog node, processing is offloaded from the main server. Emergency confirmation process is improved using the camera feed. The system prototype was implemented with 3 Raspberry Pis for speech recognition and an NVIDIA Jetson TX1 as a fog node. Compared to a conventional system architecture, where all processing takes place in the cloud, the fog-enabled system reported an average round-trip latency for processing requests that was 100 times lower.

The gateway does not always have to be a separate device. To save costs and make better use of hospital resources, [9] developed an early warning system that uses the patient's smart device as a gateway. The aim of the system was to detect a sudden deterioration in a patient's health after discharge from the ICU (intensive care unit). The system consists of wearable IoT devices, a smartphone or tablet that serves as a gateway, and a remote early warning and analysis system. The medical data from the IoT devices is transmitted to the gateway via Bluetooth. There, the raw data is processed using specially developed mathematical models and the current health metrics are generated. Communication between the gateway and the remote analysis system takes place via local or cellular networks. The remote analysis system displays the readings to the medical staff, but also provides automatic alerts with an early warning value for the emergency response team. By utilizing existing resources, the team was able to quickly implement the system and conduct a successful pilot study with more than one hundred patients.

The examples of architectures described above show that fog computing paradigms may offer several potential benefits for use in IoT-based Remote Patient Monitoring (RPM) systems. It combines the strengths of edge and cloud computing while mitigating their respective limitations. By decentralizing computational tasks closer to the data sources, fog architectures offer improved data protection, reduced server load and improved emergency response capabilities. Comparing the solutions examined, fog implementations differ in their approach to data protection, latency and resource utilization, despite their common basis in the distribution of processing tasks. For example, the integration of advanced data processing and emergency protocols directly in the fog layer, as seen in [5], contrasts with architectures that prioritize patient privacy through localized consent mechanisms [6]. Furthermore, the innovative use of existing devices as gateways [9] emphasizes the versatility and cost-effectiveness of fog computing in healthcare. Overall, these examples indicate a potential role for fog computing in the development of RPM systems by providing a balanced, adaptable framework that meets the critical needs of healthcare IoT infrastructure.

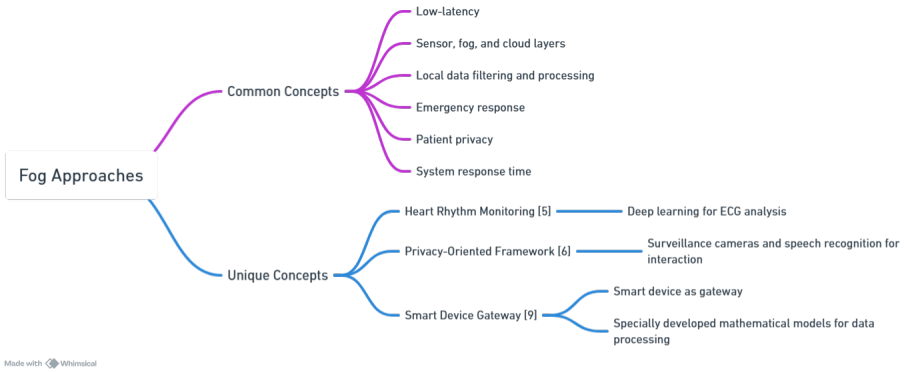


Fig. 1. Mind map for Fog-based architectures comparison

5. Cloud-based Architecture

If we want to perform complex data processing, have centralized storage of medical records or use sophisticated data analysis approaches such as machine learning or deep learning, we need powerful resources that a cloud-based approach can provide. In such scenarios, the IoT devices only transmit the data to the cloud and sometimes also receive the analysis results. Such systems can provide substantial processing capacity, but the dependence on a central server may introduce a single point of failure. Another challenge is the latency that is sometimes required for real-time RPM systems. As the amount of data at the ingestion point increases and data processing becomes more complex, the latency can become more pronounced.

The fog node is not required if the sensor devices themselves are equipped with sufficiently powerful hardware. This means that the medical IoT device can be connected directly to the central server, for example via Wi-Fi. This simplifies the architecture, but places a greater load on the central server. By using an established cloud infrastructure provider, the load can be distributed internally.

The smart healthcare monitoring system by [13] used this approach by combining Wi-Fi IoT sensors with a cloud database Firebase. They implemented a real-time monitoring solution with two mobile applications, one for patients and one for medical staff. By developing a background service, alerts were implemented for both parties, even when the mobile application is not actively running on their devices. The Firebase Realtime Database API was used to respond to data changes without the need for user intervention or manual updating of the app, contributing to the real-time requirements of the system.

A similar hardware approach was taken in [8], where a centralized system was developed for people infected during a pandemic. It allows the authorities to verify that the person is following quarantine instructions and the medical staff and family to monitor the patient's health status while providing a standardized EHR (electronic health record) cloud data storage solution. The system consists of three layers. IoT wearable devices take medical measurements and forward them to the online platform along with the device's GPS (Global Positioning System) data. The platform consists of a database and a server on

which the data processing takes place. First, the platform checks whether the transmitted vital data is within the normal range and only if this is not the case are the GPS coordinates checked. Based on this data, an emergency alert can be triggered to the authorities if the patient's location does not match the agreed quarantine location. The medical measurement is then standardized as an EHR and written to the cloud database. A web-based monitoring interface has been implemented for medical staff. The server also servers the data to the mobile companion app, which the patient's relatives can use to find out about their current state of health. A limitation of this solution is that the automatic alerts are only implemented for quarantine violations, but not for abnormal medical measurements, which still require manual review by medical staff or the patient's relatives.

IoT Remote Patient Monitoring (RPM) systems that use a cloud-based infrastructure share a certain degree of common characteristics of the models discussed. All of the architectures examined leverage the power of cloud-based data processing and centralized storage, which underpins their robust ability to process complex data operations and enable monitoring capabilities through IoT devices. Despite these common attributes, both solutions focused on different types of functionalities. Article [13], for example, features seamless real-time data synchronization and uses a background alert service to enable seamless monitoring without direct app interaction. On the other hand, Article [8] integrates GPS data to enforce quarantine compliance, providing a unique layer of patient management. It also introduces a web-based interface for healthcare professionals and selectively automates alerts for quarantine violations, while other critical alerts are not automated. These nuanced differences highlight the tailored approaches to address specific healthcare monitoring challenges and emphasize the importance of context-driven architecture design in IoT-based patient care.

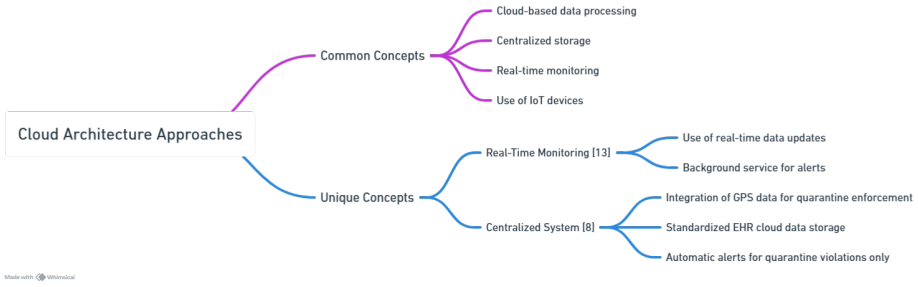


Fig. 2. Mind map for Cloud-based architectures comparison

6. Hybrid Architecture

The edge-based solutions are decentralized, straightforward to implement and not dependent on another external system to perform the basic tasks. They usually consist of small battery-powered IoT devices that may or may not communicate with external systems, but perform their data analysis independently. The advantage of these devices is that they

are able to run simple algorithms to perform basic local signal analysis. However, due to their size, they are limited in terms of resources and power. This subsection discusses architectural solutions that mainly utilize edge computing paradigms but seek to mitigate limitations of requiring powerful edge devices by utilizing the cloud for more complex tasks and data centralization.

In both solutions [2] and [4], approaches were developed in which a ubiquitous device, e.g. the patient's smartphone, served as a gateway between the IoT devices and the cloud. The main difference to the fog-based solution is that apart from data aggregation and forwarding, no data processing takes place locally on the device itself.

The solution in [2] utilized IoT smart patches to measure ECG signals from patients and developed an RPM framework for monitoring and real-time data analysis using a microservice architecture to achieve better performance and scalability. Data acquisition, streaming, processing, storage and reporting were implemented. To overcome the resource constraints of smart patches, data is transmitted via BLE (Bluetooth Low Energy) to the nearest paired smartphone. An app was developed for this device, which serves as a gateway to the cloud and enables the management of IoT devices, the display of real-time data and alerts for patients. The app also creates a background service that connects to the backend via MQTT (Message Queuing Telemetry Transport) protocol to enable bidirectional communication. The core of the backend consists of the open source IoT platform ThingsBoard. It enables the system to store patient personal data and time series data from the IoT patches, send notifications to the smartphone app and communicate with other backend services via the so-called "integrations". Two main integrations are those for the MQTT broker for data acquisition and for the event storage and stream processing platform. The scalability of the system is achieved through the Apache Kafka event store. It enables the system to forward the data to the corresponding microservice without overloading its resources. It prepares the data and waits until the target service is ready to process it. The key microservice is the one for data analysis with machine learning algorithms, which enables real-time alerts and decision making. After the data is analyzed, a new event is created containing the results of the data classification along with the raw data to be written to the database for long-term storage. This database is also used as a source for the dashboard web application, which displays the ECG charts to the medical staff and informs them of any anomalies in their patients' measurements. An advantage claimed for this approach, which overcomes the traditional disadvantage of cloud architecture, is the use of an event processing platform together with the microservice backend architecture to achieve sustained performance of the system as the amount of data at the entry point increases.

Another case of a hybrid approach was described in [4], in which both the cloud and the resources already available (patient's smartphone) were used. The system implements all the main functions of the larger systems, but by reusing existing resources and developing a single code base, the solution is very compact and therefore easy to maintain and deploy. The aim of the system was to implement RPM, in particular to measure heart rate and oxygen saturation and to provide a a solution intended to be readily available and low-cost. The developed app not only provides the management interface for the system, but can also capture medical data from the device's camera to measure heart rate and blood oxygen saturation using photoplethysmography. The same measurements could also be taken in a more precise way using an IoT sensor device with more specialized sensor

equipment. In this case, the IoT device transmits all data to the cloud independently, and the smartphone is only used to display the data by querying the server. The same app is used for patients and doctors. Patients can view their measurement data, receive medical alerts and manage medical services. Doctors can view their patients' data and communicate with them via the integrated chat function.

Hybrid architectural approaches in IoT Remote Patient Monitoring (RPM) systems show a mix of edge and cloud computing paradigms to leverage the strengths of both systems while mitigating their respective weaknesses. Commonalities between the solutions discussed include the use of patient-owned devices as intermediary gateways, system scalability and ensuring user-friendly interfaces. Differences can be seen in the scope of data processing, with some approaches favoring minimal edge computation and others incorporating extensive cloud-based analytics and decision-making capabilities. Of note is the innovative use of existing resources (e.g., smartphones) to reduce system complexity and cost, as in solution [4], in contrast to the more elaborate service-oriented architecture of solution [2], which emphasizes performance and scalability through microservices and event-driven design. These approaches demonstrate the critical balance between resource efficiency, real-time responsiveness, and system scalability in designing effective remote patient monitoring (RPM) systems.

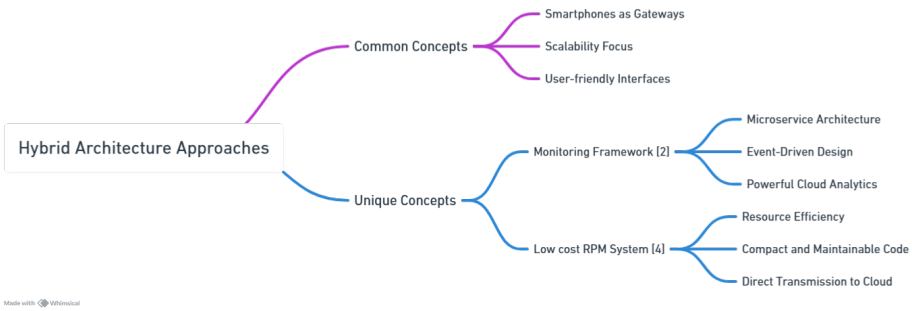


Fig. 3. Mind map for Hybrid architectures comparison

7. Blockchain Technology in RPM

Blockchain technologies have seen broader adoption beyond digital currencies. Several articles that dealt specifically with how utilizing blockchain could improve the current RPM system architectural approaches. Even though, the described solutions utilize the paradigms of all previously described architectures, we decided to include it into a separate section as they still act distinctly differently from the traditional systems.

This section discusses potential benefits of blockchain technologies to IoT RPM systems, how blockchain technology can be integrated into system architecture, how alerts can be automated using smart contracts, what consensus mechanisms can be used between nodes, and how confidential data can be stored in a decentralized manner aiming to support the confidentiality, integrity and availability of data.

All articles that include blockchain technologies in their solutions show some similarities in their approach [14, 12, 10, 3]. All solutions chose blockchain to enable efficient and secure transfer of medical data, protect patient privacy and manage access to confidential information. A private blockchain was considered useful in this context as the consensus mechanism between nodes can be simplified, allowing the system to consume fewer resources, which is a potential benefit in resource-constrained IoT environments. None of the solutions relied on commercial blockchains (such as Ethereum), but developed their own stripped-down version that is less resource-intensive. Furthermore, none of the solutions stored the data directly on the blockchain, but used a type of cloud storage for the actual data and only stored the data hashes on the blockchain to improve the resource efficiency and response times of the system. This was complemented by the use of fog nodes to bring compute, storage and network services closer to the IoT devices, reduce latency, improve data protection and increase scalability. Although the same technology was used, the exact implementation specifics and different innovative approaches were used, as described below.

The researchers in [14] have developed a blockchain-based system based on Hyperledger Fabric, a permissioned distributed ledger, and combined it with cloud storage to improve robustness and data CIA (confidentiality, integrity, and availability) in IoT RPM systems. The system architecture comprises four layers: Hyperledger Fabric, Hyperledger Composer, Cloud layer and RPM application layer.

Hyperledger Fabric represents an enterprise-focused private blockchain that provides peer certificate issuance services, a distributed immutable ledger, and a specialized smart contract service called Chaincode. To build the business logic, Hyperledger Fabric uses a connection profile to Hyperledger Composer, which provides a domain-specific language for modeling the business network with its associated assets (medical data), participants (patients and doctors) and transactions. The same service also enables the easy creation of a REST (Representational State Transfer) API service to query the model.

The data from the medical IoT devices is first sent to an IoT Fog Gateway in the local network for data aggregation and temporary storage in case of network unavailability. The gateway then forwards the information to the Hyperledger Fabric running in the cloud. The cloud ledger consists of encrypted medical measurements stored in chained blocks. Once the data is received, the ledger is updated and transactions are stored in cloud storage to improve efficiency and scalability. Patients and doctors interact with the system via the RPM web application, which retrieves the data via the REST service.

Placing a private ledger at the fog layer showed lower server response times and steadier throughput than a cloud-only baseline as device counts and packet sizes rise, indicating that edge aggregation and permissioned consensus can keep end-to-end latency acceptable for wearables.

The model in [12] attempts to eliminate security issues by using blockchain technologies tailored to the resource-constrained environment of IoT. It eliminates the Proof of Work (PoW) consensus mechanism and explores effective cryptographic methods that still match the computing power of IoT devices. An overlay P2P (”peer-to-peer”) network based on a distributed architecture consists of devices called nodes grouped into clusters managed by a cluster head. Communication takes place via the cluster heads which use public key encryption to secure the data and a custom made blockchain consisting of data hashes. The original patient data is not stored in the blockchain but in the cloud to improve

the efficiency of the system. The data is stored in identical blocks and encrypted with the patient's public key before its hash is sent to the overlay network. A smart contracts environment is also implemented to enable automatic alerts for healthcare providers. The main cryptographic concepts in this paper are the ARX encryption algorithms and ring signature. ARX algorithms (Addition, Rotation and XOR) are, as the name suggests, a branch of symmetric encryption algorithms that are implemented with simple operations. This makes them suitable for use in computationally less powerful devices. The system uses ARX for double encryption. The medical data is first encrypted with a symmetric key before public key cryptography is used. Another technique, known as ring signature, allows the signer to sign the data anonymously. When the patient wants to sign their data, the system collects other signing requests from other participants and mixes them before sending them across the network. By blending their signature with other signatures in the ring, no one but the signer knows who signed which medical record.

In [10], a privacy-preserving scheme for the secure transfer of patient data and medical diagnoses using a private blockchain and swarm exchange methods is proposed. Swarm is a decentralized P2P file transfer paradigm represented by the nodes in the blockchain network. Files are identified by their hashes and distributed across the network in chunks that are stored in the peer pool. Multiple nodes store the same chunks, ensuring availability, redundancy and resilience to partial outages. The policy, known as Swarm exchange, handles node incentivization that is shared with the nodes which participate in the process by storing the file chunks or providing requested data. In this case, the files the swarm is working with are patient EHRs. Heterogeneous data from the IoT device pool is collected by the aggregator module and converted into an EHR. The data is then encrypted with the healthcare professional's public key. After that, the data itself is stored in the swarm exchange (implemented with InterPlanetary File System (IPFS)), while its hash is sent to the private blockchain, where the transaction is mined and added to the chain. The healthcare professional can then download the file from the swarm exchange after the system confirms through hash checks that the files have not been tampered with, resulting in double-layer security. Exactly the same principles are applied in the reverse scenario, where the doctor makes a diagnosis for the patient.

Swarm availability during uploads remained 10 ms across average EHR sizes, with upload/download times scaling with file size which could be an evidence that off-chain storage with on-chain hashes can bound consensus-induced delays.

Paper [3] presents a private blockchain-based Wireless Body Area Network (WBAN) for the secure and private collection and management of medical data. The main components of the platform are: cloud server, fog device and edge devices. The cloud server represents the medical facility, stores all patient and measurement data and acts as a full blockchain node. The fog devices (usually the patients' smartphones) are also instances of full nodes, which is the opposite of what similar platforms have implemented in the past. The fog devices aggregate communication with the WBAN and are the only components that communicate directly with the cloud service. These devices also manage medical IoT devices and authorize data transfer by performing local checks with the blockchain. By offloading the cloud node, it was possible to achieve process redistribution that offers higher performance, better scalability and more flexibility compared to systems where fog devices do not act as full nodes.

An analysis of the similarities between the approaches examined reveals a number of common features. The blockchain paradigms are used for secure and efficient transfer of medical data as well as robust protection of patient privacy and management of access to sensitive information. The consensus for using private blockchains as opposed to public blockchains stems from the need to minimize resource consumption — a critical consideration in the inherently resource-constrained IoT ecosystems - while allowing an overseeing party to manage system participants. Furthermore, the strategic decision to forgo direct data storage on the blockchain and instead opt for cloud storage with blockchain-based hash references underscores a universal commitment to improving system responsiveness and resource efficiency. Despite these similarities, there are differences in implementation specifics and innovative approaches. Unique cryptographic methods [12], the use of fog computing nodes [14, 3], and novel data management strategies [10] are examples of solutions tailored to specific system requirements and constraints.

While we recognize that network and consensus latencies could still be optimized, the experimental evidence suggests blockchain-based RPM is feasible for low-power IoMT when:

- Permissioned chains avoid PoW,
- Lightweight cryptography (e.g., ARX)ring signatures are used judiciously,
- Data are kept off-chain with fog-level consensus.

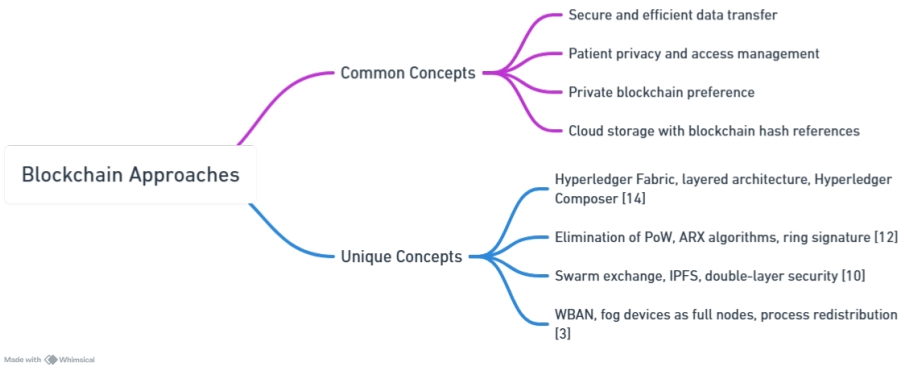


Fig. 4. Mind map for Blockchain architectures comparison

8. Conclusion

In conclusion, the review of architectural approaches in IoT-based Remote Patient Monitoring (RPM) system design has described diverse approaches aimed at enhancing healthcare delivery through technology.

Four main paradigms—cloud-based, fog-based, hybrid, and blockchain-enabled were identified and analyzed, reflecting ongoing efforts to balance computational efficiency, data privacy, and real-time responsiveness.

Fog-based solutions focus on low latency, local data filtering, quick system response and good patient privacy. For resource-intensive analytics and robust data storage needs, cloud-based infrastructure may be well suited. Hybrid approaches seek to combine the benefits from cloud- and edge-based solutions which enable better scalability while still keeping the overall system costs down by utilizing patients' existing devices. Lastly, the innovative blockchain approaches build on the ability to enable secure data exchange and robust patient privacy, utilizing private blockchains.

Discussed results come from small prototypes, with limited, non-uniform reporting of end-to-end quality of service (Qos) (latency, availability), privacy guarantees, and real-world fault tolerance, making comparisons difficult. Going forward, we see value in standardizing evaluation (benchmarks, shared datasets) and systematically testing live deployments, so missing elements in prototypes' deployment assessments (e.g. interoperability, sustained QoS, energy use, and compliance) are explicitly identified.

The review classifies different approaches for an RPM system design to compare the similarities and differences of the examined systems. This may help researchers and system designers assess which solutions are appropriate for specific contexts and therefore drive innovation of IoT-based healthcare solutions.

Table 1. Compact comparison of IoT RPM architectures and how each addresses core metrics

Architecture	Latency	Scalability	Implementation Cost	Performance / Throughput
Fog-based	*Short round-trip time* ([5, 6])	*Offload central server*	*Reusing existing hardware* ([9])	*Processing proximity, server offload*
Cloud-based	*Real-time-capable; load-sensitive* ([13])	*Elastic cloud infrastructure*	*Fewer devices - lower cost*	*Infinitely scalable processing capacity*
Hybrid	*Real-time via ML* ([2])	*Microservices scalability* ([2])	*Reusing existing hardware* ([4])	*High ingress capability* ([2])
Blockchain-enabled	*Fog nodes; off-chain hashes* ([14, 10, 12, 3])	*Fog full nodes* ([3])	*Private chain utilization* ([12])	*Processing redistribution* ([3])

References

1. Arthi, K., Chidhambararajan, B., Revathi, A.R.: A deep investigation of architectural elements and computing technologies for internet of medical things. In: 2022 6th International Conference on Electronics, Communication and Aerospace Technology. pp. 556–563 (2022)
2. Badr, A., Elgazzar, K.: A framework for real-time remote ecg monitoring and diagnoses. In: 2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA). pp. 1–10 (2022)
3. Baucas, M.J., Spachos, P., Gregori, S.: Private blockchain-based wireless body area network platform for wearable internet of thing devices in healthcare. In: ICC 2023 - IEEE International Conference on Communications. pp. 6181–6186 (2023)
4. Chauhan, A., Farmah, K., Goel, A., Gandotra, A.: A novel patient monitoring system using photoplethysmography and iot in the age of covid-19. In: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). pp. 427–437 (2021)
5. Cheikhrouhou, O., Mahmud, R., Zouari, R., Ibrahim, M., Zaguia, A., Gia, T.N.: One-dimensional cnn approach for ecg arrhythmia analysis in fog-cloud environments. IEEE Access 9, 103513–103523 (2021)
6. Jayson Baucas, M., Spachos, P.: Fog and iot-based remote patient monitoring architecture using speech recognition. In: 2020 IEEE Symposium on Computers and Communications (ISCC). pp. 1–6 (2020)
7. Mahmmod, B.M., Naser, M.A., Al-Sudani, A.H.S., Alsabah, M., Mohammed, H.J., Alaskar, H., Almarshad, F., Hussain, A., Abdulhussain, S.H.: Patient monitoring system based on internet of things: A review and related challenges with open research issues. IEEE Access 12, 132444–132479 (2024)
8. Palli, G.H., Mirza, G.F., Chowdhry, B.S.: Novel iot-based e-health system: Hospital management, telemedicine and quarantine management for covid-19. In: 2022 Third International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT). pp. 1–9 (2022)
9. Pathinarupothi, R.K., Sathyapalan, D.T., Moni, M., Menon, K.A.U., Ramesh, M.V.: Rewoc: Remote early warning of out-of-icu crashes in covid care areas using iot device. In: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 2010–2013 (2021)
10. Ray, P.P., Chowhan, B., Kumar, N., Almogren, A.: Biothr: Electronic health record servicing scheme in iot-blockchain ecosystem. IEEE Internet of Things Journal 8(13), 10857–10872 (2021)
11. S, D.R., B, A.: Iot revolutionizing healthcare: A survey of smart healthcare system architectures. In: 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE). pp. 1–5 (2023)
12. Srivastava, G., Crichigno, J., Dhar, S.: A light and secure healthcare blockchain for iot medical devices. In: 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE). pp. 1–5 (2019)
13. Sultana, S., Rahman, S., Rahman, M.A., Chakraborty, N.R., Hasan, T.: An iot based integrated health monitoring system. In: 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA). pp. 549–554 (2021)
14. Zaabar, B., Cheikhrouhou, O., Ammi, M., Awad, A.I., Abid, M.: Secure and privacy-aware blockchain-based remote patient monitoring system for internet of healthcare things. In: 2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). pp. 200–205 (2021)

Lan Sovinc (m), MSc, is a full-stack software engineer and IoT developer who completed his master's studies at the Faculty of Computer and Information Science, University of

Ljubljana. His professional work spans web and IoT system development. He is particularly interested in the design of scalable architectures that combine cloud, fog, and edge computing, as well as the use of AI-driven services. Since 2023, he has been working as a freelancer and participating in projects involving web3 technologies, decentralized finance and artificial intelligence.

Bajec Marko (m), PhD, is a Professor of computer science at the Faculty of Computer & Information Science, University of Ljubljana. He is the Head of the Laboratory for Data Technologies and Iot Demo Center. From 2013 to 2017, he acted as a Vice Dean for Economic Affairs at the Faculty for Computer and Information Science. His research interests related to data technologies include speech and language technologies, information retrieval, web search and extraction, data integration, data management, and data analysis. In the last 20 years, he has led or coordinated more than 40 R&D projects in total amount more than 50 FTE. For his work, he has received several awards and recognitions, such as an Award for contribution in transferring knowledge to industry, Slovenian Society Informatika (2010), an Award for ongoing achievements, international conference on Information Society (2013), a Gold Plaque for outstanding achievements in research and pedagogical work, University of Ljubljana (2014), a Mentor of the Year award, Slovenian association of doctoral students (2014). In his career, he has served as a program board member, program chair or general chair of many international conferences.

Received: February 08, 2025; Accepted: November 20, 2025.

Computer Science and Information Systems 23(1):75–112 https://doi.org/10.2298/CSIS240430073G

DomiCom: Discovery of Top-k Dominant Communities in Networks with Node Attributes

Nikolaos Georgiadis[1], Eleftherios Tiakas[2], and Apostolos N. Papadopoulos[1]

[1] Aristotle University of Thessaloniki
Thessaloniki, Greece
ngeorgii@csd.auth.gr
papadopo@csd.auth.gr
[2] International Hellenic University
Thessaloniki, Greece
tiakas@ihu.gr

Abstract. The community structure is an inherent property of real-world networks. Broadly, a set of nodes S forms a community if its nodes exhibit a significantly higher level of interconnection with each other than with nodes outside S. The precise notion of a community, however, depends on the mathematical formulation adopted. The problem becomes more intriguing when graph nodes are enriched with attributes, as these attributes can influence how communities are defined. In this work, we incorporate node attributes to construct more meaningful communities that reflect both structural connections and attribute information. Specifically, we introduce the concept of dominance relationships between nodes: a node u is considered more important than a node v if the attributes of u *dominate* those of v. Experimental evaluation on real-world attributed networks demonstrate the efficiency and effectiveness of the proposed approach. The reported communities are meaningful and robust based on their significance and structural coherence.

Keywords: community detection, attributed graphs, dominance relationships.

1. Introduction

The study of real-world networks [1] enables the discovery of significant knowledge that can be used to solve important problems such as community detection, outlier discovery, link prediction, just to name a few. A network is represented as a graph $G(V, E)$ where V is the set of nodes or vertices and E is the set of edges or links. In this study, we focus on undirected annotated graphs, where each node contains additional information (attributes) in the form of a vector. Our target is to detect subgraphs that are robust with respect to node interconnections and also strong with respect to the domination score, which is being used as a measure of node (and subgraph) significance.

In this work, we focus on the task of community detection in annotated (attributed) graphs. More specifically, given an annotated graph, where each node $u \in V$ contains a set of d attributes represented as a vector $vec(u)$, our aim

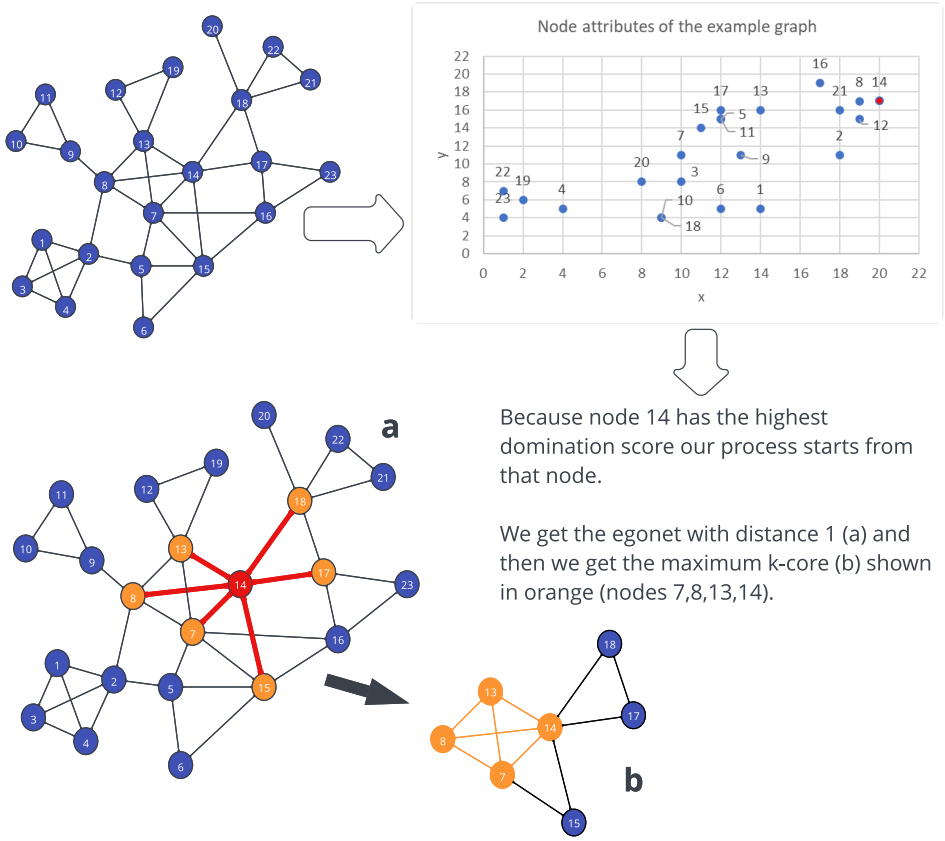


Fig. 1. Overview of the DomiCom process. An initial attributed graph (top left) is transformed into a scatter plot of node attributes (top right) to compute domination scores. The highest-scoring node (14) is chosen as the seed. (a) The 1-hop egonet around node 14 is induced. (b) The maximum k-core of this egonet yields the final dense community 7, 8, 13, 14

is to discover the c *most important* subgraphs (i.e., communities). The importance of a community is quantified using graph properties as well as properties related to node attributes. Without loss of generality, we assume that the larger attribute values are more significant than the smaller ones. Therefore, a node u with $vec(u) = [5, 7, 4]$ is considered more significant than another node v with $vec(v) = [2, 3, 2]$.

The domination score of a node u is computed by counting the number of nodes *dominated* by u and expresses the *relative power* of a node with respect to its attributes and can be interpreted differently depending on the application: in citation

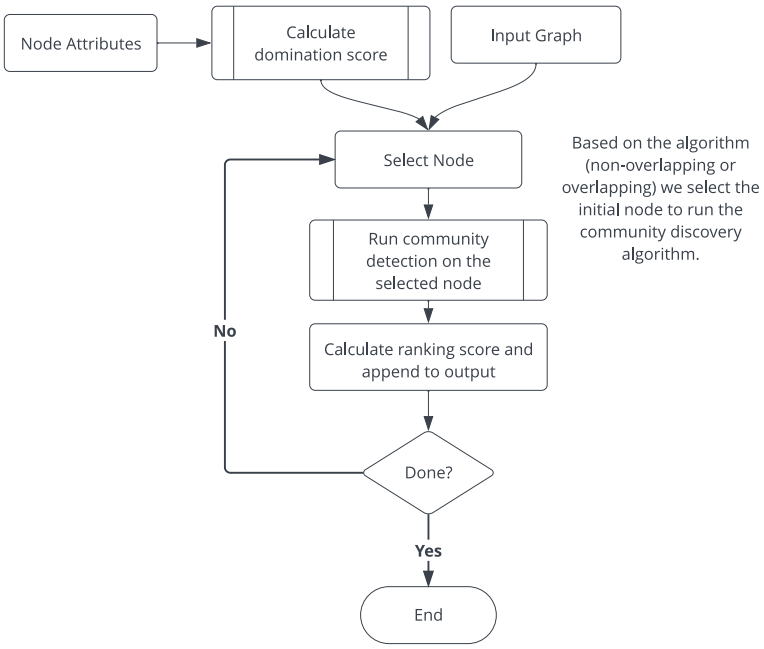


Fig. 2. Data flow of the DomiCom algorithm. The process starts by calculating domination scores from node attributes. A seed node is then selected from the input graph, and the community detection algorithm is run. A ranking score is calculated for the new community, and the process iterates until the desired number of communities is found

networks it reflects researcher *influence*, in social networks it captures *activity* or *visibility*, while in biological networks it may indicate *functional centrality*. Beyond ranking individual nodes, the distribution of domination scores strongly affects the resulting communities: skewed distributions yield star-like structures around a few dominant nodes, whereas balanced distributions produce denser, multi-centered groups.

Briefly, the formation of communities is based on two different concepts: a) the *domination score* of the nodes participating in the community, which is calculated using node attributes, and b) the *structural coherence* of the community, which is computed by means of the *core decomposition* process. The domination score of a node represents the power of the node to dominate other nodes based on the values of the attributes. For example, node u with $vec(u) = [5, 6]$ dominates another node v with $vec(v) = [1, 2]$. The core decomposition process identifies subgraphs that are highly connected and are characterized by structurally coherent. More details of these concepts will be given in the subsequent sections.

In the sequel, we provide a simple example to introduce the methodology of the proposed approach. Figure 1 illustrates a high-level overview of our approach using a small example graph, while Figure 2 shows the data flow of the proposed approach. The first step in our algorithm is to compute the domination score for

Table 1. The domination score of nodes in the example

node ID	**dom. score**	**node ID**	**dom. score**
14	21	7	8
8	20	3	7
21	18	1	5
16	17	6	4
12	16	20	4
13	16	18	1
17	13	4	1
2	12	10	1
5	11	19	1
11	11	22	1
9	10	23	0
15	9		

each node. Based on node attributes, the domination score for each node is given in Table 1.

In this example, our objective is to detect communities at distance one (i.e., 1-hop). However, our techniques also work for larger hop values, as well as in cases where random walks are used to generate node context [18]. Starting from the node with the highest domination score, we induce the subgraph of distance 1 as shown in Figures 3a and 3b. This results in a subgraph that contains the following set of nodes {7, 8, 13, 14, 15, 16, 17, 18}. Next, we compute the maximum core, which in our example is the subgraph shown in Figure 3b, consisting of nodes {7, 8, 13, 14}. Depending on the method we use to form communities, i.e., non-overlapping or overlapping, we continue until the top c communities are finally formed. Once the process is completed, the communities are ranked using a specialized ranking function and returned to the user.

In Figure 4, we illustrate two different results of our dominant community detection algorithm for the non-overlapping and overlapping case. The difference between the non-overlapping and the overlapping case is that in the non-overlapping case the nodes can be part of only one community, whereas in the overlapping case each node may participate in more than one community, offering greater flexibility.

Overlapping community/group search can have an advantage in several scenarios, as for example: (i) Modern researchers often straddle multiple fields, e.g., someone working at the intersection of machine learning and healthcare informatics. Assigning them to just "Computer Science" or just "Health Sciences" loses critical nuance. Therefore, overlapping has the advantage of capturing multiple research domains per researcher. (ii) In topic modeling of documents, it is unrealistic that a document belongs only to one topic. Here, overlapping allows multi-topic attribute search. (iii) In the case of social circles search, it is unrealistic that friends belong only in distinct groups. Therefore, overlapping can capture people in multiple social circles.

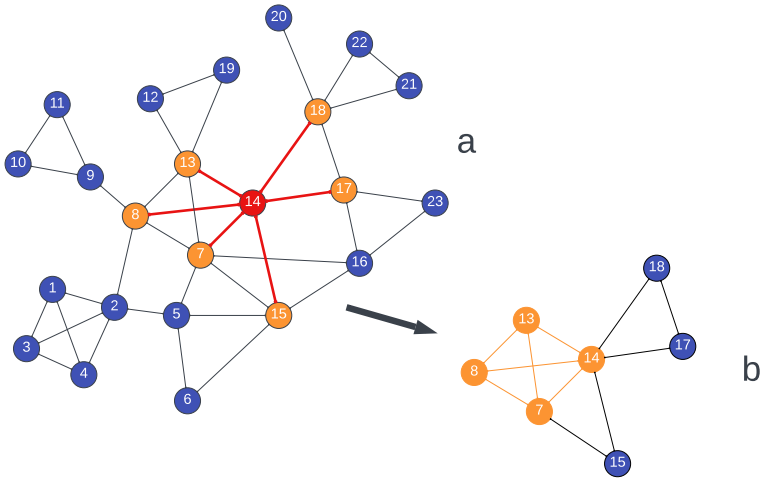
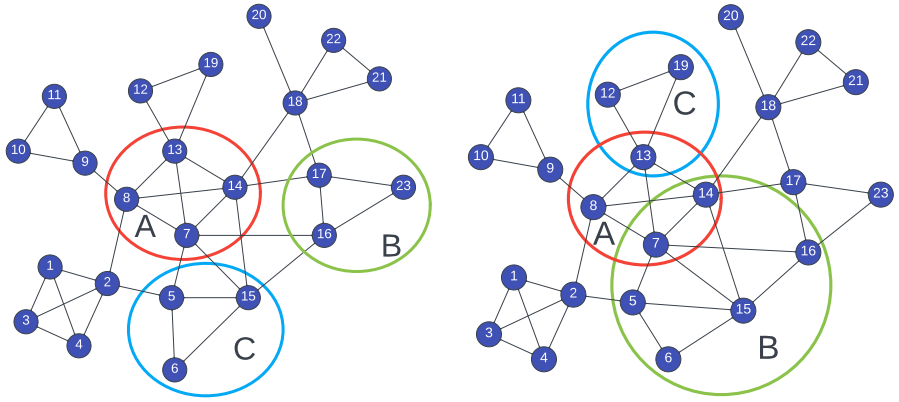


Fig. 3. The two-step community formation process. (a) First, the 1-hop egonet is generated from a seed node (node 14, in red) and its immediate neighbors (in orange). (b) The maximum k-core is then computed from this egonet, yielding the final, densely connected community of nodes 7, 8, 13, 14



(a) non-overlapping (b) overlapping

Fig. 4. Comparison of non-overlapping and overlapping community structures. (a) In the non-overlapping case, each node belongs to only one community. (b) In the overlapping case, nodes (such as 7, 13 and 14) can be members of multiple communities

According to the definitions and properties of each method (overlapping and non-overlapping search), we provide some practical guidelines:

– Use non-overlapping search when the domain implies exclusivity (e.g., membership in mutually exclusive categories), or overlap may exist but is minor and not relevant to your goals.

– Use overlapping search when nodes naturally belong to multiple groups, or the understanding of the overlap itself is valuable (bridging nodes, interdisciplinarity), or you need richer modeling (e.g., weighted memberships, soft boundaries), or you aim to discover structural overlap phenomena like community bridges, layered hierarchies, or fuzzy boundaries.

A practical advice is first to start simple and run a non-overlapping method to get rough intuition. This can be our baseline. Then, we may inspect "unnatural" nodes: for example, nodes with high modularity but ambiguous affiliation may be a hint to try overlapping models. Sometimes, a hybrid approach can be valuable, thus we may first cluster non overlapping and then allow overlapping sub communities within large clusters. In each case, the results must be interpreted carefully. Domain experts can guide or validate the results.

Contributions. The main contributions of our paper are briefly described as follows:

– Many of the existing works consider graphs with specific assumptions and domains on their attributes. Our methodology can be applied to attributed graphs without any assumption for node attributes.
– Current state-of-the-art methods define and use domination relationships between entire communities, and they are taking into account aggregations based on node attributes. Moreover, different algorithms are required for different dimensionalities. In our method, we use the natural domination relationship between nodes of the graph by taking into account only their original attributes before the extraction of communities. Therefore, the proposed algorithms can be used on data with any dimensionality.
– We propose efficient algorithms for both non-overlapping and overlapping community detection on attributed graphs. This flexibility allows for choosing the most suitable technique based on the application. We note that existing algorithms are mainly focused on the non-overlapping case.
– As an alternative solution, we also propose more efficient *approximation* algorithms for both non-overlapping and overlapping community detection. These algorithms enable to trade accuracy for speed, meaning that the final result may not be 100% accurate, but the computations require significantly less time.

We demonstrate the efficiency and effectiveness of the proposed approach by using real-life network data with node attributes.

Roadmap. The rest of the article is organized as follows. Section 2 presents related work in the area. The proposed methodology is described in detail in Section 3. Performance evaluation results based on real-life data sets are given in Section 4, whereas Section 5 concludes our work and briefly discusses interesting future research directions.

2. Related Work

In this section, we discuss briefly related work in the area, focusing mainly on research that focuses on community detection or community search in attributed graphs, highlighting also research efforts that use preference-based techniques.

2.1. Community detection in attributed graphs

In [6] a community search methodology in attributed graphs is proposed. The supported graphs contain textual attributes and the returned results are communities that satisfy both structural cohesiveness and keyword cohesiveness.

Another related research [13],[14] introduces a community model (called k-influential community) based on the concept of k-core that takes into account the node's influence. Improved online search algorithms are also proposed for the influential community search variation [4],[2].

Another aspect in the literature is to find k-truss communities, instead of k-core. In [7] a k-truss community model is proposed with several efficient community search algorithms. The proposed algorithms can be applied also in a dynamic graph setting with frequent insertions and deletions of graph vertices and edges. An extension of the influence-based works has been proposed in [22] where a novel scoring function and algorithm is designed, that tries to find the attributed k-truss community by maximizing the attribute and influence relevance scores.

The works of [13],[14], [4],[2], [22] consider only one numerical attribute, i.e. the node's influence, or they are based on scoring functions related to that specific attribute.

In [9] a greedy-based algorithmic framework is proposed, which detects a closest truss community with small diameter. The method iteratively removes the furthest nodes from the graph in order to support the small diameter, and achieves approximations to the optimal solution. In [8] an extension is proposed to iteratively remove nodes with the least popular attributes, and shrink the graph into an attributed k-truss community.

In [10] efficient algorithms for searching k-truss communities are also proposed. These algorithms can be applied in dynamic graphs with frequent insertions and deletions of vertices and edges, and can also support k-truss community search over massive graphs which cannot entirely fit in main memory.

The work of [21] propose an efficient algorithm for complex network community partitioning by combining fitness optimization with community similarity.

In [5] the authors present an algorithm for detecting overlapping communities base on node similarity.

All the above works are substantially different from our study as they are applied on graphs of specific domains on their attributes, or they do not consider preference-based values in their calculations.

2.2. Preference-based community detection

In [12] the skyline community search problem is presented and studied, where the skyline operator is applied over all communities that are derived from a multi-valued graph. The domination relationship between two communities is defined by

Table 2. Key differences between our method and most related methods

Feature	**Proposed Method (Ours)**	**Methods of [12], [11], [24]**
Domination Scope	Between individual **nodes** using original attribute values	Between **entire communities**, often using aggregated attributes
Dimensionality Handling	Single algorithm handles **all dimensionalities**	Requires **different algorithms** for different dimensionalities
Use of Skyline Operator	**Not used**; relies on domination score	**Used** to identify key communities
Community Cohesiveness (k-core)	Considered as a **second step** in the algorithm	Not always explicitly incorporated
Community Type	Supports both **overlapping and non-overlapping** communities	Often restricted to **non-overlapping** communities
Top-k Community Retrieval	**Supported** via user-defined k parameter	**Not supported**; number of communities is uncontrolled

comparing the corresponding vectors that are derived from the minimum values for all their nodes in each dimension i (f_i-values). The authors apply specific criteria for their proposed community detection approach: a resulted community H must be a k-core subgraph, there does not exist any induced subgraph that is also k-core and dominates H, and there does not exist a subgraph of H that is also k-core and it has the same f_i-values. The proposed algorithms handle cases with specific attribute dimensionalities ($d = 2, 3$), and a different algorithm is proposed for $d > 3$. Moreover, a dimensionality reduction-based algorithm is also proposed.

An extension is presented in [11], where for the cases of $d = 3$ and $d > 3$ more details are presented, and additional pruning rules are proposed, improving the overall performance.

In [24], more performance improvements are proposed to algorithms of [12] and [11], taking into account dimensionality reduction and pruning strategies. Additional criteria, like the maximum entropy model, are applied in order to remove insignificant attributes and to improve the efficiency of community search strategies. The basic algorithm MICS and the improved algorithm P-MICS are proposed, which can be applied in larger graphs.

In our study, which is very different compared to [12], [11], [24], we apply the natural domination relationship between the graph nodes, by taking into account their original attribute values. We do not define domination relationships between entire communities, or taking aggregations (like minimum) of node attributes. Thus, in our approach, there is no need to have different algorithms for different dimensionalities. Moreover, we do not apply the skyline operator, but instead we use the domination score in order to retrieve important nodes. We are also taking into account the cohesiveness (k-core communities), but as a second step to our algorithms. In addition, we study the community detection problem with both overlapping and non-overlapping communities. This flexibility was not possible using the previously proposed approaches. Finally, our technique supports top-k retrieval of the best communities, where k is a user-defined parameter. The skyline-based community detection algorithm does not support this parameterization, and

the number of returned communities cannot be controlled by the user. This means that, depending on the dataset, the number of communities may be too small or too large.

Table 2 presents a summary of key differences between our method and the most related prior work to ours ([12], [11], [24]). Note that, as these prior methods are very similar, the key differences hold for all. To the best of the authors' knowledge, this work is the first to study the graph community discovery problem by using the domination score of nodes based on multi-valued node attributes, supporting both non-overlapping and overlapping community detection.

3. DomiCom: The Proposed Approach

In this section, first we present some fundamental topics related to our proposed methodology and then we study in detail the algorithmic techniques supporting the proposed approach for preference-based community detection.

3.1. Background

The concept of dominance has been used in many research works related to multi-criteria optimization and preference-based query processing [17]. Dominance refers to a property that decides if an object p is *better* than another object q, where objects are represented as multi-dimensional points (vectors). We assume, without loss of generality, that in each dimension, large attribute values are preferable.

Definition 1 (dominance). *An object* $p = (p.x_1, p.x_2, \ldots, p.x_d) \in \mathbb{R}^d$ *dominates another object* $q = (q.x_1, q.x_2, \ldots, q.x_d) \in \mathbb{R}^d$*, i.e.,* $p \succ q$*, when:* $\forall i \in \{1, \ldots, d\} : p.x_i \geq q.x_i \wedge \exists i \in \{1, \ldots, d\} : p.x_i > q.x_i$*. This means that* p *is as good as* q *in all dimensions, and it is strictly better than* q *in at least one dimension.*

The previous definition of dominance holds also when there are dimensions in which lower attribute values are better (e.g., age, cost), if we transform their scales so that higher values represent better outcomes. The most common transformation to preserve the total ordering, is to subtract the values from their global max: $x' = x_{max} - x$. In cases that in a dimension there are ordinal attribute values, there is a clear natural ordering or magnitude (which value is considered better), and the dominance definition can also be applied. Therefore, we can encode the ordinal values with appropriate numbers (integers or reals) by preserving this natural ordering (e.g. low=1, medium=2, high=3). In cases that in a dimension there are q distinct categorical attribute values without order (there is not any kind of ordering or magnitude), we can replace this dimension with q binary dimensions, one for each categorical value. Therefore, if a node belongs to a specific category, we assign the value 1 in the corresponding binary dimension and 0 in the rest binary dimensions. Moreover, we can also encode nodes that may belong to multiple categories by assigning 1 in all corresponding binary dimensions (extremely useful for overlapping search).

Based on the concept of dominance, one can identify points that are *not dominated*. These points are considered as the *best points* we can get from the multi-dimensional dataset. In the database literature, these *best* points are known as the *skyline* of the dataset [3]. Based on the definition of the skyline, all points in the skyline are equivalent, since there is no *ranking* concept for skyline points. The concept of *domination score* has been introduced in [23] in order to rank points based on their dominance power. Formally we have:

Definition 2 (domination score). *The domination score of a point p, dom(p) is defined as: dom(p) = $|\{q \in D : p \succ q\}|$. A top-k dominating query returns the k objects with the highest domination scores.*

The domination score is an essential ingredient of our approach. We use the domination score to determine the order in which we discover communities but also to compute the score of each community. The multi-dimensional points correspond to the attribute vectors of the graph nodes. There are in general two different techniques to compute the domination score: *i*) exact and *ii*) approximate. To enable efficient computation of the domination score in both cases, a simple *grid-based* approach, firstly reported in [23], is being used. The address space is partitioned into *cells* and each point is assigned to a unique cell based on its coordinates. The number of cells is an important parameter that affects performance.

Exact computation of domination scores. For the exact domination score computation, we need to scan the multi-attribute dataset and assign each point to the grid cell it belongs based on its attribute values. The size of the grid is a parameter that defines in how many parts the specific dimension is divided. For example, if we want to process a two-dimensional dataset we may define a 10×10 grid, which means that the x and y axes are divided in ten equal sections based on the min and max of each attribute (see Figure 5).

This approach does not take into account the data distribution, meaning that if the data concentration is denser in some areas it may impact performance. However, to avoid such a case, we could construct the grid based on the data distribution and place the data points to the grid as evenly as possible. To keep the discussion simple, we use the uniform grid approach. Depending on the definition of the domination score, the process starts from the highest nonempty dominant grid cell and continues towards the lowest by first computing the *base domination score*, i.e. the score that all elements in this grid cell have. To compute each cell's *base domination score* we need to take the sum of the number of elements that are in the d-1 dimension of the current cell.

The next step involves the examination of the points that are either in the same cell and in the same dimension (in case of two-dimensional grid, the same row and column). In the exact domination score algorithm we need to compare each node in the under investigation set (nodes that are on the same cell, row and column, for the two-dimensional example) exhaustively with every other node to get the domination score of that node.

In the example of Figure 5, suppose that we are trying to compute the domination score of a point (n_x) that resides in cell A located at (9,8). We can decide

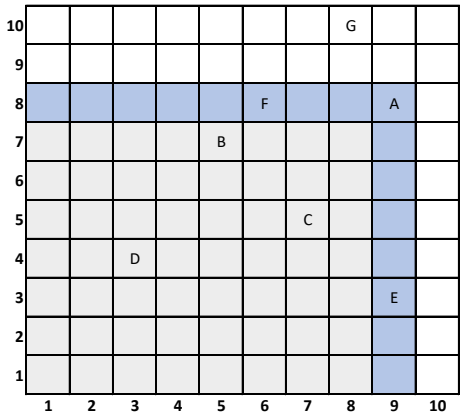


Fig. 5. Illustration of the grid-based approach for domination score computation. For a point located in cell A, it immediately dominates all points in the shaded region (cells B, C, D). Dominance relationships for points in partially dominated cells (E, F) must be calculated individually, while points in non-dominated cells (G) can be ignored

immediately that n_x dominates all points in cells B, C, D and that it does not dominate any node from cell G. Then, we need to calculate the number of points

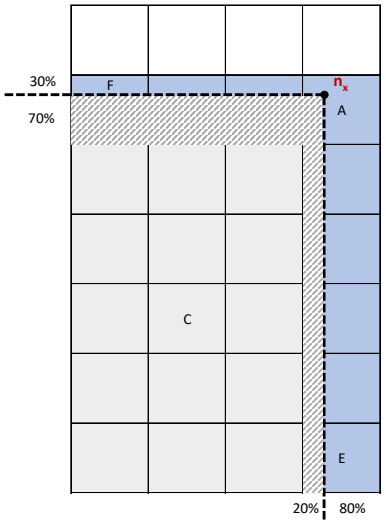


Fig. 6. Approximation of domination score using the piecewise uniformity assumption. For a point n_x in cell A, its dominance over points in partially dominated cells (e.g., F and E) is estimated based on its relative position. Assuming a uniform distribution within each cell, n_x approximately dominates a fraction of points in these cells

that point n_x dominates from the remaining cells that we could not get an immediate answer (cells A, E, and F). The number of points that we need to examine in this second pass depends on the dimensionality of the data set as well as the data distribution. The domination score of point n_x equals the total number of points contained in cells B, C, and D plus the number of points from cells A, E, and F dominated by n_x.

Approximate computation of domination scores. In this case, the domination score is approximated only for the part where we were unable to get an immediate result. To approximate the domination score, we assume that all points in the related cells are distributed uniformly (piecewise uniformity assumption). Thus, the coordinates of the point n_x determine the number of points dominated. In the example shown in Figure 6, the point n_x approximately dominates 70% of cell F regarding the y axis, 20% of cell E regarding the x axis, and therefore approximately 14% of the total number of points residing in cell A.

The use of the approximate domination score computation results in faster processing, and as we illustrate in the performance evaluation section, this approximation does not affect the accuracy of the final results significantly. The outline of the exact and approximate domination score computation is shown in Algorithm 1.

Illustrative Example

To compute the domination scores, we created a toy dataset, shown in Table 3, based on the grid example of Figure 5. Then we executed the Algorithm 1 step by step to compute the exact domination scores of each node. The computation proceeds as follows:

1. **Grid assignment:** Each attribute is normalized to a 10×10 grid using its minimum and maximum values. Each node is then mapped to a grid cell (n_1, n_2). For example, node $A = (87, 75)$ is placed in cell $(9, 8)$, while node $D = (22, 37)$ is placed in cell $(3, 4)$.
2. **Base score:** A node immediately dominates all nodes in strictly dominated cells (i.e., cells to the right, below, or both). For instance, node A dominates all points lying in cells that are to its lower-left region.
3. **Refinement:** For nodes that lie in the same row, column, or partially dominated cells, pairwise comparisons are carried out to confirm whether one node dominates the other.
4. **Final score:** The domination score of a node is the sum of its base score and refinements. For example:
   - Node $A = (87, 75)$ dominates B, C, D, E, and $F \Rightarrow$ score $= 5$.
   - Node $G = (78, 94)$ dominates B, C, D, and F, but not $A \Rightarrow$ score $= 4$.
   - Node $F = (55, 73)$ dominates B and $D \Rightarrow$ score $= 2$.
   - Node $C = (67, 44)$ dominates $D \Rightarrow$ score $= 1$.
   - Node $B = (43, 67)$ dominates $D \Rightarrow$ score $= 1$.
   - Node $D = (22, 37)$ cannot dominate anyone $\Rightarrow$ score $= 0$.
   - Node $E = (85, 24)$ cannot dominate anyone $\Rightarrow$ score $= 0$.

Algorithm 1: Domination Score Computation

```
// points is any array that holds the nodes attributes
Data: points
// These are the parameters of the algorithm
// approximate: boolean, if true we use approximate domination score
   calculation
// stats: min and max of each dimension
// gsize: grid size, defines in how many parts each dimension is
   divided
Input: approximate, stats, gsize
Result: domination score of each point
1  begin
       // distribute points into grid cells
2      foreach p in points do
           // translate point to grid cell key
3          key ← translate(p, stats)
           // append point to corresponding cell
4          append(grid[key]) ← p
       // get all non-empty grid keys and assign them to an array
5      grid_keys ← unique grid keys that each correspond to a cell with at least
         one point
       // sort grid keys in descending order
6      sort(grid_keys)
7      for i in grid_keys do
8          for j in grid_keys[i :] do
9              if j is less than i then
10                 base_score += number of points in grid[j]
11             else
12                 append(process_next) ← the points in grid[j]
13         for n in grid[i] do
14             if approximate then
15                 score ← approximate domination score of n over process_next
                     points
16             else
17                 foreach l in process_next do
18                     if n dominates l then
19                         score += 1
20             domination[n] = base_score + score
```

This staged process illustrates how raw attribute values are translated into grid coordinates, and then systematically transformed into domination scores.

Table 3. Toy dataset (nodes A–G) with attribute values, grid-cell assignments (10×10 grid), and exact domination scores

Node	**Attribute 1**	**Attribute 2**	**Grid Cell** (n_1, n_2)	**Domination Score**
A	87	75	(9,8)	5
B	43	67	(5,7)	1
C	67	44	(7,5)	1
D	22	37	(3,4)	0
E	85	24	(9,3)	0
F	55	73	(6,8)	2
G	78	94	(8,10)	4

3.2. Dominant Community Detection

In this section, we present details for the definition of the community score and the algorithms that perform non-overlapping and overlapping community detection. Table 4 includes the most frequently used symbols for the next discussion.

Table 4. Frequently used symbols

Symbol	Description
G	The undirected graph
δ	Maximum node degree in G
d	Number of dimensions of node vectors
DSA	Domination score array sorted in descending order
MAX_{dom}	Global maximum domination score of the current dataset
N	Total number of nodes in the community
n	Current node
k	Number of top-k nodes to be processed
h	Number of hops of the community to be formed
c	Number of communities reported

The domination score can help us rank the graph nodes regarding its attributes. The higher the domination score, the better attributes the node has compared to the other. Since we want to form communities that include nodes with high domination score the best possible option would be to have nodes that are closer to the maximum domination score of the entire graph. To achieve that, we compute a ranking score for each community based on Equation 1.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} |dom_i - MAX_{dom}|^2}{N}} \tag{1}$$

For a community with N nodes we compute the deviation (σ) from the maximum domination score of the entire graph (MAX_{dom}). With this expression we

are able to evaluate a community and see how close or far away the participating nodes are from the global maximum domination score. If σ is small then all the community nodes are closer to the maximum and so the community has a great value.

We are essentially trying to capture the relationship between the global maximum domination score and the domination score of each node in the community. Communities that have nodes with wide a range of domination score will rank lower that communities with narrower range and same maximum. For example in Figure 7, community B will have better ranking score than A, because the nodes of B are closer to the global maximum. For a community C with N nodes, if

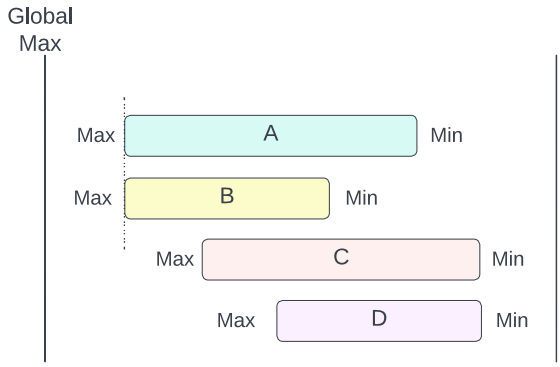


Fig. 7. An illustration of the community ranking score calculation σ. Each bar represents the range (Min to Max) of domination scores of nodes within four different communities (A, B, C, D). The ranking function rewards communities whose scores are tightly clustered and close to the global maximum domination score. For example, Community B would receive a better ranking than Community A because its nodes have a smaller deviation from the global maximum

$dom_i, i = 1, 2, ..., N$ are the corresponding domination scores of the nodes of C, the final score of C is defined as:

$$norm(C) = \frac{1}{MAX_{norm}\sqrt{\frac{1}{N}\sum_{i=1}^{N}|MAX_{dom} - dom_i|^2}} \tag{2}$$

where MAX_{dom} is the global maximum domination value from all nodes and $maxnorm$ is the global maximum $norm$ value (from all communities). The larger the value of $norm(C)$, the better the community. Note that, the value of MAX_{norm} is used only for the normalization of score values in $[0, 1]$.

3.3. Comparison with other ranking functions

We selected the deviation-based ranking function because it captures our goal of finding communities where nodes are not just strong on average, but are consistently dominant. The function evaluates a community by measuring how far its node domination scores deviate from the global maximum score. A small deviation signifies that all members of the community have scores tightly clustered near this peak value. This approach rewards groups that are uniformly powerful and ensures that a highly-ranked community is significant in the context of the whole network rather than just having a high internal average.

This method is more robust than simpler alternatives. A ranking based on the average score, for example, can be misleading, as a few nodes with exceptionally high scores could inflate the average and mask the presence of much weaker members. Our deviation-based approach provides a more balanced evaluation by considering the scores of all nodes while specifically rewarding the cohesive dominance that is central to our definition of a top-k dominant community.

To demonstrate the effectiveness of our deviation-based ranking function, we compared it with an average-based ranking function using the same dataset. We applied both ranking methods to the same set of communities and analyzed the spread of their scores. The results in Figure 8, show that the deviation-based ranking function produces a tighter spread of domination score distribution, indicating that it effectively identifies communities with consistently high domination scores. In contrast, the average-based ranking function resulted in a wider spread of domination scores which suggests that it may include communities with significant internal disparities. Figures 9 (a and b) illustrate the scatter plots of the average-based and deviation-based ranking functions, respectively. The deviation-based ranking function (Figure 9b) shows a more concentrated distribution of spread, highlighting its ability to identify communities with tighter domination scores.

There are two main approaches regarding the community detection algorithms: a) the non-overlapping and b) the overlapping approach. In the non-overlapping case, the nodes that participate in a community cannot be part of another, whereas in the overlapping case, nodes can participate in multiple communities. Our methodology supports both cases.

In our community detection algorithm, we start from the node with the highest domination score and stop when we have processed the top-k nodes. With each node we try to form a community by taking its neighbors with distance h (number of hops) and then applying the *maximum core* [15] operation. Next we compute the ranking score of the community and add it to the output list.

Recall that the maximum core of a graph G is the maximum induced subgraph S of G, where each node in S has a degree at least r, where r is the maximum possible value. By using this approach, not only we get nodes that are important based on the domination score, but also the community is cohesive.

Depending on the current case (non-overlapping or overlapping), we can point out some characteristics. In the non-overlapping case the high domination score nodes will have the opportunity to include nodes first to their community. On the other hand the overlapping method does not have this issue because the same node can be part of multiple communities. Another important difference between the

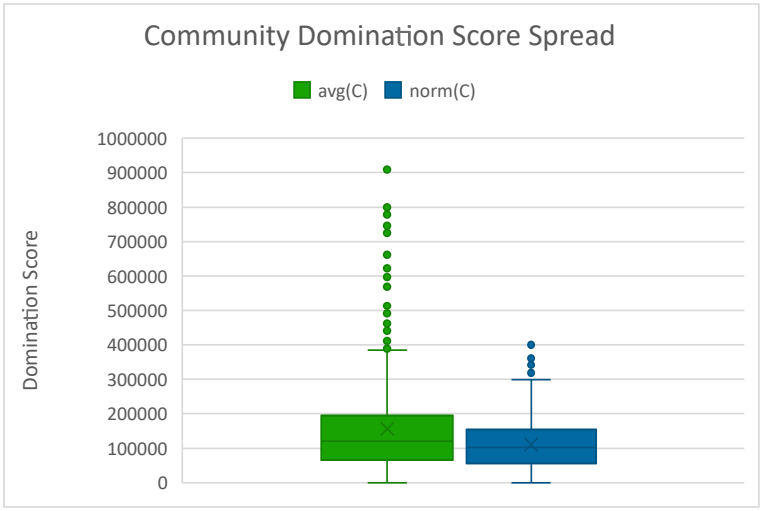
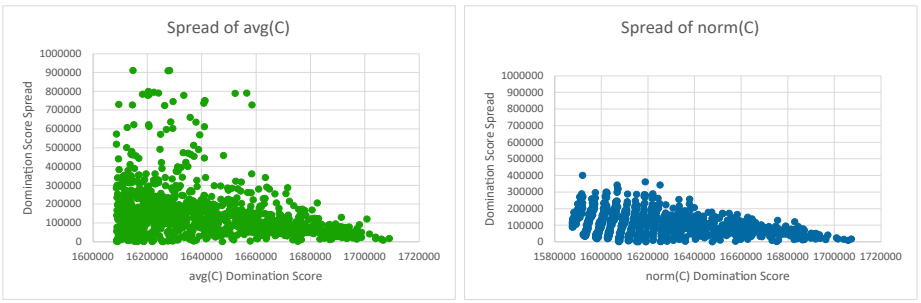


Fig. 8. Distribution of domination scores of the average-based (avg(C)) versus the deviation-based (norm(C)) ranking function. The deviation-based function yields tighter domination score distribution with fewer outliers, indicating its effectiveness in identifying communities with consistently high domination scores



(a) (b)

Fig. 9. Scatter plots visualizing the domination score spread of the two ranking functions (a) The average-based (avg(C)) function (b) The proposed deviation-based function (norm(C))

two cases regarding the computation is that in the non-overlapping case we have to compute all communities sequentially from top to bottom, because we exclude nodes that are part of a community, while the overlapping method does not have this restriction. This means that if we want to get the community that its initial node is in position 50, in the non-overlapping case we have to find all 49 previous communities, while in the overlapping case we can directly get the community we are interested in.

Before we start the process we have to define some parameters. a. the number of hops, b. how many nodes we will be processed and c. which method we will

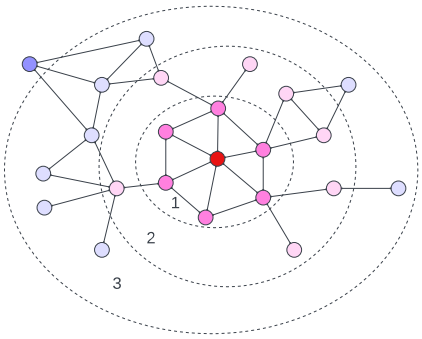


Fig. 10. Illustration of h-hop neighbors for community formation. Starting from a seed node (red), a candidate community includes all nodes within h hops (distance). Nodes at 1-hop, 2-hops, and 3-hops are shown, demonstrating how h controls the initial scope of the search

use. The number of hops is the distance from the initial node, and we will use this setting to form our basic community. Figure 10 depicts an example of a graph and the nodes that will be included using different number of hops. It is evident that as the number of hops increases, more nodes are included in the subgraph which will finally form the community.

3.4. Non-overlapping Communities

Having the exact domination scores of all nodes computed we start the community detection process from the node with the highest domination score. Based on this initial node we will try to discover a community by taking the neighbors of distance h, i.e. the egonet. From this subgraph we filter out nodes that have been visited previously (i.e. are part of another community discovered before this one). Because we want the discovered community to have dense structure, we proceed in another operation and get induced subgraph of the max core. At this point the structure of the community is complete and what we want next is to calculate the ranking score of the community and append it to the output list. All the nodes from the recently discovered community will be excluded from future communities, since this is the non-overlapping case. Once we process the top-k nodes from our sorted domination score list, we stop, sort the output by the ranking score and return the results. We expect that the communities at the top of this list will have nodes with high domination score and dense structure. The outline of the non-overlapping case is shown in Algorithm 2.

Complexity Analysis. Let $n = |V(G)|$ and $m = |E(G)|$. Also, let δ denote the maximum node degree in G, and c the desired number of top communities.

Algorithm 2: Non-overlapping algorithm

Input: G, DSA, max_dom, k, h
Output: *results*

```
// Initialize variable to hold the visited nodes
```
1 *visited* ← {}

```
// Assign the top-k nodes from the sorted domination score array
```
2 top_k ← DSA[0:k]

3 **foreach** n *in top_k* **do**

```
    // Skip if node has been visited
```
4 **if** $n \in$ *visited* **then**
5 continue

```
    // Get the induced subgraph starting from n and include all
       neighbors with distance h from the graph G
```
6 $e \leftarrow$ egonet(G, n, h)
7 filter out all nodes from $e \in$ *visited*
8 *comm* ← max core induced subgraph of e
9 add all nodes of *comm* to *visited*
10 *norm* ← calculate ranking score from *comm*
11 append *comm* and *norm* to *results*

12 sort *results* based on norm in descending order
13 return *results*

Now let us derive the complexity of Algorithm 2. The max domination scores have been calculated from Algorithm 1, thus lines 1,2 have a total complexity of $O(n+m) + O(nlogn) + O(n) = O(nlogn + m)$

The main for loop (lines 3-11) is executed till we found the desired top communities (c times). In lines 6-11 we derive a sub-graph from a starting node with h hops. Therefore, in the worst case the number of nodes in this graph are: $1 + \delta + \delta^2 + ... + \delta^h = \frac{\delta^{h+1}-1}{\delta-1}$ (if all derived nodes have the maximum degree). Thus, we have a complexity of $O(\delta^{h+1})$. The max core step has a complexity of the sum of nodes and edges in the derived sub-graph, which in the worst case that is a complete graph returns $O(\delta^{h+1}(\delta^{h+1}-1)/2) = O(\delta^{2h+2})$. The remaining calculations do not contribute more to the complexity. Therefore, the overall complexity of the algorithm is: $O(c\delta^{2h+2} + nlogn + m)$.

Theorem 1. *If community C_b is discovered after community C_a then it always holds that: $norm(C_a) \geq norm(C_b)$, according to the defined norm measure.*

Proof. Let $doma_i, i = 1, 2, ..., N_a$ the domination scores of the N_a nodes of the community C_a, and $domb_j, j = 1, 2, ..., N_b$ the domination scores of the N_b nodes of the community C_b.

In the case of non-overlapping communities and without loss of generality, we have the following domination scores order:

$doma_1 \geq doma_2 \geq ... \geq doma_{N_a} \geq domb_1 \geq domb_2 \geq ... \geq domb_{N_b}$
$\Leftrightarrow MAX_{dom} - doma_1 \leq MAX_{dom} - doma_2 \leq ... \leq MAX_{dom} - doma_{N_a} \leq$
$\leq MAX_{dom} - domb_1 \leq MAX_{dom} - domb_2 \leq ... \leq MAX_{dom} - domb_{N_b}$
where MAX_{dom} is the global maximum domination value.

As all domination scores are less than or equal to MAX_{dom}, all the above differences are non-negative numbers, thus we can take their squares without changing any order:
$|MAX_{dom} - doma_1|^2 \leq |MAX_{dom} - doma_2|^2 \leq ... \leq |MAX_{dom} - doma_{N_a}|^2 \leq$
$\leq |MAX_{dom} - domb_1|^2 \leq |MAX_{dom} - domb_2|^2 \leq ... \leq |MAX_{dom} - domb_{N_b}|^2$
Therefore, it definitely exists a real non-negative number L, such that:

$$\begin{aligned} |MAX_{dom} - doma_1|^2 \leq |MAX_{dom} - doma_2|^2 \leq ... \\ ... \leq |MAX_{dom} - doma_{N_a}|^2 \leq L \leq |MAX_{dom} - domb_1|^2 \leq \\ \leq |MAX_{dom} - domb_2|^2 \leq ... \leq |MAX_{dom} - domb_{N_b}|^2 \end{aligned} \tag{3}$$

From Inequality 3 due to transitivity property for the $doma_i, i = 1, 2, ..., N_a$ values we derive:
$|MAX_{dom} - doma_1|^2 \leq L$
$|MAX_{dom} - doma_2|^2 \leq L$
...
$|MAX_{dom} - doma_{N_a}|^2 \leq L$
and by using summation we get
$\sum_{i=1}^{N_a} |MAX_{dom} - doma_i|^2 \leq N_a L \Leftrightarrow \frac{1}{N_a} \sum_{i=1}^{N_a} |MAX_{dom} - doma_i|^2 \leq L.$
Therefore, it holds that:

$$\sqrt{\frac{1}{N_a} \sum_{i=1}^{N_a} |MAX_{dom} - doma_i|^2} \leq \sqrt{L} \tag{4}$$

With a similar way, from Inequality 3 due to transitivity property for the $domb_j, j = 1, 2, ..., N_b$ values we derive:
$L \leq |MAX_{dom} - domb_1|^2$
$L \leq |MAX_{dom} - domb_2|^2$
...
$L \leq |MAX_{dom} - domb_{N_b}|^2$
and by summing we derive:
$N_b L \leq \sum_{j=1}^{N_b} |MAX_{dom} - domb_j|^2 \Leftrightarrow L \leq \frac{1}{N_b} \sum_{j=1}^{N_b} |MAX_{dom} - domb_j|^2,$
Therefore, it holds that:

$$\sqrt{L} \leq \sqrt{\frac{1}{N_b} \sum_{j=1}^{N_b} |MAX_{dom} - domb_j|^2} \tag{5}$$

By combining Inequalities 4 and 5 we derive:
$\sqrt{\frac{1}{N_a} \sum_{i=1}^{N_a} |MAX_{dom} - doma_i|^2} \leq \sqrt{\frac{1}{N_b} \sum_{j=1}^{N_b} |MAX_{dom} - domb_j|^2}$
$\Leftrightarrow \frac{1}{\sqrt{\frac{1}{N_a} \sum_{i=1}^{N_a} |MAX_{dom} - doma_i|^2}} \geq \frac{1}{\sqrt{\frac{1}{N_b} \sum_{j=1}^{N_b} |MAX_{dom} - domb_j|^2}}$

$$\Leftrightarrow \frac{1}{MAX_{norm}\sqrt{\frac{1}{N_a}\sum_{i=1}^{N_a}|MAX_{dom}-doma_i|^2}} \geq \frac{1}{MAX_{norm}\sqrt{\frac{1}{N_b}\sum_{j=1}^{N_b}|MAX_{dom}-domb_j|^2}}$$

where MAX_{norm} is the global maximum $norm$ value.

Therefore, we get that: $norm(C_a) \geq norm(C_b)$, and the proof has been completed.

3.5. Overlapping Communities

The overlapping case is outlined in Algorithm 3, and it is very similar to the non-overlapping case except that in this case we do not exclude any node or keep track of visited nodes. Consequently, nodes are allowed to participate in multiple communities without any restriction.

A variation that could combine the non-overlapping and the overlapping case, is to apply a threshold on the number of communities that each node is allowed to participate.

Algorithm 3: Overlapping algorithm

Input: G, DSA, max_dom, k, h
Output: $results$

```
// Assign the top-k nodes from the sorted domination score array
```
top_k ← DSA[0:k]
foreach n *in top_k* **do**
```
    // Get the induced subgraph starting from n and include all
       neighbors with distance h from the graph G
```
    e ← egonet(G, n, h)
    $comm$ ← max core induced subgraph of e
    $norm$ ← calculate ranking score from $comm$
    append $comm$ and $norm$ to $results$
sort results based on norm in descending order
return $results$

4. Performance Evaluation

In this section, we provide performance evaluation results with respect to the computation of the domination score and the community detection algorithms. All experiments have been conducted on a machine with AMD Ryzen 7 PRO 5850U CPU@1.90 GHz with 32GB RAM and SSD drive.

The dataset used in the performance evaluation is the publicly available *Aminer* dataset [3] [20], from which we have used the co-authorship network which contains

[3] https://www.aminer.org/aminernetwork

1,712,433 authors and 4,258,615 collaboration relationships. Each author contains the following attributes: Index ID, Name, Affiliations, Publication Count, Citation Number, H-Index, P-Index with equal A-index of this author, the P-index with unequal A-index of this author, and Research Interests. We have used the following four (4) numerical attributes in our experiments: Publication Count (PC), Citation Number (CN), H-Index (HI) and P-Index (PI) [19].

The purpose of these experiments is to evaluate the performance for different setups for the exact and approximate algorithm, and also to test the error of the approximate domination computation. The algorithm takes as a parameter the grid size, which depends on the dimensionality of the data set. For example if we use a two-dimensional dataset we have to apply also the size of the grid defining in how many parts the two-dimensional data set will be divided to. If we apply a grid size of 10×10, then each dimension is divided into 10 equal parts, resulting in 100 cells. Since we do not normalize the values of each dimension, the resulting cells will not be squares. The grid sizes in our tests had the same size in each dimension, e.g., for the 3-dimensional data sets we tested the cases of (10,10,10), (25,25,25), (50,50,50), (100,100,100), (250,250,250) and (500,500,500).

In the experimental study, we evaluate our community detection algorithms by measuring the performance and community structure for various combinations of the algorithm's parameters, such as total number of communities, number of hops, overlapping or non-overlapping. More specifically, we evaluate the proposed algorithms for both non-overlapping and overlapping cases and for different parameter combinations (1, 2, 3 hops, 5k, 10k, 50k, 100k, 500k communities). The source code for all algorithms covered in this paper is available at `https://github.com/ngeorgiadis/community-detection`.

4.1. Comparison with Existing Techniques

Before diving into details related to the performance of the proposed approach, we present representative results with respect to the comparison between our DomiCom approach and the SkyCom approach, which is also based on dominance relationships. We point out that in general the two approaches are not directly comparable, since they generate communities under different criteria. However, it is interesting to see some results that mainly demonstrate that our proposal is more general and also more effective.

First, we present some results showing the performance of SkyCom for the Aminer data set. Table 5 presents the runtime, the number of communities returned, the average number of nodes per community, and the total number of nodes contained in each core subgraph. Moreover, Table 6 presents the same information where the node attributes follow an anticorrelated distribution, which is expected to pose significant challenges due to the computation of the skyline. Runtime results for the DomiCom approach are given in Table 7.

Based on the values reported in the previous tables, we observe the following:

– The SkyCom approach cannot control the number of communities returned, since it is based on skylines. This means that the number of communities strongly depends on the distribution of attribute values. For example, for the

Table 5. SkyCom results for the AMiner data set

Core	Time (s)	Num. of Results	Avg. Num. of Nodes	Num. of Nodes
1	69.77	5	2.00	1560640
2	46.23	4	3.50	1238105
3	140.91	16	5.81	878167
4	99.41	16	5.75	578194
5	46.44	9	6.11	379976
6	31.14	8	7.75	256078
7	36.60	11	9.00	180276
8	32.06	12	9.83	130556
9	19.71	11	10.91	97208
10	13.70	10	12.10	72943
11	9.20	8	13.63	56046
12	7.63	7	15.00	43093
13	7.22	8	14.38	33588
14	5.26	8	15.00	25917
15	3.35	7	16.00	20200
16	3.11	9	17.00	15848
17	1.61	8	18.00	12295
18	0.90	6	19.00	10399
19	0.58	5	20.00	9246
20	0.59	5	23.00	8233
21	0.84	8	22.00	7073
22	1.31	9	23.00	6313
23	0.53	9	24.00	5831
24	0.56	11	25.00	5118
25	0.53	8	26.00	4699
26	0.33	7	27.00	4290
27	0.50	5	28.00	3716
28	0.21	6	29.00	3408
29	0.17	8	30.00	2980
30	0.13	7	31.00	2719

Table 6. SkyCom results for the anticorrelated distribution

Core	Time(s)	Num. of Results	Avg. Num. of Nodes	Num. of Nodes
50	195.526	19512	51.000	899
40	291.567	23738	41.000	1436
30	721.305	24241	31.001	2719
20	2451.07	20056	21.127	8233
15	7563.35	15157	16.011	20200

AMiner data set we observe that the maximum number of communities returned is 12 and the minimum is 4 (Table 5). On the other hand, for the anticorrelated distribution, we get several thousand communities in the result, which cannot be easily managed (Table 6). In contrast, DomiCom accepts the

Table 7. Exact and approximate domination score computation and total running time (in seconds) for AMiner and Anti-correlated distribution

Aminer Dataset		
	1 hop (sec)	**2 hop (sec)**
Exact domination score	21.84	21.84
Evaluations of the top-50k communities	18.84	195.27
Total time	40.68	217.11
Approximate domination score	7.86	7.86
Evaluations of the top-50k communities	18.84	195.27
Total time	26.70	203.13
Anti-correlated Dataset		
	1 hop (sec)	**2 hop (sec)**
Exact domination score	313.25	313.25
Evaluations of the top-50k communities	18.84	195.27
Total time	332.09	508.52
Approximate domination score	105.12	105.12
Evaluations of the top-50k communities	18.84	195.27
Total time	123.96	300.39

number of returned communities as a parameter, and therefore the number of results may vary based on user preferences.

- In SkyCom, the number of nodes per community is highly restricted by the core value being used (first column). We observe that, in general, the number of nodes per community is around the core value being applied, resulting in communities with a very small number of nodes. This is evident in both the AMiner data set and in the data set with an anti-correlated distribution of node attribute values.
- The performance of SkyCom is highly dependent on the core value used. Higher core values are expected to require less computational cost, because the number of nodes in the selected subgraph is significantly reduced. This is shown in Table 5. On the other hand, in anticorrelated distributed data, the run-time explodes due to the difficulty in the skyline computation. In addition, the number of returned communities is totally uncontrolled and, also, the number of nodes per community is almost the same as the core number. In contrast, DomiCom our which depends primarily on the domination score computation, performs very well even if the attributes follow an anticorrelated distribution (Table 7).
- By default, SkyCom returns overlapping communities and there is no way to force the algorithm to return non-overlapping ones. In contrast, in DomiCom the user may select either the non-overlapping or the overlapping case, depending on the application.

We highlight again that the two algorithms use different definitions for the formation of communities and therefore they cannot be compared directly. We have pointed out some important findings that reveal some of the advantages and disadvantages of SkyCom and DomiCom and presented some representative results. In the following sections, we focus on our approach and present more detailed experimental evaluation results.

4.2. Domination Score Computation

In Figure 11 we observe the domination score histogram. We notice that almost half of the authors have domination score 0 (over 800k authors). We also notice, that the next non-zero domination score is placed in the 0.8M-0.9M bin.

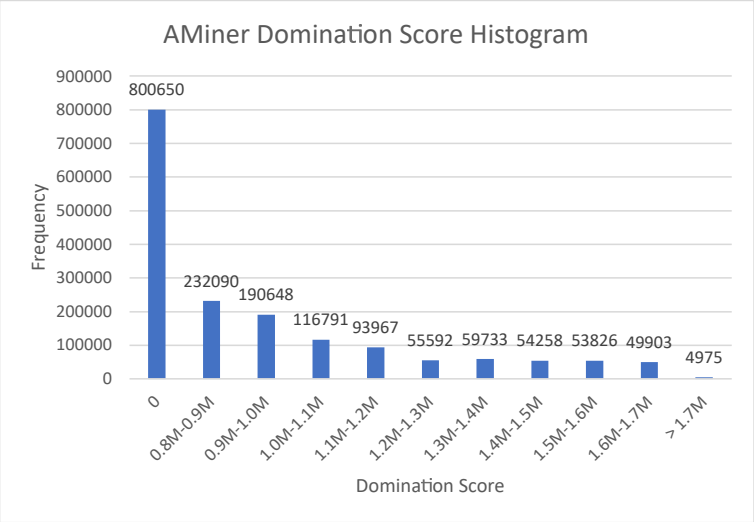


Fig. 11. Domination score histogram of the AMiner dataset

In the domination score performance evaluation, we show how much time is needed to compute the domination score of each author for the entire dataset. There are basically two parameters that we vary, one is the dimensionality and the other one is the grid size. In Figures 12 and 13 we present how the computation time progress as the grid size increases, for the exact and approximate cases respectively.

Figure 14 shows the cost comparison between the exact and the approximate domination score computation algorithms for the 4D case. The same performance has been observed for smaller dimensionalities. We observe that as the grid size increases, initially, the computation time decreases up to a point (grid size of 100 cells per dimension). After this point, the computation cost increases since the number of grid cells increases significantly.

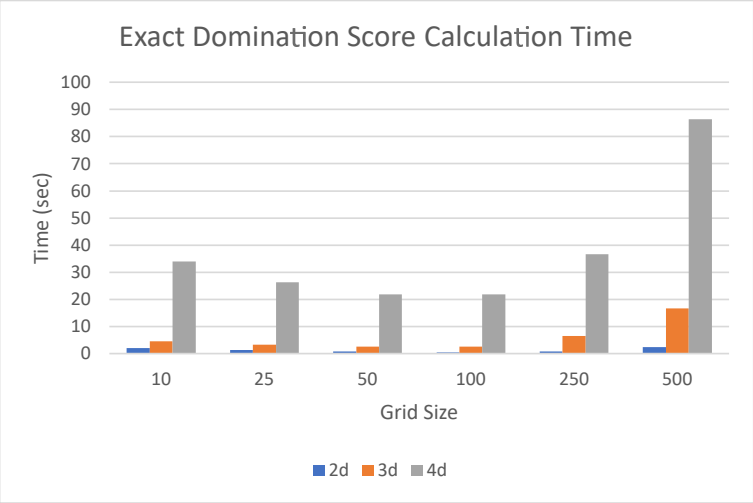


Fig. 12. Exact domination score computation time (sec) for 2, 3 and 4-d data

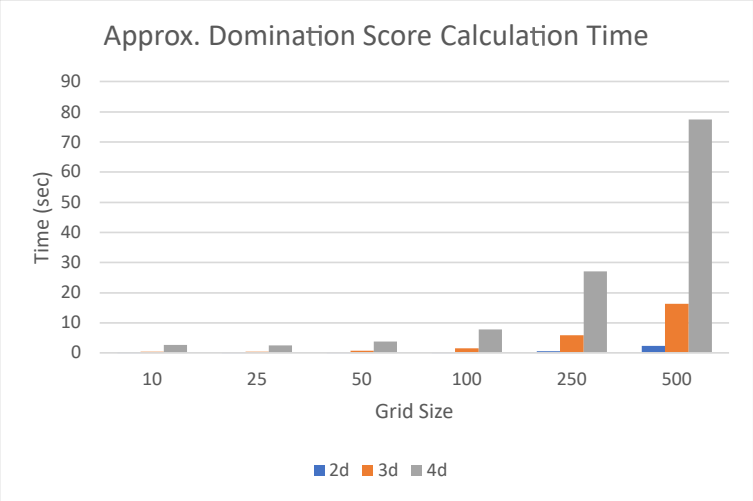


Fig. 13. Approximate domination score computation

When we approximate the domination score, we expect an error compared to the exact computation. This error highly depends on the grid size because as the grid size increases, the approximate score values tend to get closer to the exact ones. In this experiment, we compute the exact and approximate domination scores for grid sizes of 50, 100, 250, 500, and then sort by domination and compute the error on the top-10k nodes.

Figure 15 shows the Average and Root-mean-squared error, respectively. We observe that the error is very small in the 250, 500 grid size cases, whereas it is

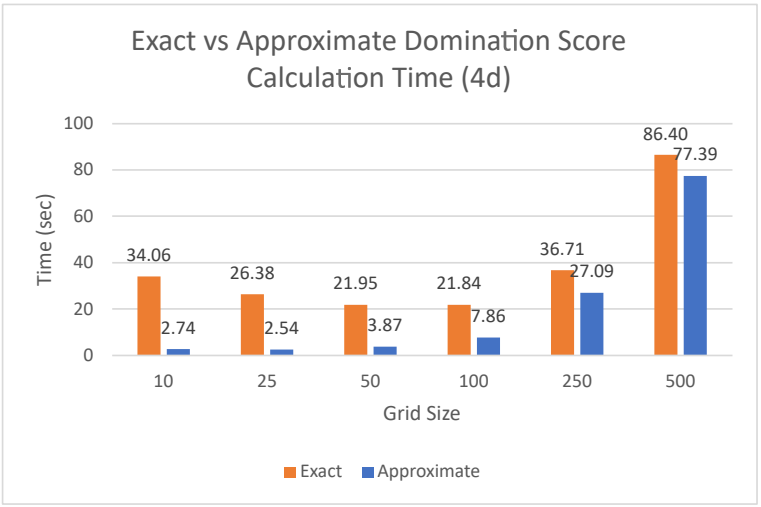
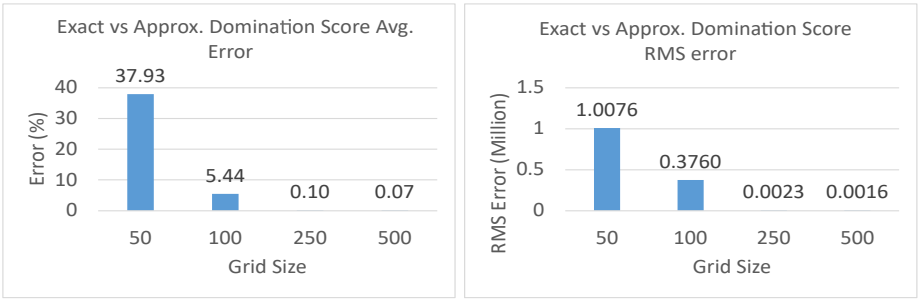


Fig. 14. Exact vs. approximate domination score calculation for 4-d data



(a) Average % error (b) RMSE

Fig. 15. Approximation error of domination scores (a) Average percentage error (b) Root mean squared error (RMSE)

quite large for a grid size of 50. Therefore, a grid size between 100 and 200 is a good compromise between runtime and error.

To summarize these observations, we provide some practical recommendations for selecting between the two variants. Exact computation is more appropriate when the dataset has low dimensionality or when the attribute distribution is highly skewed or anticorrelated, since in such cases approximation may introduce significant inaccuracies. On the other hand, approximate computation is recommended in higher dimensional settings or when runtime is a primary constraint. As shown in Figures 14 and 15, for grid sizes of 100 or larger per dimension the approximation achieves negligible error while offering up a significant speedup. Therefore, the approximate variant is particularly suitable for large-scale or exploratory analyses, while the exact variant should be preferred in accuracy-critical applications.

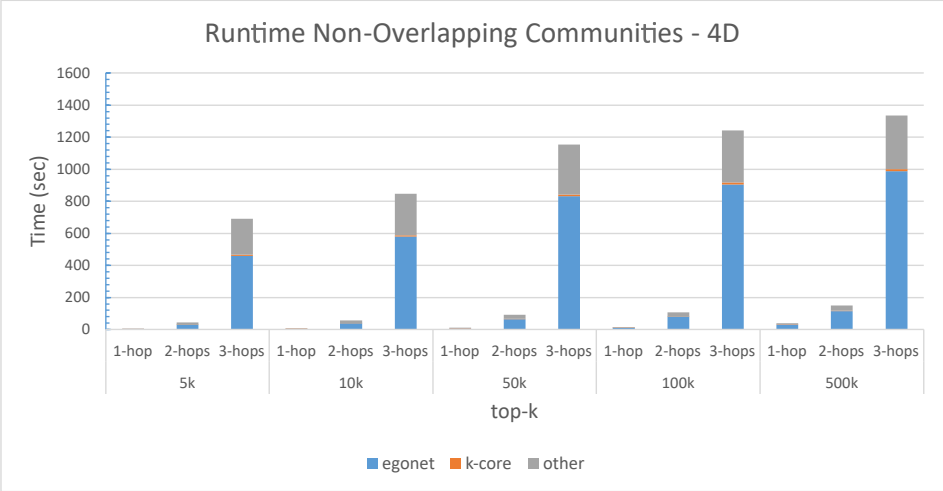


Fig. 16. Runtime breakdown of the non-overlapping community detection for 4D data. Time is split into egonet construction, k-core extraction, and other steps

4.3. Community Detection

In this section we evaluate the performance of our community detection algorithm for both non-overlapping and overlapping cases. We will examine how they perform varying the data dimensionality as well as the input parameters which are the number of the top-k nodes that are processed (5k, 10k, 50k, 100k, 500k) and the number of hops (1, 2, 3). We split the total computation time in three parts: the egonet time, core time and the rest of the process (denoted as other). In the following sections, for the domination score computation we have used the exact algorithm with a grid size of 100.

The non-overlapping case for the 4-d data is presented in Figures 16. We observe that most of the computation time is attributed to the egonet computation, i.e. to the creation of the subgraph starting from a certain node with distance h, while the core operation is the less time-consuming. There is a significant increase in runtime between the 2-hop and the 3-hop cases, while comparing the execution time between different data dimensionalities, we do not observe any significant difference. The rest of the computation marked as *other* is also significant because we need to maintain some additional information regarding the nodes that have been already processed.

The results for the overlapping are shown in Figure 17 for the 4-d case. We observe that the overlapping case is, in general, more time-consuming than the non-overlapping case. The difference between them is low when the top-k is small but, it becomes significantly large as the top-k number increases, especially when we are exploring 3-hop communities. We point out that overlapping communities provide the opportunity to include the same nodes in different communities. On the other hand, this process is more time-consuming because the number of potential communities is significantly larger.

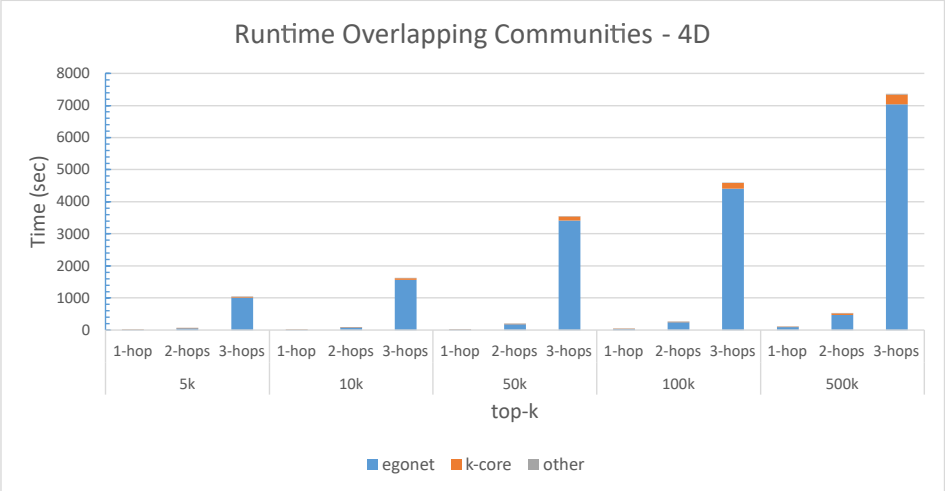


Fig. 17. Runtime breakdown of the overlapping community detection for 4D data. Time is split into egonet construction, k-core extraction, and other steps

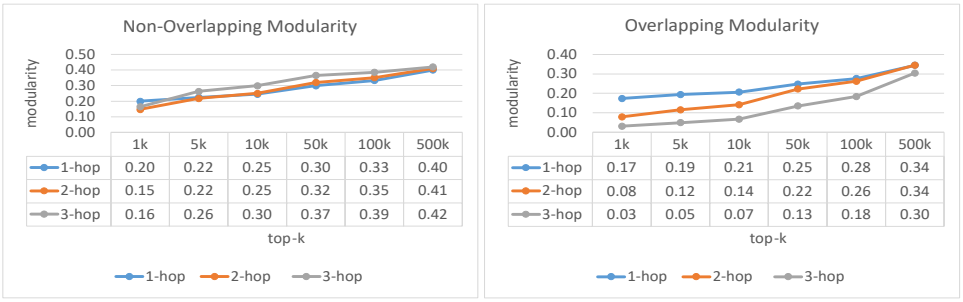
For example, in the overlapping case for the 4-dimensional data and top-50k nodes the algorithm needs 18.84 seconds for 1-hop, 195.27 seconds for 2-hops and 3546.57 seconds for 3-hops case. The non-overlapping execution time for the same parameters is 9.73 seconds for 1-hop, 91.77 seconds for 2-hops and 1155.05 seconds for 3-hops. As we observe, the overlapping case is 2 to 3 times more time-consuming than the the non-overlapping algorithm. The difference is increasing as the top-k increases and in the top-500k case the overlapping algorithm is 2.5 - 5.5 times slower.

The majority of the time here is also consumed by the egonet operation. In contrast, the *other* part in the overlapping case is very small since we don't need to monitor and maintain any list regarding which nodes have been visited.

4.4. Community Evaluation

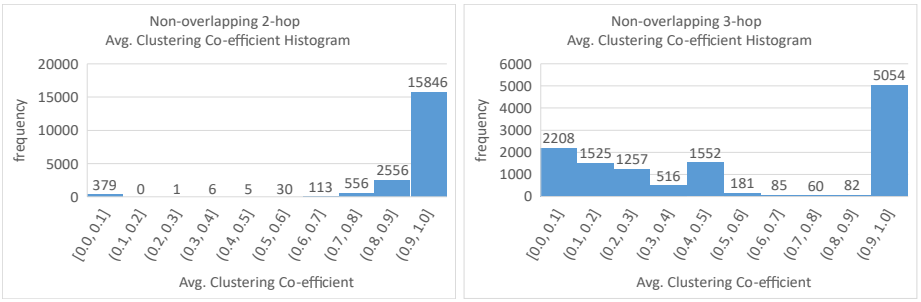
In this section, we evaluate the communities discovered by our algorithms. We also present some quality measures for both the non-overlapping and the overlapping case. In Figure 18, we observe how the average modularity of the returned communities progresses. For the non-overlapping case we observe almost the same values for 1-hop and 2-hop, while the 3-hop case has a slightly increased modularity. In the overlapping case we observe the opposite. The 1-hop case has better values than the 2-hop and 3-hop cases. In both non-overlapping and overlapping cases the highest modularity is observed when we compute the top-500k. The modularity value is above 0.3 which indicates a strong community structure, as studied in [16].

Next we explore the average clustering coefficient and the density of the discovered communities for different options for the non-overlapping and the overlapping case. In this experiment, the top-k parameter was set to 50k, whereas the detection process was executed for 1, 2 and 3 hops.



(a) non-overlapping (b) overlapping

Fig. 18. Average modularity of discovered communities varying parameters k and h for (a)non-overlapping case and (b) overlapping case



(a) 2-hop (b) 3-hop

Fig. 19. Average clustering coefficient for non-overlapping communities

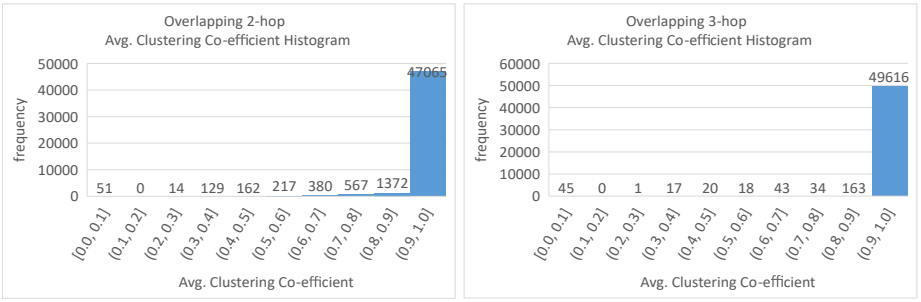
In Figure 19 we present the results of the average clustering coefficient for the non-overlapping case for the 2-hop and 3-hop cases. We observe that while most of the communities in all three cases are in the 0.9-1.0 bin, the 3-hop non-overlapping case also has a significant amount of communities fall in the lower bins (0.0 - 0.5).

In Figure 20 we present the results of the average clustering coefficient for the overlapping case for the 2-hop and 3-hop case. Here also the bin 0.9-1.0 contains the most communities and in contrast to the non-overlapping case, in the 3-hop case almost all 50k communities fall in the 0.9-1.0 bin.

In Figure 21, we present the community density histograms for the non-overlapping case for 2-hop and 3-hop. The pattern we observe here is analogous in some part to the average clustering coefficient. Most of the communities fall into the 0.9-1.0 bin. The overlapping case is similar to the non-overlapping one, as presented in Figure 22.

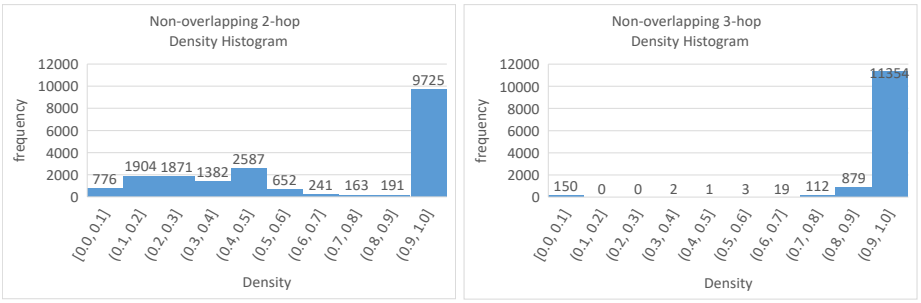
4.5. Precision of Communities with Approximate Domination Score

In this series of experiments, we evaluate the non-overlapping and the overlapping algorithms to get the top-10k communities applying domination scores that were



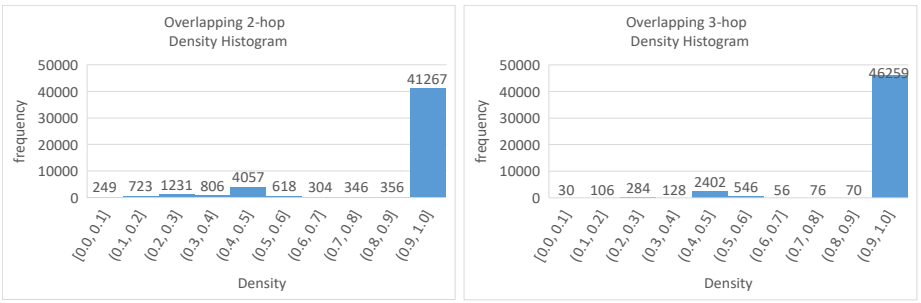
(a) 2-hop (b) 3-hop

Fig. 20. Average clustering coefficient for overlapping communities



(a) 2-hop (b) 3-hop

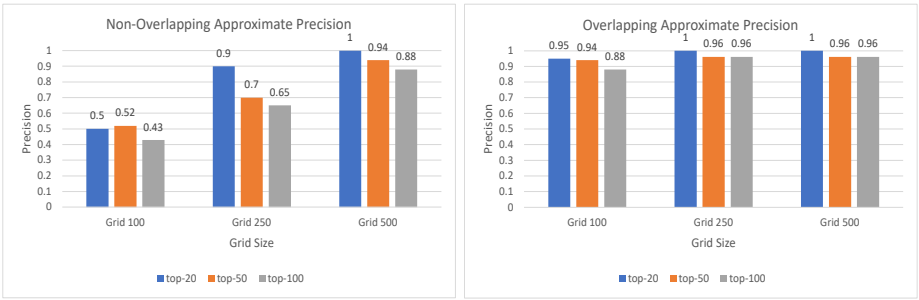
Fig. 21. Density histogram for non-overlapping communities



(a) 2-hop (b) 3-hop

Fig. 22. Density histogram for overlapping communities

computed using the approximate algorithm for various grid sizes. Then we measure the precision of the communities returned compared to the communities discovered when we apply the exact domination score computation.



(a) non-overlapping (b) overlapping

Fig. 23. Precision of approximate domination score

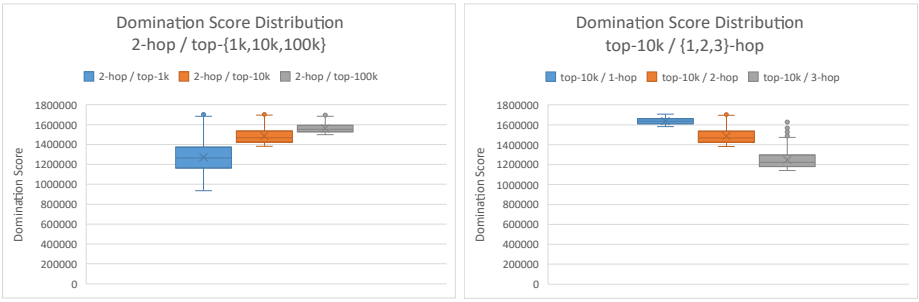
In Figure 23 we present the precision of the top 20, 50 and 100 communities for the non-overlapping and overlapping algorithms when we apply the approximate domination score for grid size 100, 250 and 500. The precision for the top-20 communities is higher for all grid sizes, with the highest precision achieved with the 100-grid. The precision for the top-50 and top-100 items is also higher for the smaller grid sizes, but the 250-grid and 500-grid still performs relatively well.

For the non-overlapping case, the precision is generally lower than the overlapping case. The precision for the top-20 communities is the highest for the 500-grid, followed by the 250-grid and then the 100-grid. For the top-50 and top-100 communities, the precision is lower for all grid sizes, with the highest precision achieved for the 500-grid.

4.6. Parameter Sensitivity

In this section we study and evaluate the impact of our method's key parameters. The number of top-k nodes to be processed (k) and the number of hops (h). To provide a robust comparison of their effects, for each parameter configuration (e.g., h=2, k=100k), we generate a set of candidate communities and then analyze the score distribution of the top-500 resulting communities, ranked by their final score based on our proposed metric ($norm(C)$). This approach allows us to evaluate which parameter settings are most effective at producing a high volume of elite, high-scoring communities, which is a key objective for practitioners seeking optimal results. The goal is to isolate the effect of each parameter on this elite subset, thereby offering guidance on tuning the method.

To achieve this, we conducted two sets of experiments. In the first, we fix the hop count (h=2) and vary k to measure how the initial search scope impacts the quality of the final top-500 communities. The hypothesis is that a broader initial search (larger k) will yield a superior elite subset. In the second, we fix k=10k and vary h to quantify how search granularity affects the quality of even the best communities found. This allows us to determine if broader, large-hop communities can compete in quality with more compact, small-hop ones when comparing their respective top-ranked results.



(a) fixed h=2, varying k (b) fixed k=10k, varying h

Fig. 24. Top-500 community score distribution. (a): fixed h=2, varying k. (b): fixed k=10k, varying h

Figure 24 presents the score distribution of the top-500 discovered communities, illustrating the impact of the search scope (k) and the community radius (h). The plot on the left (a) demonstrates that increasing the number of top-k seed nodes from 1k to 100k systematically improves the quality of the discovered communities, evidenced by a clear upward shift in the median and overall score distribution. This suggests that a broader search scope is more effective at identifying optimal communities. On the other hand, the plot on the right (b) reveals that increasing the hop count (h) from 1 to 3, causes a sharp decrease in the community scores, indicating that the highest-quality communities are compact and localized. Therefore, these findings strongly suggest that for maximizing the community score, the ideal strategy is to employ a large search scope (k) combined with a minimal community radius (h=1 or 2). Additionally, if we also consider the runtime as discussed in Section 4.3, we can see that the choice of h has a much more significant impact on performance than k. Higher selection of $h \geq 3$, is leading to exponentially longer runtimes. This further reinforces the recommendation to keep h low for practical applications.

To provide a clear and actionable summary of these findings, we present a detailed guide for parameter selection in Table 9 and Table 8. Table 9 outlines the role of the hop count (h), detailing how its value can be tuned to control the desired community granularity, from dense, high-quality local clusters (h=1) to broader, macro-level structures for exploratory analysis ($h \geq 3$), while considering the significant trade-offs in performance and average community score. Concurrently, Table 8 provides guidance on selecting the search scope (k), explaining how a focused search (small k) offers efficiency, whereas an exhaustive search (large k) maximizes the potential for discovering the optimal, highest-scoring communities. Together, these tables offer a practical framework for practitioners to configure our method based on their specific analytical goals and computational constraints.

Table 8. Guidance for Selecting the k Parameter (Search Scope)

Setting	Use-Case	Outcomes & Trade-offs
k = Small (Focused)	For fast, efficient community detection when the goal is to find good communities seeded only by the most elite nodes.	**Outcome:** A limited set of high-quality communities. **Trade-off:** May miss the single best community if it is not seeded by a top-ranked node.
k = Medium (Robust)	To achieve balance between search time and the quality of results, often representing the optimal setting.	**Outcome:** A comprehensive set of excellent communities. **Trade-off:** Our experiments show this range provides a great return on computational investment.
k = Large (Exhaustive)	Maximize the likelihood of finding the absolute best-scoring community, especially in graphs where many nodes have similar high scores.	**Outcome:** The highest potential for community quality. **Trade-off:** Increased runtime, best used when result optimality is important.

Table 9. Guidance for Selecting the h Parameter (Community Granularity)

Setting	Use-Case	Outcomes & Trade-offs
h = 1 (Local)	To find the highest-quality, dense core communities with maximum efficiency.	**Outcome:** Small, elite communities with high domination scores. **Trade-off:** Highly localized; will not discover broader structures.
h = 2 (Balanced)	To find well-structured communities that extend beyond immediate neighbors, offering a balance between size and quality.	**Outcome:** Medium-sized communities that are often a good compromise between local detail and broader context. **Trade-off:** Slower than h=1 but often manageable.
h ≥ 3 (Broad)	For exploratory analysis of large-scale, macro-level structures across the network.	**Outcome:** Very large communities. **Trade-off:** Significantly lower average quality and exponentially higher runtime.

5. Conclusions

In this work, we have studied the problem of dominant community detection in networks with node attributes. Our approach is composed of two main concepts: the power of nodes expressed by means of the domination score of the attributes and the community coherence which is computed by means of the maximum core. One of the novelties of our approach is the support of both non-overlapping and overlapping community detection. The whole process is parameterized by the number of hops. Performance evaluation results based on real-life attributed networks demonstrate the efficiency and effectiveness of the proposed approach. Efficiency is measured with respect to the overall runtime, whereas effectiveness is measured by means of the density and the clustering coefficient of the discovered communities. In addition to the exact computation of domination scores, approximate domination scores are also applied towards faster processing, without significant sacrifice of the accuracy of the results.

The proposed methodology can be applied to a variety of real-world analytical tasks. In academic citation and co-authorship networks, nodes correspond to authors annotated with attributes such as publication count, citation number or h-index. High domination score nodes highlight influential researchers, and the induced dominant communities reveal cohesive groups of high-impact authors or laboratories. The non-overlapping case can be applied to partition the network into disjoint fields, whereas the overlapping case is appropriate to capture cross-disciplinary researchers belonging to multiple groups. In this way, DomiCom can support tasks such as identifying elite research clusters, emerging teams, or bridging authors who connect different disciplines. In social networks, attributes like activity level, number of followers, and engagement metrics allow DomiCom to discover communities of influential users, supporting applications such as viral marketing, influence maximization, or misinformation monitoring. In biological interaction networks, nodes represent genes or proteins with attributes such as expression levels or disease relevance, and dominant communities correspond to functional modules that are both structurally coherent and jointly strong across biological attributes. These examples illustrate that DomiCom can be deployed in diverse settings, from compact elite clusters to broader macro-level structures, by tuning parameters such as hop count, overlapping vs. non-overlapping detection and exact or approximate score computation.

Some interesting future research directions are: 1) the use of random walks instead of hops, 2) the use of parallelism to speed-up computation, and 3) the application of node embedding techniques.

References

1. Barabási, A.L., Pósfai, M.: Network science. Cambridge University Press, Cambridge (2016), http://barabasi.com/networksciencebook/
2. Bi, F., Chang, L., Lin, X., Zhang, W.: An optimal and progressive approach to online search of top-k influential communities. Proc. VLDB Endow. 11(9), 1056–1068 (may 2018), https://doi.org/10.14778/3213880.3213881

3. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proceedings of the 17th IEEE International Conference on Data Engineering (ICDE). pp. 421–430 (2001), https://doi.org/10.1109/ICDE.2001.914855
4. Chen, S., Wei, R., Popova, D., Thomo, A.: Efficient computation of importance based communities in web-scale networks using a single machine. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. p. 1553–1562. CIKM '16, Association for Computing Machinery, New York, NY, USA (2016), https://doi.org/10.1145/2983323.2983836
5. Chen, Z., Jia, M., Yang, B., Li, X.: Detecting overlapping community in complex network based on node similarity. Comput. Sci. Inf. Syst. 12(2), 843–855 (2015)
6. Fang, Y., Cheng, R., Luo, S., Hu, J.: Effective community search for large attributed graphs. Proc. VLDB Endow. 9(12), 1233–1244 (aug 2016), https://doi.org/10.14778/2994509.2994538
7. Huang, X., Cheng, H., Qin, L., Tian, W., Yu, J.X.: Querying k-truss community in large and dynamic graphs. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. p. 1311–1322. SIGMOD '14, Association for Computing Machinery, New York, NY, USA (2014), https://doi.org/10.1145/2588555.2610495
8. Huang, X., Lakshmanan, L.V.S.: Attribute-driven community search. Proc. VLDB Endow. 10(9), 949–960 (may 2017), https://doi.org/10.14778/3099622.3099626
9. Huang, X., Lakshmanan, L.V.S., Yu, J.X., Cheng, H.: Approximate closest community search in networks. Proc. VLDB Endow. 9(4), 276–287 (dec 2015), https://doi.org/10.14778/2856318.2856323
10. Jiang, Y., Huang, X., Cheng, H.: I/o efficient k-truss community search in massive graphs. The VLDB Journal 30(5), 713–738 (apr 2021), https://doi.org/10.1007/s00778-020-00649-y
11. Li, R.H., Qin, L., Ye, F., Wang, G., Yu, J.X., Xiao, X., Xiao, N., Zheng, Z.: Finding skyline communities in multi-valued networks. The VLDB Journal 29(6), 1407–1432 (jun 2020), https://doi.org/10.1007/s00778-020-00618-5
12. Li, R.H., Qin, L., Ye, F., Yu, J.X., Xiao, X., Xiao, N., Zheng, Z.: Skyline community search in multi-valued networks. In: Proceedings of the 2018 International Conference on Management of Data. p. 457–472. SIGMOD '18, Association for Computing Machinery, New York, NY, USA (2018), https://doi.org/10.1145/3183713.3183736
13. Li, R.H., Qin, L., Yu, J.X., Mao, R.: Influential community search in large networks. Proc. VLDB Endow. 8(5), 509–520 (jan 2015), https://doi.org/10.14778/2735479.2735484
14. Li, R.H., Qin, L., Yu, J.X., Mao, R.: Finding influential communities in massive networks. The VLDB Journal 26(6), 751–776 (dec 2017), https://doi.org/10.1007/s00778-017-0467-4
15. Malliaros, F.D., Giatsidis, C., Papadopoulos, A.N., Vazirgiannis, M.: The core decomposition of networks: theory, algorithms and applications. The VLDB Journal 29(1), 61–92 (nov 2019)
16. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Physical review E 69(2), 026113 (2004)
17. Papadopoulos, A., Tiakas, E., Tzouramanis, T., Georgiadis, N., Manolopoulos, Y.: Skylines and Other Dominance-Based Queries. Synthesis Lectures on Data Management, Morgan-Claypool (2020)
18. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (03 2014)

19. Stallings, J., Vance, E., Yang, J., Vannier, M.W., Liang, J., Pang, L., Dai, L., Ye, I., Wang, G.: Determining scientific impact using a collaboration index. Proceedings of the National Academy of Sciences 110(24), 9680–9685 (2013), https://www.pnas.org/doi/abs/10.1073/pnas.1220184110
20. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: Extraction and mining of academic social networks. In: KDD'08. pp. 990–998 (2008)
21. Wang, M., Yang, S., Wu, L.: Improved community mining method based on LFM and EAGLE. Comput. Sci. Inf. Syst. 13(2), 515–530 (2016)
22. Xie, X., Song, M., Liu, C., Zhang, J., Li, J.: Effective influential community search on attributed graph. Neurocomputing 444, 111–125 (2021), https://www.sciencedirect.com/science/article/pii/S0925231221001223
23. Yiu, M.L., Mamoulis, N.: Efficient processing of top-k dominating queries on multi-dimensional data. In: Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB). pp. 483–494 (2007), http://www.vldb.org/conf/2007/papers/research/p483-yiu.pdf
24. Yu, D., Zhang, L., Luo, Q., Cheng, X., Yu, J., Cai, Z.: Fast skyline community search in multi-valued networks. Big Data Mining and Analytics 3(3), 171–180 (2020)

Nikolaos Georgiadis is a Ph.D. candidate at the School of Informatics of Aristotle University of Thessaloniki (AUTH). He received a B.Sc. in Software Engineering from the Technical University of Kavala and a M.Sc. in Information Systems from the School of Informatics of AUTH. His main research interests focus on efficient algorithmic techniques for Knowledge Discovery from Big Data, Data mining, Social Networks, Data Structures, Similarity Search.

Eleftherios Tiakas, received a B.Sc. in Mathematics (1994) from the Department of Mathematics, a B.Sc. in Computer Science - Informatics (2006), an M.Sc. in Computer Science - Informatics with focus on Information Systems (2006), and a Ph.D. in Informatics (2011) from the Department of Informatics, Faculty of Sciences, Aristotle University of Thessaloniki (AUTH). During the period 2011-2022 he served as Adjunct Lecturer, a postdoctoral researcher, and in the Laboratory Teaching Staff of the Department of Informatics, AUTH. Since 2022, he is an Assistant Professor in the Department of Accounting and Information Systems, International Hellenic University (IHU). His research interests include Databases, Data Structures, Data Mining, Information Retrieval, Similarity Search, Social Networks. He is the author/co-author of 16 articles published in international journals, 26 papers published in international conference proceedings, 1 book, 2 book chapters, and 2 technical reports. His publications have received over 850 citations. He participated in 16 research projects and programs funded by the European Union and Greece. He also serves as a PC-Member and reviewer for several international journals and conferences.

Apostolos N. Papadopoulos received his 5-year Diploma Degree from the Computer Engineering and Informatics Department of the University of Patras and his PhD Degree in Computer Science from the Department of Informatics of Aristotle University of Thessaloniki, Greece. Since May 2023, he is a Full Professor with the Department of Informatics of Aristotle University. His research interested include:

efficient query processing techniques in database systems, algorithms for mining graphs, parallel and distributed algorithms for big data, and data management and mining data streams. According to Google Scholar he has 4567 citations and his h-index is 35, whereas according to American Mathematical Society his Erdős number is 3. https://scholar.google.com/citations?user=b8ptH6oAAAAJ&hl=en

Received: April 30, 2024; Accepted: October 30, 2025.

Computer Science and Information Systems 23(1):113–132 https://doi.org/10.2298/CSIS250612004W

ASD-RRT*: An Enhanced Path Planning Algorithm Based on RRT* for Multi-Obstacle Environments

Chao Wang and Wenbin Li

Computer School, Beijing Information Science and Technology University
Beijing 100192, China
wangchao@bistu.edu.cn
emaillwb@163.com

Abstract. The efficiency of sampling-based motion planning brings wide application in autonomous vehicles. The conventional rapidly exploring random tree (RRT) algorithm and its variants have gained significant successes, but there are still challenges for the efficient motion planning in complex and multi-obstacles environments. Conventional sampling methods perform unconstrained sampling across the entire search space, often resulting in suboptimal paths. In this paper, we propose a novel algorithm, Adaptive Sampling and Densification RRT* (ASD-RRT*), for path planning in multi-obstacle environments. Our method extends RRT*-based sampling methods by incorporating adaptive sampling to enhance performance in complex environments. The adaptive sampling approach allows the algorithm to focus on effective regions, reducing sampling of irrelevant points and finding feasible solutions with fewer samples while maintaining the asymptotic optimality of RRT*. Further, we introduce a new optimization method for high-curvature paths and a routing strategy that satisfies vehicle dynamics constraints, aiming to improve path quality. The effectiveness and efficiency of the proposed ASD-RRT* are proved through comparative experiments in different environments. Experimental results demonstrates our method offers a reduction of 55.7% in planning times and 18.7% in path lengths over RRT* in a variety of environments.

Keywords: Motion planning, path planning, RRT*, adaptive sampling, dynamic smoothing.

1. Introduction

Autonomous driving technologies hold promise for reducing road traffic accidents, enhancing traffic efficiency and providing convenient travel solutions [1]. Typical on-board modules in autonomous vehicles include perception, prediction, localization, planning and control [2]. Among these, trajectory planning is crucial as it aims to devise a safe and comfortable path for an autonomous vehicle from its starting position to its destination, considering the environment and constraints. A safe path avoids collisions, while a comfortable path minimizes sudden changes in speed or lane [3]. Numerous researchers have contributed to this field, proposing methods such as curve planning, artificial potential field method, sampling method, model predictive control algorithm and data-driven planning algorithm [4, 5].

Curve planning is widely employed for vehicle lane changing, turning and local trajectory smoothing [6]. While it produces smooth trajectories, it often struggles to explore all feasible paths and may become trapped in local optima, making it less effective

in complex environments [7]. The artificial potential field method simulates interactions between objects to plan paths, offering simplicity, intuitiveness, and computational efficiency, making it suitable for real-time obstacle avoidance and path smoothing [8]. However, improperly constructed potential fields can lead to poor path smoothness, coherence, or entrapment in local optima [9]. Sampling-based methods [10, 11] excel in real-time performance and can accommodate various constraints during planning, yet their paths often exhibit zigzag patterns and lack smoothness. Model predictive control(MPC) [12] integrates environmental and vehicle dynamics constraints effectively through well-designed objective functions and constraints, but it relies heavily on on-board sensor data and incurs significant computational overhead. Deep learning-based end-to-end motion planning algorithms [13, 14] map sensor inputs directly to control outputs, mimicking human driving behaviors. Despite this, they struggle with generalizing to extreme scenarios and lack interpretability. Reinforcement learning-based motion planning algorithms [15] learn optimal paths through trial and error interactions with the environment, but they are highly dependent on the design of the reward function and need to train the model in advance [16].

Considering the strengths and weaknesses of these algorithms, sampling-based methods are notable for their computational efficiency, real-time performance, and adaptability to various driving scenarios. However, adjustments to their sampling strategies are necessary to satisfy different constraint conditions. This paper proposes an efficient adaptive sampling ASD-RRT* motion planning algorithm for autonomous global path planning. Initially, the sampling region is determined based on a simplified raster map, and then optimized to obtain a safe, collision-free drivable path. Compared to traditional sampling and optimization-based planning algorithms, the proposed method significantly enhances solution speed by optimizing the sampling region. Relative to the RRT* algorithm, the average computation time reduced by 57.63% and the average path length reduced by 18.7%. This algorithm integrates the RRT* algorithm with control point curve optimization. An initial guidance path is obtained using a graph search algorithm, which subsequently defines the sampling region for the RRT* algorithm. This approach not only ensures the completeness of the RRT* algorithm but also mitigates performance losses due to non-differentiable search spaces.

2. Related Work

Combining sampling-based and optimization-based methods is the preferred strategy for reducing solution time and ensuring high-quality paths in autonomous vehicle path planning. Sampling-based planning algorithms, such as probabilistic road map (PRM) [17] and rapidly-exploring random tree (RRT) [18], select the best path by randomly sampling candidate paths in the environment and evaluating their safety. PRM and its various improvements (Lazy PRM [19], Anytime PRM [20], PRM* [21], Fast Marching PRM [22], etc.) require environmental information to construct the probabilistic roadmap. Consequently, when the surroundings are unknown, finding a feasible solution may be impossible. Additionally, limited sampling density and high connection costs between sampled points reduce planning efficiency, particularly in dense obstacle areas or narrow passages, making it unsuitable for real-time applications in autonomous vehicle scenarios [23].

RRT algorithms are highly efficient and can plan collision-free feasible paths without requiring precise global map information. However, the paths generated by RRT are often tortuous and lack directional guidance in the search. To address these issues, researchers have proposed numerous RRT algorithm improvements for vehicle motion planning. To improve path quality, Karaman et al. [21] proposed the asymptotically optimal RRT* algorithm, which converges to the optimal solution given sufficient time by searching for low-cost parent nodes and adopting pruning optimization strategies. However, the process of finding new parent nodes and rewiring reduces the algorithm's efficiency. Webb et al. [24] addressed the issue of RRT* not meeting autonomous vehicle dynamic constraints by proposing the Kinodynamic RRT* algorithm. This method optimizes the trajectory planned by the RRT* algorithm using controllable linear dynamic systems, resulting in a trajectory that satisfies the dynamic constraints of autonomous vehicles. Nevertheless, there is still room for improvement in convergence speed and optimal path quality. Armstrong et al. [25] proposed AM-RRT*, which combines auxiliary metrics with Euclidean metrics to improve path quality in obstacle-rich environments while maintaining the expected performance of RRT variants. However, the planning performance is not satisfactory, particularly in terms of search time. Sun et al.[26] proposed NAMR-RRT, a neural adaptive motion planning algorithm for robots in dynamic environments. It uses neural network-generated heuristic regions to dynamically guide exploration, improving planning efficiency and reducing trajectory length.

To accelerate the algorithm's convergence speed and reduce search time, bidirectional search strategies are widely employed. Klemm et al. [27] proposed the RRT*-Connect algorithm, which builds search trees with the start and goal points as root nodes. When overlapping sub-nodes are discovered during the search, bidirectional backtracking is initiated, repeatedly computing paths until planning is completed, thereby improving search efficiency. Hussain [28] proposed the IB-RRT* algorithm, which uses a bidirectional tree structure and intelligent sample insertion heuristic algorithm to quickly converge to the optimal solution, enhancing the algorithm's efficiency and quality. Ma et al. [29] proposed the Bi-Risk-RRT, which constructs paths based on bidirectional search sampling strategies and predicts collision risks with dynamic and static obstacles, effectively improving the motion planning efficiency of autonomous vehicles in dynamic environments. However, the algorithm relies on predicting the motion of dynamic obstacles, and if the prediction model is inaccurate, the planned path may be infeasible in practice. Wang et al. [30] proposed the B2U-RRT, which also adopts a bidirectional search strategy and switches to single search mode when encountering the opposite tree to avoid boundary value issues during connection. Similarly, Wang et al. [31] proposed BI-AM-RRT, which optimizes motion planning performance in dynamic environments by introducing auxiliary metrics with larger connection distances. It also employs bidirectional search sampling strategies to reduce search time and proposes a reconnection method based on roots and goals to shorten path length. Yao et al.[32] paper proposes NG-RRT*, a neural-guided path planning algorithm for flexible needle puncture, using 3D U-Net for entry point selection and neural-guided sampling to enhance efficiency. It also introduces an arc trajectory search for smooth paths. Simulations demonstrate improved performance and clinical feasibility.

The positional relationship between the starting point and the goal point affects the sampling region's restriction, which can be optimized by limiting the tree structure's sampling area. For example, Informed-RRT* [33] and Informed RRT*-Connect [34] guide

sampling and connection using restricted areas, utilizing heuristic information to accelerate the path search process. N. Li et al.[35] proposed the ABi-RRT algorithm, which employs an adaptive sampling strategy and adjustable step size to enhance search efficiency and integrates a greedy path pruning algorithm and cubic B-spline curves to optimize path smoothness and length in complex 3D environments, achieving a 100% success rate and significantly reduced running time. Y. Su et al.[36] proposed a dynamic path planning method combining improved RRT* and DWA, using APF for initial path generation, adaptive step-size for optimization, and enhanced DWA for obstacle avoidance in dynamic environments. Simulations and experiments show it generates high-quality paths and effectively navigates unknown obstacles. D. Uwacu et al.[37] proposed HAS-RRT, a hierarchical RRT-based motion planning strategy guided by a workspace skeleton, achieving up to 91% runtime reduction and a 30% smaller tree than competitors while maintaining competitive path costs. Inspired by Informed-RRT*, a novel search sampling method based on a adaptive sampling area, i.e., ASD-RRT*, is proposed in this article to further reduces the search time and the path length.

3. Proposed ASD-RRT*

The ASD-RRT* algorithm aims to improve the inefficiencies and tortuous paths characteristic of the classical RRT* algorithm. By introducing adaptive sampling regions and an obstacle recurrence superposition strategy, the algorithm achieves iterative global path planning, effectively reducing the performance loss associated with the undifferentiated treatment of the search space in RRT*. Additional, this algorithm employs a smoothing method that satisfies vehicle dynamic constraints to enhance path smoothness, thereby further improving the quality of the generated paths.

3.1. Algorithm Framework

Algorithm 1 describes the details of ASD-RRT*. The search tree in RRT is defined as $\mathcal{T} = \{n_i \mid i = 1, 2, \cdots, N\}$, where $n_i = [x_i, y_i, t_i, c_i, dx_i]$ represents the information of the ith node, containing node coordinates $[x_i, y_i]$, current time t_i, current node cost c_i, and the parent node number dx_i. P_{start} and P_{goal} represent the planning start and target points, respectively. δ is the adaptive sampling factor, *thr* is the threshold value used when simplifying the map with obstacles. map_s is the simplified raster map, S is the sampling area, $Path_{init}$ and $Path_s$ are the initial bootstrap path and final output path, respectively. Finally, a dynamic smoothing method with dynamics constraints is employed to enhance the quality of the path solution.

Next, the procedures on which the algorithm depends are described. The *InitializeTree*(P_{start}) function initializes the search tree $\mathcal{T}$ with P_{start} as the root node. *SimplifyMap*(*thr*) simplifies the obstacles based on the *thr* and returns the obstacle information *Obs* and the simplified map map_s. *InitialGuidingPath*(map_s, P_{start}, P_{goal}) generates an initial guiding path using graph search concepts. *SamplingArea*($Path_{init}$, δ) creates an adaptive sampling region based on $Path_{init}$ and δ. *ObstacleRecurrencePlan*($\mathcal{T}$, map_s, S, ξ) iteratively overlays obstacles based on the recurrence scale factor ξ and plans the path. *HighCurvatureProcessing*($Path_s$) returns the final $Path_s$ after smoothing.

Algorithm 1 ASD-RRT*

Input: $P_{start}, P_{goal}, \delta, thr, map, \xi$
Output: $Path_s$

$\mathcal{T} \leftarrow InitializeTree(P_{start})$
$Obs, map_s \leftarrow SimplifyMap(thr)$
$Path_{init} \leftarrow InitialGuidingPath(map_s, P_{start}, P_{goal})$
$S \leftarrow AdaptiveSamplingRegion(Path_{init}, \delta)$
$Path_s \leftarrow ObstacleRecurrencePlan(\mathcal{T}, Obs, S, \xi)$
$Path_s \leftarrow HighCurvatureProcessing(Path_s)$
return $Path_s$

3.2. Adaptive Sampling Area

The Rapidly-exploring Random Tree (RRT) algorithm explores the entire map space through continuous sampling and expansion. When the RRT algorithm expands a new vertex, it calculates the distance between that point and the target. If the distance is less than a predetermined threshold, the algorithm has found the target point and stops the search. Otherwise, the search continues. However, this uniform sampling across the entire state space struggles to effectively cover critical regions, leading to repeated examination of certain areas, resulting in wasted computational resources and reduced convergence speed.

Figure 1 illustrates the relationship between map size and sampling efficiency. The RRT and RRT* algorithms perform uniform sampling across the entire map space, causing the time taken by the algorithms to grow almost linearly with increasing map size. After finding an initial path, Informed-RRT* restricts the sampling area to an ellipse, defined by the heuristic space of the currently known path. While the time taken by Informed-RRT* also increases with map size, it is significantly less than that of the RRT and RRT* algorithms.

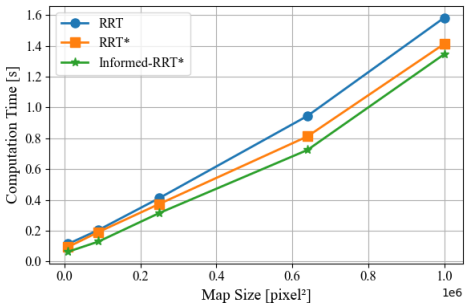


Fig. 1. Relationship between map size and RRT sampling efficiency

To address this issue, this paper proposes an adaptive sampling strategy that introduces initial feasible solutions and sampling regions to improve algorithm efficiency. To minimize the impact of obstacles on sampling efficiency, we simplify the global raster map

and introduce an initial guided path based on this simplified map, thereby determining the sampling region for the RRT* algorithm.

The simplification of the raster map involves several steps [38]. First, the occupancy information in the raster map is classified. Then, the obstacle information in the connected raster cells is segmented to ensure that each obstacle region can be individually identified. Finally, obstacles are sorted based on the size of the occupancy data, and obstacles smaller than a predefined simplification threshold are removed. As shown in Figure 2(a), the numbers represent the proportion of the obstacle area to the total map area. Figure 2(b) shows the map after simplifying obstacles that occupy less than 5% of the area.

The process of generating the initial guided path is presented in Algorithm 2. The priority queue Q_f_{open} stores nodes to be expanded, while the closed set S_f_{closed} keeps track of nodes that have already been expanded. The start node T_{start}, based on P_{start}, is initialized with its cost and other relevant information before being inserted into Q_f_{open} to initiate the path search process (Lines 1-3).

Algorithm 2 InitialGuidingPath

Input: map_s, P_{start}, P_{goal}
Output: $Path_{init}$
1: $Q_f_{open} \leftarrow \emptyset$; $S_f_{closed} \leftarrow \emptyset$
2: $T_{start} \leftarrow CreateNode(P_{start})$
3: *InitializePriorityQueue*(Q_f_{open}, T_{start})
4: **while** not *IsEmpty*(Q_f_{open}) **do**
5: $x_{current} \leftarrow Dequeue(Q_f_{open})$
6: **if** $x_{current}$ *IsGoal*(P_{goal}) **then**
7: $Path_{init} \leftarrow ReconstructPath(x_{current})$
8: **return** $Path_{init}$
9: **end if**
10: *RemoveFromPriorityQueue*($Q_f_{open}, x_{current}$)
11: *InsertIntoSet*($S_f_{closed}, x_{current}$)
12: **for** *neigh* **in** *GenerateNeighbor*($x_{current}, map_s$) **do**
13: **if** *neigh IsIn*(S_f_{closed}) **then**
14: **continue**
15: **end if**
16: $cost \leftarrow CalculateCost(x_{current}, neigh)$
17: **if** *neigh IsIn*(Q_f_{open}) **and** $cost \geq neigh.cost$ **then**
18: **continue**
19: **end if**
20: $neigh.cost \leftarrow cost$
21: $neigh.parent \leftarrow x_{current}$
22: *Enqueue*($Q_f_{open}, neigh$)
23: **end for**
24: **end while**
25: **return** $\emptyset$

During the loop, while Q_f_{open} is not empty, the node with the lowest cost is dequeued as the current node $x_{current}$. If $x_{current}$ matches the goal point P_{goal}, the goal is considered reached, and the path is constructed by backtracking through the parent information

recorded in each node (Lines 5-9). $x_{current}$ is added to S_f_{closed} to mark it as processed and removed from Q_f_{open} to avoid reconsideration (Lines 10-11). For each neighbor of $x_{current}$, if it is already in S_f_{closed}, it is skipped. Otherwise, the cost from the start point to this neighbor is calculated (Lines 12-16). If the neighbor is already in Q_f_{open} and its cost is not lower than the known cost, it is not updated to avoid path degradation (Lines 17-19). Otherwise, its cost and parent information are updated, and it is enqueued into Q_f_{open} (Lines 20-22).

The initial guided path provides a foundation for path planning in real road environments. To ensure that the planned path remains within the lanes, an appropriate expansion

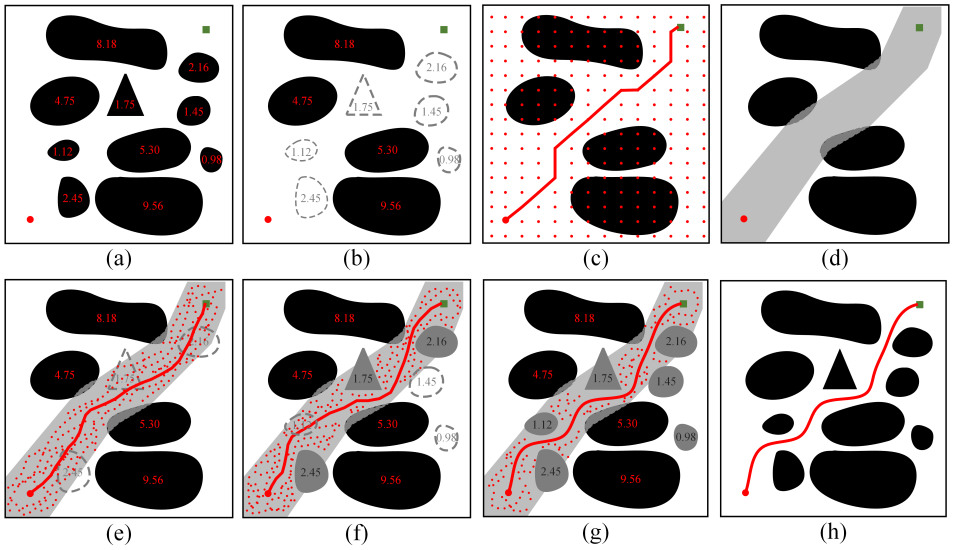


Fig. 2. Algorithm workflow

distance is selected to determine the sampling region. In a structured road environment, the expansion distance can be adjusted based on lane information, ensuring it falls within the drivable area for the autonomous vehicle. The formula for the expansion distance is as follows:

$$S = \frac{E}{\delta} \tag{1}$$

where S is the path expansion distance, E is the edge length in the environment map, and δ is the initial sampling area expansion factor.

Algorithm 3 provides the adaptive sampling region process. Initially, the input path is split into segments, and the initial path is assigned to S (Line 1). Normal vectors and translated segment lists for outward and inward translations are initialized (Line 2). The evaluateRegion(S) function is used to assess the generated sampling region to ensure that a feasible solution can be obtained within the region. The while loop continues until the sampling region is adequately evaluated (Line 3). Within the loop, each line segment

Algorithm 3 AdaptiveSamplingRegion

Input: $\text{Path}_{\text{init}}$, δ
Output: S

1: $Line_{seg} \leftarrow SplitSeg(Path_{init})$, $S \leftarrow Path_{init}$;
2: $NroVectors \leftarrow \emptyset$, $Trans_{out} \leftarrow \emptyset$, $Trans_{in} \leftarrow \emptyset$;
3: **while** not $evaluateRegion(S)$ **do**
4: **for** $i = 0$ to $Strlen(Line_{seg}) - 1$ **do**
5: $p_1 \leftarrow Line_{seg}[i], p_2 \leftarrow Line_{seg}[i+1]$;
6: $L \leftarrow \sqrt{(p_1.x - p_2.x)^2 + (p_1.y - p_2.y)^2}$;
7: $n_x \leftarrow -\frac{(p_1.y - p_2.y)}{L}, n_y \leftarrow \frac{(p_1.x - p_2.x)}{L}$;
8: $NroVectors.append((n_x, n_y))$;
9: $x_{out1} \leftarrow p_1.x + \delta \cdot n_x, y_{out1} \leftarrow p_1.y + \delta \cdot n_y$;
10: $x_{out2} \leftarrow p_2.x + \delta \cdot n_x, y_{out2} \leftarrow p_2.y + \delta \cdot n_y$;
11: $Trans_{out}.append((x_{out1}, y_{out1}), (x_{out2}, y_{out2}))$;
12: $x_{in1} \leftarrow p_1.x - \delta \cdot n_x, y_{in1} \leftarrow p_1.y - \delta \cdot n_y$;
13: $x_{in2} \leftarrow p_2.x - \delta \cdot n_x, y_{in2} \leftarrow p_2.y - \delta \cdot n_y$;
14: $Trans_{in}.append((x_{in1}, y_{in1}), (x_{in2}, y_{in2}))$;
15: **end for**
16: **end while**
17: $Trans_{out}.pop()$, $Trans_{\text{in}}.\text{pop}()$;
18: $S \leftarrow Trans_{out} + reverse(Trans_{in})$;
19: **return** S

is processed to calculate the normal vectors and translated segments. For each segment, the current point is dequeued, and the segment endpoints are identified (Lines 4-6). The segment length and normal vector components are computed (Lines 7-8), and the outward and inward translated points are calculated and appended to their respective lists (Lines 9-16). The end of the loop is reached after processing all segments (Line 17). Once the loop condition is satisfied, the last points from the translated segments are removed to avoid duplication (Line 17), and the final sampling region is constructed by combining the outward segments and the reversed inward segments (Line 18). The function returns the constructed sampling region (Line 19).

Figure 2 (a)-(d) illustrates the generation process of the adaptive sampling region. Figure 2 (a) shows the original map, Figure 2 (b) the simplified map, Figure 2 (c) the initial path from the red starting point to the green endpoint, and Figure 2 (d) the sampling region expanded by the sampling factor. The distance between the initial path and the gray boundary is S. To ensure that the width of the sampling region is sufficiently large and to maintain search efficiency, δ is generally set to 4, though this value can be adjusted to modify the search range and time.

3.3. Obstacle Recurrence Stacking Strategy

When dealing with environments with multiple obstacles, the RRT algorithm often generates excessively long paths due to obstacle avoidance. Additionally, the local minimum problem in such environments poses a significant challenge. The adaptive sampling region obtained from the simplified map greatly reduces the sampling space, enabling the

RRT* algorithm to quickly generate a planned path. However, there is a probability that the planned path within the sampling region may intersect with simplified obstacles.

After completing the path planning within the sampling region using RRT*, this paper introduces an obstacle superposition and reproduction strategy. This strategy involves gradually superimposing and reproducing obstacles on the simplified map. An obstacle superposition and reproduction scale factor, ξ, is introduced to control the batches of obstacle superposition and reproduction, ensuring the algorithm adapts to obstacles of different sizes. The maximum ξ value equals the number of simplified obstacles, representing the scenario where obstacles are reproduced one by one. Typically, ξ is set to 2 or 3, dividing the obstacles into two or three parts based on their area for superposition and reproduction, thereby enhancing the algorithm's ability to handle different types of obstacles.

Algorithm 3 summarizes the obstacle superposition and reproduction process. Initially, obstacles are sorted by area, and the map_s is assigned to the current map $Curr_{map}$ (Lines 1-3). The $obsRecurrence(Curr_{map}, obs)$ function is responsible for gradually overlaying and reproducing obstacles onto the simplified map, with path re-planning occurring after each overlay. This process continues until all obstacles are reproduced on the map. In each iteration, the current batch of obstacles is obtained based on ξ (Line 5), added to the current map (Lines 6-8), and the path is re-planned using the updated map (Line 9). The batch counter (batch) is updated after each batch (Line 10) to track the reproduced obstacle batches. Once all obstacles are reproduced, the function returns the final planned path (Line 12).

Algorithm 4 ObstacleRecurrencePlanning

Input: T, map_s, S, ξ
Output: $Path_s$
1: $Path_s \leftarrow \emptyset$; $batch \leftarrow 0$
2: $Sorted_obs \leftarrow$ *SortObstacles(map)*
3: $Curr_map \leftarrow map_s$
4: **while** $batch < \xi$ **do**
5: $Obs_batch \leftarrow$ *getBatch(*$Sorted_obs, batch$*)*
6: **for** obs in Obs_batch **do**
7: $Curr_map \leftarrow$ *obsRecurrence*$(Curr_map, obs)$
8: **end for**
9: $Path_s \leftarrow$ *ReplanPath*$(T, Curr_map, S)$
10: $batch \leftarrow batch + 1$
11: **end while**
12: **return** $Path_s$

The process of successive obstacle superposition and reproduction is illustrated in Figure 2 (e)-(h), with ξ set to 2. Figure 2 (e) shows the initial planning result of RRT* within the sampling region, Figure 2 (f) shows the superposition and reproduction result of obstacles with an area ratio between 1.5% and 5%, Figure 2 (g) shows the result for obstacles with an area ratio below 1.5%, and Figure 2 (h) shows the final planning result of the algorithm. Additionally, when adding new nodes based on the ASD-RRT* algorithm with adaptive sampling, a new node selection method is introduced. The new random

nodes must satisfy Equation (2) to be added to the random search tree. This node selection method ensures that the resulting path has better connectivity, improving search efficiency.

$$C_{new} \leq C_{path} - C_{goal} \tag{2}$$

where C_{new} is the current cost of the new node P_{new}, C_{path} is the total cost of the current optimal path, and C_{goal} is the Euclidean distance from the new node to the destination.

3.4. Path Optimization Strategy

Due to the inherent randomness of the RRT* algorithm, the generated paths often exhibit excessive curvature and lack smoothness, leading to instability in the motion of autonomous vehicles and resulting in mechanical wear and tear. To address these issues, this paper introduces a high-curvature dense sampling point method and a dynamic smoothing control point routing strategy that satisfy vehicle dynamics constraints. These methods are used for post-processing the generated paths to create continuous and smooth paths, thereby improving their quality.

A key parameter is the vehicle's maximum lateral acceleration $a_{\text{lat,max}}$, which is typically determined by the vehicle's design and safety considerations. At a specific speed, the vehicle's maximum curvature can be calculated using the following equation:

$$a_{\text{lat,max}} = v^2 k_{\text{max}} \tag{3}$$

where v is the vehicle speed and k_{max} is the maximum curvature. To satisfy the vehicle dynamics constraints and ensure driving safety, the maximum curvature of the trajectory planning path must not exceed the vehicle's maximum curvature, as given by the following equation:

$$k \leq k_{\text{max}} \tag{4}$$

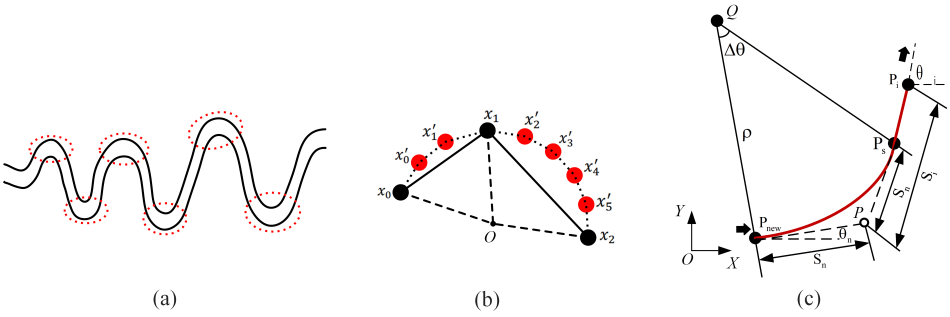


Fig. 3. Path optimization strategy (a) High curvature road; (b) Increasing the density of sampling points; (c) Rewiring strategy for control nodes

Algorithm 5 HighCurvatureProcessing

Input: $Path_s$
Output: $Path_s$

$a_{lat,max}, v \leftarrow VehicleDynamicsMode()$
$k_{max} \leftarrow \frac{v^2}{a_{lat,max}}$
for $i = 0$ **to** $(strlen(Path_s) - 3)$ **do**
  $P_1 \leftarrow Path_s[i]$, $P_2 \leftarrow Path_s[i+1]$, $P_3 \leftarrow Path_s[i+2]$
  $f_1' \leftarrow SubtractVectors(P_1, P_2)$
  $f_2'' \leftarrow SubtractVectors(P_2, P_3)$
  $k \leftarrow \frac{\|f_2''\|}{(1+\|f_1'\|^2)^{3/2}}$
  if $k_{max} \leq k$ **then**
    $L \leftarrow Distance(P_1, P_2, P_3)$
    $\Delta s \leftarrow \frac{1}{k_{max}}$
    $N \leftarrow \lceil \frac{L}{\Delta s} \rceil$
    $\Delta t_{in} \leftarrow \frac{1}{N}$
    for $j = 1$ **to** N **do**
      $t \leftarrow j \times \Delta t_{in}$
      $P_{den} \leftarrow CubicInterpolate(P_1, P_2, P_3, t)$
      $Path_s \leftarrow InsertPointPath(Path_s, P_{den}, i+1)$
    end for
  end if
end for
return $Path_s$

As shown in Figure 3 (a) and (b), when the path curvature is excessively high, the method of increasing the density of sampling points is employed to smooth the path. The curvature calculation of the discrete points satisfies the following equation:

$$k = \frac{|x'y'' - x''y'|}{(x'^2 + y'^2)^{3/2}} \tag{5}$$

where x' and y' are the first derivatives of the sampling point coordinates x and y, respectively, and x'' and y'' are the second derivatives of the sampling point coordinates. In regions of high curvature, the spacing Δs between densely sampled points needs to be sufficiently small to ensure path smoothness, satisfying the following equation:

$$\Delta s \leq \frac{1}{k_{\text{thread}}} \tag{6}$$

The number of densely sampled points N can be calculated using the following formula:

$$N = \lceil \frac{L}{\Delta s} \rceil \tag{7}$$

where L is the length of the path to be densely sampled.

The densification process for high-curvature segments is presented in Algorithm 5. First, the vehicle dynamics model determines the current speed and maximum lateral acceleration of the vehicle, which are used to calculate the maximum allowable curvature

(Lines 1-2). The algorithm then iterates through the path points, estimating the curvature for each segment formed by three consecutive points (Lines 3-7). If the curvature exceeds the maximum allowable value, the necessary spacing Δs, the number of densification points N, and the interval step Δt_{in} are calculated (Lines 8-12). New sampling points are then generated on the path segment using cubic interpolation and inserted into the densified path (Lines 13-17). The function *CubicInterpolate*(P_1, P_2, P_3, t) generates the interpolation point P_{den} using cubic interpolation, and *InsertPointPath*(*Path*$_s$, P_{den}, $i+1$) inserts it into the specified position of the *Path*$_s$. Finally, the densified *Path*$_s$ is returned (Line 20). This process ensures effective densification of sampling points in high-curvature areas while maintaining path smoothness.

Figure 3 (c) illustrates the re-wiring strategy of ASD-RRT* for the densely sampled points after refinement, where P_{new} and P_i represent the newly generated state nodes by the cost-based RRT* and the existing state vertices in the random tree T, respectively. ASD-RRT* introduces a novel rewiring method that incorporates the motion dynamics constraints of the autonomous vehicle in addition to the geometric approach. The rewiring between state nodes P_i and P_{new} must satisfy the following conditions: the extension lines of P_i and P_{new} intersect at point P, and the distance between P_i and P (denoted by S_i) is greater than the distance between P_{new} and P (denoted by S_n). The point P_s is defined as the point between P and P_i, and the distance between P and P_s is equal to S_n. P_s helps to generate a trajectory between P_{new} and P_i via the arc $\overline{P_{\text{new}}P_s}$ and the line $\overline{P_sP_i}$. The point O represents the center of the arc and ρ represents the curvature. P_s can be considered a virtual node with the same orientation as the node P_i. The angle subtended by the arc is equal to the change in direction from P_{new} to P_i, denoted by $\Delta\theta$. The rewiring process is also divided into an arc phase and a straight line phase.

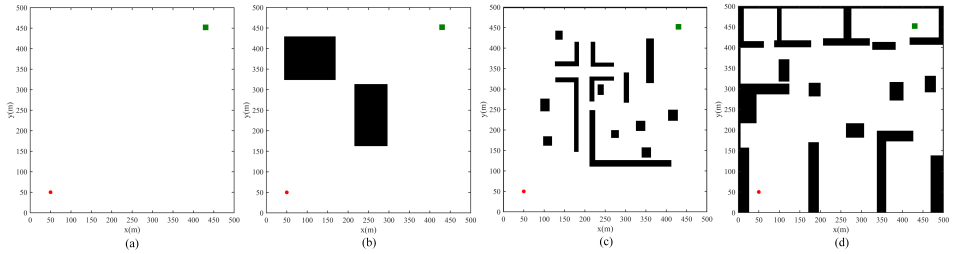


Fig. 4. Environments for the simulations: (a) Black; (b) Simple; (c) Narrow; (d) Closed Trap

4. Simulation and Experiment

To verify the performance of the ASD-RRT* algorithm, this study conducted a comparative analysis of the RRT, RRT*, Informed-RRT*, and ASD-RRT* algorithms across the four scenarios illustrated in Figure 4. The grid map size was 500 × 500 pixels with a resolution of 1 pixel. All algorithms used the red point with coordinates (50, 48) as the starting point and the green rectangle with coordinates (430, 450) as the global target area. The red curves represented the final planned paths. The tree's expansion step was set to 5

pixels, and the pruning radius was set to 10 pixels. The maximum sampling value was set to 3000, if the number of samples exceeded this value, the path planning was considered a failure.

Algorithm performance was evaluated by execution time, final path length, and the average number of iterations. Due to the random nature of the algorithms, the average of 100 simulation experiments was used for evaluation. The simulation platform was MATLAB R2022a, and all algorithms used the min function to find neighboring nodes when expanding the tree nodes. The collision detection program employed the incremental sampling method.

4.1. Simple Environments

The experimental scenario was divided into two parts: one part was the case where there were no obstacles between the start and end points, and the other part was the case where there were very few obstacles, corresponding to maps (a) and (b) in Figure 4. The average path length and the difference from the optimal path were employed to evaluate the algorithm's performance. Since there are no simplifiable obstacles in the maps, the threshold for simplified grid maps in the ASD-RRT* algorithm could be set to any value, which was set to 0 in this instance.

Black Environment Figure 5 illustrates the planning scenarios of the four algorithms in a simple environment. This scenario primarily evaluates the performance loss due to additional computational costs. In an empty scene, the RRT algorithm finds a path by randomly sampling the state space and incrementally building a tree that connects the free space. Due to its randomness, the generated path contains many unnecessary twists. RRT* enhances RRT by optimizing the path and reconnecting surrounding nodes to reduce its twistiness. Informed-RRT* further optimizes path quality by focusing heuristic sampling in more optimal areas. ASD-RRT*, by introducing adaptive sampling regions, effectively reduces invalid samples, resulting in a more direct and smoother path.

Table 2 presents the performance of the algorithms in scenario (a). All four algorithms achieved a planning success rate of 1.00. However, the path planned by ASD-RRT* is nearly optimal, being 7.97% shorter than that of Informed-RRT*, 11.1% shorter than RRT*, and 21.18% shorter than RRT.

Simple Environment Figure 6 presents the planning results of the four algorithms in scenario (b). This scenario primarily evaluates the impact of non-simplifiable obstacles on ASD-RRT*. The RRT algorithm does not consider the global optimality of the path when expanding nodes, resulting in longer and less smooth paths when encountering obstacles. RRT* improves path length to some extent through re-connection operations when expanding nodes in the presence of obstacles. Both Informed-RRT* and ASD-RRT* reduce ineffective random sampling to varying degrees, significantly improving sampling speed and path smoothness.

As shown in Table 3, the success rate of RRT in this scenario is only 0.98, while RRT*, Informed-RRT*, and ASD-RRT* all achieve a success rate of 1.00. The average path length planned by ASD-RRT* is reduced by 21.62% compared to RRT, by 10.13% compared to RRT*, and by 7.61% compared to Informed-RRT*. This demonstrates ASD-RRT*'s ability to generate more effective global paths, resulting in superior path quality.

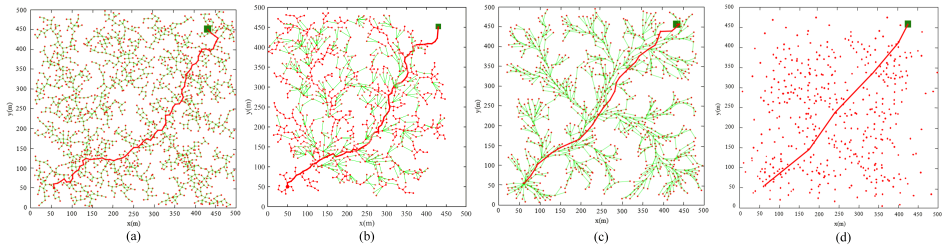


Fig. 5. Performance comparison of four algorithms in environment (a). (a) RRT; (b) RRT*; (c) Informed-RRT*; (d) ASD-RRT*

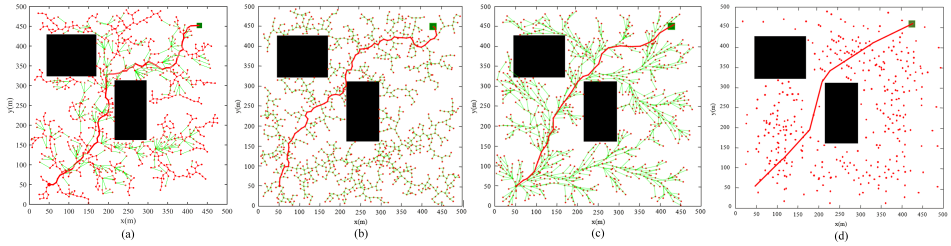


Fig. 6. Performance comparison of four algorithms in environment (a). (a) RRT; (b) RRT*; (c) Informed-RRT*; (d) ASD-RRT*

Table 1. Results for Planning in the Simple Environment (a)

Algorithm	Avg. Length of Path	Success Rate
RRT	718.75	1.00
RRT*	637.24	1.00
Informed-RRT*	615.52	1.00
ASD-RRT*	566.49	1.00

Table 2. Results for Planning in the Simple Environment (b)

Algorithm	Avg. Length of Path	Success Rate
RRT	748.23	0.98
RRT*	652.62	1.00
Informed-RRT*	634.82	1.00
ASD-RRT*	586.48	1.00

4.2. Complex Environments

This section validates the effectiveness of global path planning algorithms in complex scenarios. The experimental setup includes maps with multiple obstacles and narrow passages, corresponding to maps (c) and (d) in Figure 4. The performance of the algorithms is evaluated based on the average number of iterations, average computation time, average length of the final planned path, and planning success rate. In these experiments, the

scenes contain numerous obstacles, and the threshold for simplifying the grid map in the ASD-RRT* algorithm is set to 100, while all other parameters remain unchanged.

Narrow Environment The overlay reproduction process of the ASD-RRT* algorithm in environment (c) is illustrated in Figure 7. In the experiments, the obstacle overlay reproduction scale factor is set to 2, meaning that the obstacles are reproduced in two batches based on their area. Obstacles with an area between 50 and 100 are reproduced in the first batch, while those with an area less than 50 are reproduced in the second batch. By restricting the sampling area, invalid sampling is effectively reduced. The staged overlay reproduction of obstacles allows the algorithm to gradually adjust the planned path to avoid obstacles and navigate through narrow passages, thus better adapting to environments with multiple obstacles and narrow paths.

The simulation results of the four algorithms in map (c) are shown in Figure 8. This scenario primarily evaluates the algorithms' ability to navigate through narrow passages with multiple obstacles. The paths planned by RRT and ASD-RRT* are roughly similar, but the RRT path exhibits significantly more sharp turns and is more convoluted, whereas the ASD-RRT* path is noticeably smoother. The paths planned by RRT* and Informed-RRT* are also similar to each other but are significantly longer compared to the path generated by the ASD-RRT* algorithm.

Table 4 presents the performance of the algorithms in the narrow environment (c). The success rates for RRT and Informed-RRT* were only 0.88 and 0.95, respectively, while both RRT* and ASD-RRT* achieved a success rate of 1.00. The average number of iterations for RRT was 3.9 times that of ASD-RRT*, while the average number of iterations for RRT* and Informed-RRT* were close to each other, being 3.4 and 3.1 times that of ASD-RRT*, respectively. In terms of average time spent, RRT was 3.8 times longer than ASD-RRT*, while RRT* and Informed-RRT* were 2.4 and 2.0 times longer than ASD-RRT*, respectively. This demonstrates the high efficiency of the ASD-RRT* algorithm. With regard to the average path length of the four algorithms, ASD-RRT* was reduced by 25.10% compared to RRT, and by 18.70% and 12.46% compared to RRT* and Informed-RRT*, respectively.

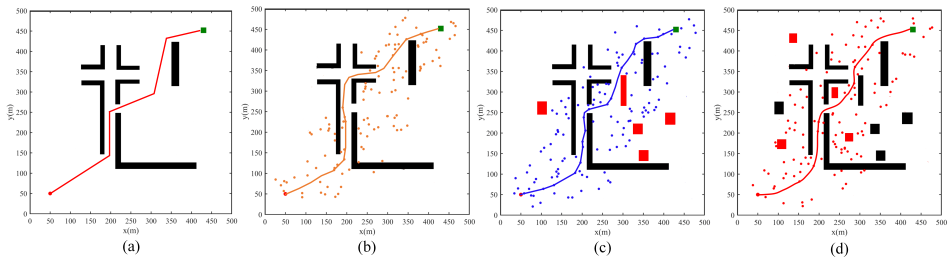


Fig. 7. The planning process of the ASD-RRT* algorithm in map (c). (a) The initial guiding path on the simplified map; (b) The RRT* initial planned path; (c) The result of the first overlay reproduction of large obstacles; (d) The result of the second overlay reproduction of small obstacles

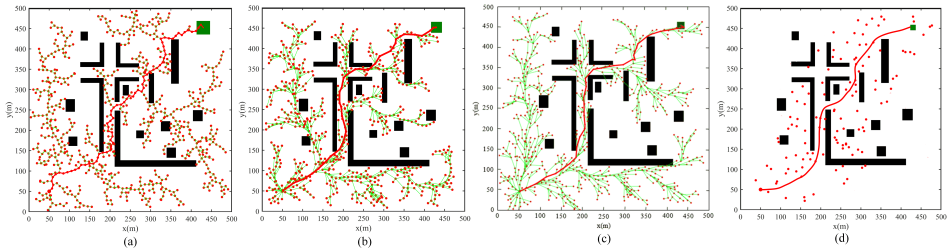


Fig. 8. Performance comparison of four algorithms in environment (a). (a) RRT; (b) RRT*; (c) Informed-RRT*; (d) ASD-RRT*

Table 3. Results for Planning in the Complex Environment (c)

Algorithm	Avg. Iteration	Avg. Time(s)	Avg. Length of Path	Success Rate
RRT	895	0.428	805.82	0.88
RRT*	794	0.274	742.38	1.00
Informed-RRT*	724	0.225	689.45	0.94
ASD-RRT*	232	0.113	603.53	1.00

Closed Environment With Multiple Obstacles The simulation results for the four algorithms in map (d) are depicted in Figure 9. This scenario primarily evaluates the performance of the algorithms in a complex environment with multiple trap-like obstacles. The structures of the four trees are generally similar, but the trajectories generated by RRT and RRT* exhibit more tortuosity and contain more inflection points compared to those produced by Informed-RRT* and ASD-RRT*. In this trap environment, the paths planned by the other three algorithms are notably more convoluted with unnecessary detours. Leveraging the initial guiding path, ASD-RRT* efficiently plans a feasible trajectory in such challenging environments.

The violin plot in Figure 10 illustrates the performance of the four algorithms in map (d). The average number of iterations for RRT was 4.2 times higher than for ASD-RRT*, RRT* was 3.7 times higher, and Informed-RRT* was 3.4 times higher. Regarding average computation time, RRT took three times longer than ASD-RRT*, and RRT* and Informed-RRT* were 2 and 1.8 times longer, respectively. In terms of average path length, ASD-RRT* was 25.56% shorter than RRT, and 22.67% and 18.43% shorter than RRT* and Informed-RRT*, respectively.

In conjunction with the aforementioned experimental analyses, the ASD-RRT* algorithm enhances convergence speed and sampling efficiency. Post-processing of the path results in a smoother global trajectory. Even within complex scenarios featuring multiple obstacles and narrow passages, the algorithm rapidly generates a safe and smooth trajectory that complies with vehicle dynamics. This capability is facilitated by the initial solution, adaptive sampling area, and obstacle superposition reproduction strategy.

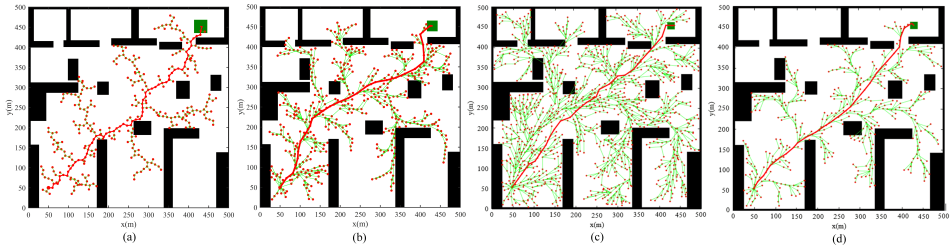


Fig. 9. Performance comparison of four algorithms in environment (a). (a) RRT; (b) RRT*; (c) Informed-RRT*; (d) ASD-RRT*

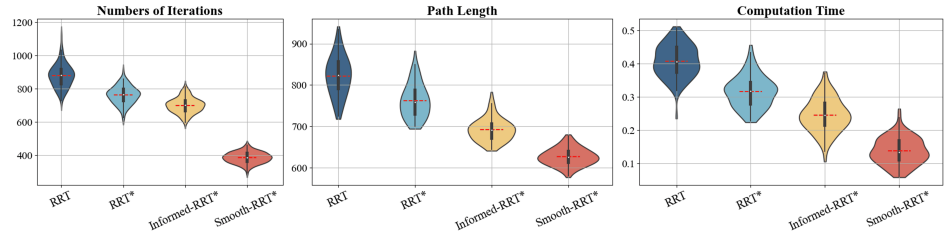


Fig. 10. Performance comparison of four algorithms in environment (a). (a) RRT; (b) RRT*; (c) Informed-RRT*; (d) ASD-RRT*

5. Conclusions

In this paper, we propose an adaptive sampling ASD-RRT* algorithm aimed at addressing challenges such as low search efficiency, slow convergence speed, and convoluted planning paths encountered in traditional RRT* algorithms. ASD-RRT* initiates by simplifying the global raster map, retaining only the larger obstacles. This simplified map guides the initial path generation process. Subsequently, the algorithm expands this initial path within an adaptive sampling area, employing adaptive sampling regions to enhance search efficiency. Concurrently, it iteratively plans around obstacles using the unsimplified global raster map to ensure the path's practicality. The algorithm further integrates a densification technique for sampling in scenarios characterized by high curvature, accompanied by a dynamic smoothing optimization process that complies with dynamic constraints. Such post-processing of the paths produced ensures their continuity and smoothness, thereby improving the efficiency of path tracking. Comprehensive experiments demonstrate that our method outperforms both RRT and RRT* in terms of convergence speed and path length. Even in narrow passages with multiple obstacles, the average number of iterations is only 25.9% of that required by RRT*, and the time spent is reduced by 58.76%. Compared to Informed-RRT*, which also employs a constrained sampling strategy, our method achieves a 56.4% reduction in average iterations, a 47.92% decrease in average computation time, and an 11.19% reduction in the average planned path length. Extensive multi-scenario experimental results validate the effectiveness of the proposed method, significantly enhancing motion planning search time and path quality.

Due to the inherent randomness of the RRT algorithm, the ASD-RRT* algorithm cannot guarantee identical trajectories under identical scenarios. Future research will concentrate on enhancing trajectory stability and aligning trajectories more closely with the global optimal path.

References

1. L. Chen and et al. Milestones in autonomous driving and intelligent vehicles: Survey of surveys. *IEEE Trans. Intell. Veh.*, 8(2):1046–1056, Feb. 2023.
2. Y. Ma, Z. Wang, H. Yang, and L. Yang. Artificial intelligence applications in the development of autonomous vehicles: a survey. *IEEE/CAA J. Autom. Sinica*, 7(2):315–329, March 2020.
3. F. Leon and M. Gavrilescu. A review of tracking and trajectory prediction methods for autonomous driving. *Mathematics*, 9(660), 2021.
4. Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE Trans. Intell. Veh.*, 7(3):652–674, Sept. 2022.
5. L. Chen and et al. Milestones in autonomous driving and intelligent vehicles—part ii: Perception and planning. *IEEE Trans. Syst. Man Cybern. Syst.*, 53(10):6401–6415, Oct. 2023.
6. X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu. Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. *IEEE/ASME Trans. Mechatronics*, 21(2):740–753, April 2016.
7. B. Li, Y. Ouyang, L. Li, and Y. Zhang. Autonomous driving on curvy roads without reliance on frenet frame: A cartesian-based trajectory planning method. *IEEE Trans. Intell. Transp. Syst.*, 23(9):15729–15741, Sept. 2022.
8. F. Bounini, D. Gingras, H. Pollart, and D. Gruyer. Modified artificial potential field method for online path planning applications. In *Proc. IEEE Intell. Veh. Symp. (IV)*, pages 180–185, Los Angeles, CA, USA, 2017.
9. Y. Tian, X. Zhu, D. Meng, X. Wang, and B. Liang. An overall configuration planning method of continuum hyper-redundant manipulators based on improved artificial potential field method. *IEEE Robot. Autom. Lett.*, 6(3):4867–4874, July 2021.
10. W. Lim, S. Lee, M. Sunwoo, and K. Jo. Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. *IEEE Trans. Intell. Transp. Syst.*, 19(2):613–626, Feb. 2018.
11. J.-N. Yuan, L. Yang, X.-F. Tang, and A.-W. Chen. Autonomous vehicle motion planning based on improved rrt* algorithm and trajectory optimization. *Acta Automatica Sinica*, 48(12):2941–2950, 2022.
12. L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng. Efficient sampling-based motion planning for on-road autonomous driving. *IEEE Trans. Intell. Transp. Syst.*, 16(4):1961–1976, Aug. 2015.
13. A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip. Motion planning networks: Bridging the gap between learning-based and classical motion planners. *IEEE Trans. Robot.*, 37(1):48–66, Feb. 2021.
14. T. Kim, H. Lee, and W. Lee. Physics embedded neural network vehicle model and applications in risk-aware autonomous driving using latent features. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pages 4182–4189, Kyoto, Japan, 2022.
15. F. Ye, S. Zhang, P. Wang, and C.-Y. Chan. A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles. In *Proc. IEEE Intell. Veh. Symp. (IV)*, pages 1073–1080, Nagoya, Japan, 2021.
16. Z. Qin, T.-W. Weng, and S. Gao. Quantifying safety of learning-based self-driving control using almost-barrier functions. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pages 12903–12910, Kyoto, Japan, 2022.

17. L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12(4):566–580, Aug. 1996.
18. S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Iowa State University, 1998.
19. R. Bohlin and L. E. Kavraki. Path planning using lazy prm. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, volume 1, pages 521–528, San Francisco, CA, USA, 2000.
20. K. Belghith, F. Kabanza, L. Hartman, and R. Nkambou. Anytime dynamic path-planning with flexible probabilistic roadmaps. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2372–2377, Orlando, FL, USA, 2006.
21. S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.*, 30(7):846–894, June 2011.
22. L. Janson, E. Schmerling, A. Clark, and M. Pavone. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *Int. J. Robot. Res.*, 34(7):883–921, June 2015.
23. M. Fuad and S. Wahyuni. Path planning and smoothing in maze exploration using virtual mobile robot-based modified probabilistic road map. In *Proc. IEEE 8th Inf. Technol. Int. Seminar (ITIS)*, pages 15–20, Surabaya, Indonesia, 2022.
24. D. J. Webb and J. van den Berg. Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 5054–5061, Karlsruhe, Germany, 2013.
25. D. Armstrong and A. Jonasson. Am-rrt*: Informed sampling-based planning with assisting metric. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 10093–10099, Xi'an, China, 2021.
26. Z. Sun, B. Xia, P. Xie, X. Li, and J. Wang. Namr-rrt: Neural adaptive motion planning for mobile robots in dynamic environments. *IEEE Transactions on Automation Science and Engineering*, 22:13087–13100, 2025.
27. S. Klemm and et al. Rrt*-connect: Faster, asymptotically optimal motion planning. In *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, pages 1670–1677, Zhuhai, China, 2015.
28. Ahmed Hussain Qureshi and Yasar Ayaz. Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments. *Robotics and Autonomous Systems*, 68:1–11, 2015.
29. Han Ma, Fei Meng, Chengwei Ye, Jiankun Wang, and Max Q.-H. Meng. Bi-risk-rrt based efficient motion planning for autonomous ground vehicles. *IEEE Transactions on Intelligent Vehicles*, 7(3):722–733, 2022.
30. Jiankun Wang, Wenzheng Chi, Chenming Li, and Max Q.-H. Meng. Efficient robot motion planning using bidirectional-unidirectional rrt extend function. *IEEE Transactions on Automation Science and Engineering*, 19(3):1859–1868, 2022.
31. Ying Zhang, Heyong Wang, Maoliang Yin, Jiankun Wang, and Changchun Hua. Bi-am-rrt*: A fast and efficient sampling-based motion planning algorithm in dynamic environments. *IEEE Transactions on Intelligent Vehicles*, 9(1):1282–1293, 2024.
32. J. Yao, Z. Fu, Z. Fang, Z. Guo, and F. Jing. Neural-guided rrt*: Learning-based planning of entry point and puncture path for steerable bevel-tip needle insertion. *IEEE Robotics and Automation Letters*, 10(9):9016–9023, 2025.
33. Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2997–3004, 2014.
34. Z. Tu, W. Zhuang, Y. Leng, and C. Fu. Accelerated informed rrt*: Fast and asymptotically path planning method combined with rrt*-connect and apf. In *Intelligent Robotics and Applications*, pages 279–292, Singapore, 2023. Springer Nature Singapore.

35. N. Li and S. I. Han. Adaptive bi-directional rrt algorithm for three-dimensional path planning of unmanned aerial vehicles in complex environments. *IEEE Access*, 13:23748–23767, 2025.
36. Y. Su, J. Xin, and C. Sun. Dynamic path planning for mobile robots based on improved rrt* and dwa algorithms. *IEEE Transactions on Industrial Electronics*, pages 1–10, 2025.
37. D. Uwacu, A. Yammanuru, K. Nallamotu, V. Chalasani, M. Morales, and N. M. Amato. Has-rrt: Rrt-based motion planning using topological guidance. *IEEE Robotics and Automation Letters*, 10(6):6223–6230, 2025.
38. Constantin Wellhausen, Joachim Clemens, and Kerstin Schill. Efficient grid map data structures for autonomous driving in large-scale environments. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2855–2862, 2021.

Wang Chao is an Associate Professor at the School of Computer Science, Beijing Information Science and Technology University, China. His main research interests are directed to unmanned systems, including object detection, navigation, and decision-making.

Wenbin Li is a Master's student at the School of Computer Science, Beijing Information Science and Technology University, China. His primary research focuses on path planning for unmanned driving.

Received: June 12, 2025; Accepted: October 30, 2025.

Computer Science and Information Systems 23(1):133–164 https://doi.org/10.2298/CSIS250613005L

ADN-YOLO: An Improved Ship Detection Model Based on YOLOv11

Tao Li[1], Dezhi Han[1], Songyang Wu[2⋆], Xiang Shen[1,3], Liqi Zhu[1], and Wenqi Sun[2]

[1] Shanghai maritime university, 1550 Pudong Ave
201306 Shanghai, China
202430310151@stu.shmtu.edu.cn
dezhi@shmtu.edu.cn
shenxiang1107@163.com
zhuliqi0309@163.com
[2] Network Security Center, The Third Research Institute of the Ministry of Public Security
200031 Shanghai, China
wusongyang@stars.org.cn
sunwenqi@gass.ac.cn
[3] School of Computer Science, The University of Sydney, Sydney, NSW2006.Australia
shenxiang1107@163.com

Abstract. Existing infrared imaging techniques have garnered considerable attention and have achieved notable progress in all weather ship target detection tasks, owing to their robustness against varying ambient lighting conditions. However, due to the inherent limitations of infrared images,such as low spatial resolution and insufficient texture information the performance of multi-scale ship target detection remains suboptimal. These challenges significantly hinder the overall improvement of detection accuracy. To address this issue and enhance the detection performance of multi-scale ship targets, particularly small ones, in infrared imagery,this paper proposes an improved You Only Look Once (YOLO) based detection model named ADN-YOLO. The model first introduces a Dynamic Upsampler (Dysample) module, which more effectively integrates semantic information across different layers. This integration balances low level detailed features with high level semantic representations, thereby enhancing the model's ability to perceive target edges and structural characteristics. Second, a lightweight downsampling module (ADown) is incorporated to reduce the parameter count while improving both the efficiency and representational capacity of feature extraction. Additionally, to address the issue of small targets being highly sensitive to localization errors, a new loss function is designed based on the Wasserstein distance. This function combines the Normalized Wasserstein Distance (NWD) with the Complete Intersection over Union (CIoU), thereby enhancing the model's ability to accurately localize small targets. Comprehensive experimental validation is conducted on a marine infrared target detection dataset. Compared to the standard YOLOv11 model, the proposed ADN-YOLO reduces the number of parameters by 20.3%, achieves a 1.9%increase in mAP, a 1.9% boost in Recall, and lowers FLOPs by 1.1G, demonstrating its effectiveness and practicality for infrared image target detection tasks.

Keywords: Target detection; Infrared images; Multi-scale objects; Deep learning;

⋆ Corresponding author

1. Introduction

With the deepening development of the modern technological revolution, the development of marine resources has risen to become a strategic priority for major coastal nations worldwide[1]. In this context, intelligent target recognition technology for maritime applications has emerged as a core enabler for ensuring marine safety. Particularly in nearshore waters characterized by high vessel traffic density and a mix of heterogeneous vessels, the ability to accurately identify and dynamically monitor diverse targets such as cargo ships, fishing vessels, and passenger ferries has become a critical challenge for enhancing maritime governance efficiency and risk management capabilities [2]. The existing ship target detection system primarily relies on three technological pathways: visible light imaging, synthetic aperture radar (SAR), and infrared thermal imaging[3][4]. While visible light imaging offers significant advantages such as high spatial resolution, rich texture features, and a wide detection range, its target recognition accuracy in low-light reflective environments is inherently limited. SAR technology, while enabling all-weather and long-range detection capabilities, is subject to multiple constraints in marine applications, including electromagnetic interference, land-based obstruction effects, port clutter, and atmospheric attenuation. In contrast, infrared imaging technology demonstrates unique advantages. Its passive radiation reception mechanism not only ensures the stealthiness of the detection process[5] but also effectively overcomes the environmental sensitivity of visible light imaging through features such as smoke interference resistance and independence from lighting conditions.

Traditional infrared ship detection methods primarily employ spatial filtering and feature modeling techniques. Filters based on morphological operations suppress uniform background noise using predefined structural templates. However, in non-uniform interference scenarios such as cloud reflections and solar flares, their linear filtering mechanism struggles to distinguish between the statistical similarity of targets and complex backgrounds, leading to a significant increase in false detection rates. To overcome this limitation, a spectral residual model based on the human visual system (HVS)[6][7][8] attempts to extract salient features from the frequency domain perspective. However, this method lacks sensitivity to dynamic sea surface clutter and struggles to effectively suppress high-frequency interference such as wave breaking. Further developed local contrast methods can enhance edge responses of high-contrast targets but fail to adapt to the thermal radiation attenuation characteristics of targets and backgrounds under low-visibility conditions like sea fog or rain/snow due to their reliance on fixed thresholds. The recently emerging low-rank representation method [9][10] achieves sparse separation of background and targets through matrix decomposition, demonstrating potential in low signal-to-noise ratio scenarios. However, it lacks the ability to model the morphological diversity of small ship targets, especially when targets and wave interference exhibit motion coupling, leading to a sharp increase in separation error. The common shortcomings of the above methods reveal the inherent limitations of traditional algorithms in complex sea conditions. Their modeling paradigm based on artificial feature engineering is unable to effectively characterize the nonlinear coupling relationship between infrared ship targets and dynamic sea backgrounds. This bottleneck has prompted researchers to turn to data-driven deep learning frameworks to break through the representation capabilities of traditional methods through end-to-end feature learning.

Current deep learning-based ship detection algorithms primarily follow two technical approaches: two-stage and one-stage, which continue to evolve. Two-stage detectors utilize a cascaded architecture composed of candidate region generation and fine-grained regression, demonstrating significant advantages in rotating ship detection tasks. For example, Gu et al.[11] developed an improved Faster R-CNN model that constructs multi-scale feature maps through multi-layer convolutional concatenation, thereby obtaining feature vectors with richer semantic information and enhancing model performance. Pang et al.[12] proposed the rotating Libra R-CNN, which introduces a three-dimensional balanced mechanism between the feature layer, sample layer, and target layer, effectively controlling the prediction error of ship azimuth and further improving the detection accuracy of rotating targets. However, the multi-stage processing workflow of such methods introduces significant computational redundancy, severely limiting their real-time performance. In contrast, one-stage detectors (such as the YOLO series) adopt a parallel architecture combining "anchor box prediction and classification regression," significantly improving detection speed while maintaining detection accuracy, better aligning with the real-time response requirements of ship detection tasks. For example, Zheng et al. [13][proposed two novel gated fusion units that effectively fuse feature maps generated by the intermediate layers of SSD, constructing various stacked and gated hybrid fusion versions; Li et al. [14] proposed YOLO-FIRI, which enhances the feature representation capability for small infrared targets through shallow network expansion and multi-scale attention mechanisms; and Luo et al. [15] developed YOLO-IR, which significantly improves target recall rates in low-contrast scenes using a focus loss function. Si et al. [16] proposed an improved YOLO-RSSD algorithm, where an enhanced bidirectional feature pyramid network structure is embedded into the feature fusion component. This enables cross-layer multi-scale weighted feature fusion and introduces a channel attention mechanism in the convolutional unit to further enhance the detection performance of ship targets in infrared images.

Although infrared image detection algorithms that incorporate deep learning have achieved significant performance improvements, their high model complexity and computational costs still pose constraints on practical deployment. As a result, lightweight design has become a research hotspot in recent years. For example, Xing et al. [17] proposed a lightweight detection algorithm named YOLOv8-FAS, which incorporates efficient FasterNet components and attention mechanisms to enhance the feature extraction capabilities of the backbone network, achieving a good balance between performance and efficiency. These methods demonstrate excellent performance in ship image detection under visible light conditions, but they still face numerous challenges in ship detection tasks targeting infrared images. Due to the complex thermal radiation distribution characteristics of infrared images, their physical imaging mechanisms significantly differ from those of visible light images. For example, the thermal wake of a ship is prone to frequency domain aliasing with background thermal radiation, leading to feature confusion; thermal diffusion effects between adjacent targets can cause boundary blurring; and under low-contrast conditions, the saliency information of targets often severely degrades, thereby affecting the visibility and accuracy of detection results. As shown in Figures 1(a) and (b), the day-night temperature difference causes the ocean and land areas to exhibit obvious gray-scale bipolarity [18][19], with high variability in contrast between ships and their backgrounds. Figure 1(c) shows that ship targets in infrared images are typically

small in scale, especially small vessels anchored near the coast, which are easily obscured by complex backgrounds. Additionally, due to the lack of obvious semantic structure in infrared images, existing methods are limited in terms of robustness and generalization capabilities.

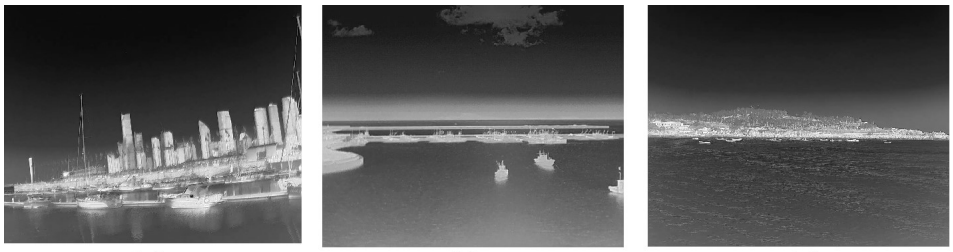


Fig. 1. (a) Targets under close imaging, (b) Targets near a harbor, (c) Multiple targets near a harbor with a complex low-contrast background

In summary, current infrared ship target detection methods continue to face significant challenges, including the difficulty of balancing detection accuracy with real time performance and the limited capability of feature extraction. At the core of these issues is the fact that existing approaches have yet to effectively incorporate the thermophysical priors inherent to infrared imaging.To overcome these limitations, there is an urgent need to develop a hybrid intelligent detection framework that seamlessly integrates thermodynamic a priori knowledge with data driven learning mechanisms. Such a framework would be better equipped to tackle the fundamental challenges of infrared ship detection, including complex background interference, low contrast imaging, and target blurring, ultimately advancing both the robustness and generalization capabilities of detection systems in real world maritime environments.

To address the aforementioned challenges, this paper proposes an infrared ship detection method based on an improved YOLOv11 architecture, named ADN-YOLO (Attention-guided Downsample and Normalized Distance enhanced YOLO). The proposed method significantly enhances the detection accuracy and feature discrimination for small- and medium-scale targets in infrared images, while preserving the lightweight characteristics essential for real-time deployment. Specifically, the backbone of YOLOv11 is modified by introducing an improved downsampling module, ADown, to replace parts of the original CBS (Conv-BN-SiLU) structure. ADown integrates an attention mechanism with a dynamic convolution strategy, effectively reducing both the parameter count and computational complexity. This design notably boosts the expressive power of the feature extractor, demonstrating superior robustness in typical infrared scenarios characterized by low contrast and complex backgrounds. To further address the issue of small targets being easily obscured by background information, a point-based dynamic upsampling module, Dysample, is introduced. Dysample adaptively enhances the spatial detail representation of low-resolution feature maps, effectively preserving shallow semantic and positional information. This enables the model to better perceive and detect multiscale targets, par-

ticularly weak infrared signals.In addition, this paper proposes a novel loss function that combines normalized Wasserstein distance with traditional bounding box regression. By modeling both the predicted and ground truth boxes as two-dimensional Gaussian distributions,and computing their normalized Wasserstein distance, this metric captures the distributional differences between predicted and actual targets. This approach supplements the geometric constraints of CIoU by incorporating statistical distribution information, thereby enhancing the accuracy of bounding box regression, particularly for small and irregularly shaped targets, while also improving the model's generalization ability and robustness.The main contributions of the proposed ADN-YOLO model are as follows:

(1) An improved detection model, ADN-YOLO, based on YOLOv11 is proposed specifically for infrared ship detection scenarios. By incorporating convolutional enhancement and a dynamic downsampling mechanism, the model effectively improves the detection capability of weak and low-contrast targets.

(2) A novel ADown downsampling module is designed to replace the original CBS structure, reducing model complexity while simultaneously enhancing the expressive power of feature extraction.

(3) The Dysample module is introduced to enhance the model's multi-scale perception of small targets, effectively addressing the detection challenges posed by blurred and low-contrast ships in infrared images.

(4) The NWD-CIoU loss function is proposed, which fuses the normalized Wasserstein distance with CIoU to address the robustness limitations of traditional IoU in small target detection, thereby improving both localization accuracy and overall detection performance.

The remainder of this paper is organized as follows: Section 2 reviews key recent research in the field of target detection. Section 3 provides a detailed description of the YOLOv11 model and the improvements introduced in our proposed method. Section 4 presents the experimental results of our model and compares its performance with several other target detection approaches. Finally, Section 5 concludes the paper by summarizing the findings and discussing the limitations of the proposed model.

2. Related work

In recent years, researchers have proposed a variety of strategies and algorithms to cope with the problems of large computational overhead, significant scale changes, and difficulty in effectively extracting target features in complex backgrounds faced by infrared images during processing. On the basis of reviewing the traditional target detection methods for infrared images, this paper focuses on the target detection technology based on deep learning, systematically combs through the main improvement directions of single-phase and two-phase methods, and discusses in depth the limitations and challenges of each.

2.1. Traditional infrared image detection

Traditional infrared target detection methods typically model infrared images as consisting of three components: the target, the background, and noise. The core idea is to

enhance the saliency of the target region by effectively suppressing background interference and image noise, thereby achieving more accurate target detection. These methods primarily rely on classical image processing techniques or manually designed feature extraction strategies and do not incorporate large-scale data-driven learning mechanisms, resulting in certain limitations under specific conditions. Among numerous traditional methods, detection algorithms based on the human visual system (HVS) are representative. For example, the Three-Layer Local Contrast Measurement (TLLCM) method enhances the saliency of the target region by simulating the human eye's sensitivity to local contrast, thereby improving target detectability. Although these methods have improved the response capability to target regions to some extent, they rely on fixed contrast enhancement mechanisms and struggle to adapt to the varying features of targets in complex backgrounds. To further enhance target detection performance, Zhao and Kong[20] proposed a spatial filtering-based infrared image target detection method. This method designs filters based on the differences in grayscale distribution between targets and backgrounds to selectively suppress background regions and highlight target regions. However, this method performs poorly when handling isolated small noise points, often leading to misclassification or omission of target regions, thereby limiting overall detection accuracy. To address these issues, Anju and Raj [21] proposed a frequency-domain-based target detection method inspired by frequency-domain analysis. This method analyzes the differences in frequency distribution between the target and background in an image, decomposing the image into high-frequency components (corresponding to target information) and low-frequency components (corresponding to background information), thereby achieving more effective target extraction. Experiments show that compared to spatial filtering methods, frequency-domain filtering performs better in suppressing background interference and improves detection accuracy, but its computational complexity significantly increases, making it unsuitable for real-time application deployment. Additionally, Jiao[22] proposed modeling infrared image target detection as a problem of separating low-rank matrices and sparse matrices, and introduced sparse representation methods to achieve precise extraction of target regions in images. This method outperforms traditional filtering methods in terms of signal-to-clutter ratio (BCR) and background suppression factor (BSF), demonstrating excellent target detection capabilities. However, due to the non-local autocorrelation characteristics of infrared image backgrounds, the distinction between targets and backgrounds is often blurred, leading to limitations in the method's suppression capabilities under complex backgrounds. In summary, while traditional infrared image detection methods still hold practical value in specific scenarios, the increasing miniaturization of targets, growing complexity of backgrounds, and continuous decline in BCR make the trade-off between detection accuracy and processing efficiency increasingly challenging. Additionally, their high computational resource consumption makes it challenging to meet the dual requirements of speed and energy efficiency for real-time detection systems. Therefore, there is an urgent need for more efficient and robust detection methods to address target recognition tasks in complex infrared scenarios.

2.2. Deep learning Based InfiRay image detection

Deep learning, a key branch of computer vision, has achieved significant advancements in tasks such as target detection. By building deep neural network models with

multilayer convolutional architectures, these methods automatically learn high level semantic features from images, thereby substantially improving target detection accuracy in complex scenes.

Two-stage object detectors In 2014, R. Girshick et al. [23] proposed the groundbreaking two-stage object detector R-CNN, whose core idea is to divide object detection into two steps: generating proposals and classification regression prediction. Although this method achieved certain breakthroughs in detection accuracy, its high computational complexity, lack of weight sharing, and slow inference speed limited its practical application. To enhance detection efficiency and accuracy, Fast R-CNN [24] and Faster R-CNN [25] were subsequently proposed, introducing ROI (Region of Interest) pooling operations and the Region Proposal Network (RPN), respectively, significantly improving overall performance. In the field of infrared image object detection, Ghose et al. [26] proposed an improved method based on the classic Faster R-CNN, incorporating saliency maps to enhance key regions in infrared images. However, the saliency network in this method was not trained in an end-to-end, multi-task joint training manner with Faster R-CNN, resulting in a time-consuming training process. Devaguptapu et al. [27] developed a multimodal Faster R-CNN that utilizes RGB channels to extract high-level infrared features, further improving detection performance. However, this multimodal fusion strategy significantly increases training computational overhead. Overall, although two-stage detectors excel in terms of accuracy, their high computational complexity and large inference latency make them unsuitable for real-time object detection requirements, particularly in applications with stringent timeliness demands such as ship monitoring. Additionally, their detection capabilities for small objects are also suboptimal. Therefore, when selecting ship target detection algorithms, it is essential to fully consider actual application requirements and scene characteristics, balancing detection accuracy with computational efficiency. Chen et al. [28] proposed the CAAN model, which designs a new method for calculating absolute positions based on the coordinates of each image region and the actual size of the image.

Single-stage object detectors To address the limitations of two-stage detectors in terms of computational efficiency and real-time performance, researchers have proposed single-stage object detection algorithms, with the YOLO (You Only Look Once) series being a representative example. This series of algorithms divides the input image into multiple grid units and detects each unit as a potential target candidate region. YOLO leverages a fully convolutional neural network architecture to integrate object classification and bounding box regression into a unified end-to-end processing workflow. Compared to traditional two-stage methods (such as Faster R-CNN), YOLO significantly improves computational efficiency and inference speed while maintaining detection accuracy. Therefore, it demonstrates excellent application potential and broad prospects in scenarios with high requirements for real-time performance and low latency, such as video surveillance and autonomous driving. YOLO's efficient detection capabilities have inspired a large number of related studies. For example, Li et al. [29] proposed a cross-layer attention network to enhance the feature expression capabilities of small objects in visible light images. In recent years, positive progress has also been made in the detection of weak small objects in infrared images. Guo et al. [30] designed a bidirectional attention feature pyramid network (BAFPN) for near-shore vessel detection, significantly improving detection performance for miniature vessels. Chen et al. [31] utilized optimized global context information to dynamically modulate and filter image features, combined with

spatial location information, to further enhance the modeling of image label dependencies and the model's inference capabilities. Shen et al. [32] proposed a sparse attention network (GFSNet) that overcomes the limitations of traditional spatial domain methods, enhancing the network's ability to perceive multi-scale features and extract significant regions. Although deep learning-based infrared image ship detection technology has made significant progress, it still faces numerous challenges that require further research and breakthroughs. First, infrared images themselves have low resolution, are often blurry, and frequently contain significant noise, severely affecting detection accuracy. Second, infrared images are mostly acquired from coastal viewpoints, which, while offering a broader field of view, also introduce more complex background interference. There are still notable shortcomings in terms of small target recognition accuracy and feature expression stability. To address these challenges, this paper proposes an improved object detection model based on the YOLOv11 framework—ADN-YOLO. The detailed structure and implementation of this method will be discussed in the next section.

3. Methodology

Addressing the challenges of marine vessel detection, such as significant multi-scale variation and the weakening of small targets in infrared images, this paper proposes an improved target detection algorithm based on the YOLOv11 framework, named ADN-YOLO. This chapter systematically details the innovative architectural design and key technological enhancements of the method. First, the ADown module is introduced to replace part of the convolutional structure in the YOLOv11 backbone. Its multi-branch design enhances feature fusion and information interaction, thereby improving the detection accuracy of distant small targets. Second, to further strengthen small target perception, the Dysample module is adopted to dynamically upsample low-resolution feature maps, effectively preserving fine target details. Finally, a novel loss function optimization strategy is proposed, combining Normalized Wasserstein Distance (NWD) with Complete IoU (CIoU) to replace the traditional IoU metric, further enhancing target localization accuracy. Specifically, Section 3.1 introduces the fundamental characteristics of the YOLOv11 model. Section 3.2 elaborates on the structure and improvements of the ADN-YOLO algorithm.

3.1. YOLOv11 Modeling

YOLOv11 is the latest iteration in the YOLO series, optimized from YOLOv8 with significant improvements. It features a redesigned backbone and neck architecture, enabling it to outperform YOLOv8 in multi-target detection tasks. Efficient feature extraction is crucial for accurate target localization and classification, and YOLOv11 greatly enhances both detection sensitivity and accuracy. Infrared images lack visible light color information, resulting in weaker texture and boundary features. Although YOLOv8 supports multi-scale detection, it is prone to missed or false detections in infrared scenarios due to the low signal-to-noise ratio.YOLOv11 also introduces a more efficient framework and training procedure. Its detection head incorporates two Depthwise Separable Convolution (DWConv) modules, which substantially reduce the model's parameters and computational load while maintaining accuracy and accelerating processing speed. Similar to

YOLOv8, YOLOv11 uses convolutional layers to downsample the input image, reducing spatial dimensions while increasing channel depth. However, in the backbone, YOLOv11 replaces YOLOv8's C2f module with a new C3k2 module. Additionally, following the SPPF module, the C2PSA module is introduced to enhance spatial attention within the feature maps. The backbone serves as the foundational component of the YOLO network, responsible for extracting high level semantic features from the input image through a series of convolutional and pooling layers. These extracted feature maps are then passed to subsequent network components for target detection. The neck module of YOLOv11 further processes these feature maps to prepare them for detection. Its main function is to extract and fuse multi-scale features, which are then fed to the head network for classification and bounding box regression. Compared to YOLOv8, YOLOv11's use of the C3k2 module in the neck allows for more detailed multi-scale feature extraction. The C3k2 module offers flexibility through parameterization: when set to false, it behaves like the C2f module with a standard bottleneck; when set to true, the C3k2 module replaces the bottleneck with a C3 module, as illustrated in Figure 2.

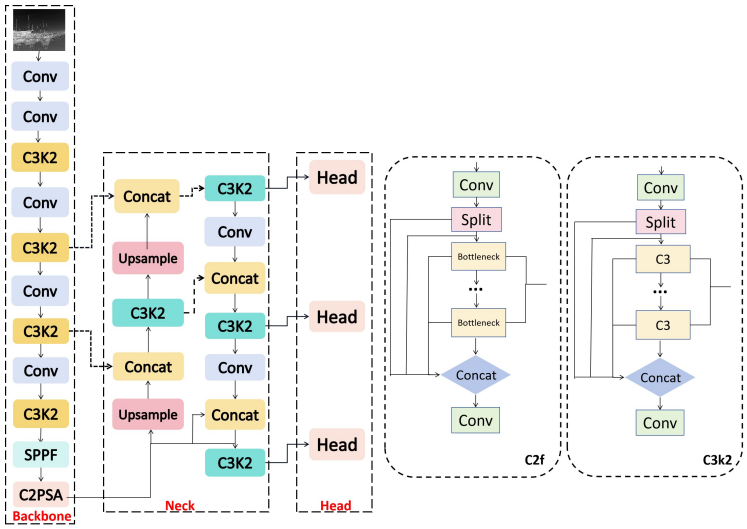


Fig. 2. Comparison of the original YOLOv11 models C2f and C3K2

3.2. Proposed methodology

In this paper, we propose an efficient and lightweight network, ADN-YOLO, designed to address the challenges of missing texture details in small infrared targets caused by low scale and limited spatial resolution, as well as the problem of highly variable multi-scale ship distributions. Instead of conventional convolutions in the feature extraction network, the ADown module is employed to capture locally significant features while preserving global information. It combines average and max pooling to achieve multi-scale feature extraction. The innovative ADown downsampling component plays a crucial role in preserving original high-frequency details, minimizing information loss, and enhancing

computational efficiency. This results in a significant improvement in small-target detection performance while reducing model parameters and increasing detection speed. To further boost feature extraction for small targets, the upsampling process is dynamically adjusted using DySample, which requires fewer parameters and computations than traditional upsampling methods, effectively reducing model complexity and computational resource consumption. Given the common issues of positional errors and shape distortions in small target detection, we combine the Normalized Wasserstein Distance (NWD) loss with the Complete IoU (CIoU) loss. The NWD loss optimizes localization accuracy, while the CIoU loss refines both the shape and position of the target bounding boxes at a finer granularity. The overall structure of the ADN-YOLO network is illustrated in Figure 3.

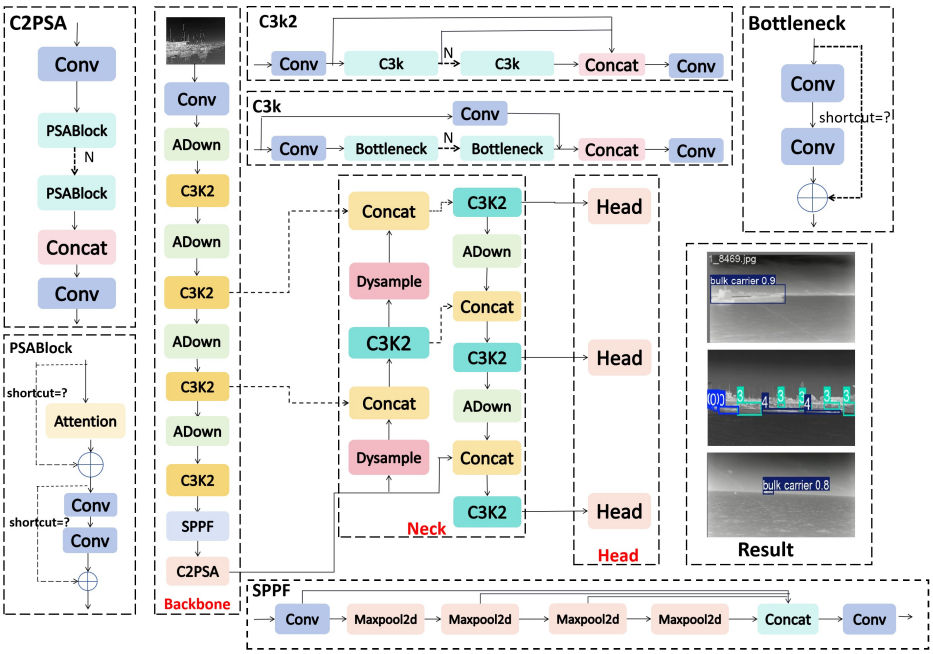


Fig. 3. Structure of ADN-YOLO

ADown Since the resolution of infrared images is lower than that of visible light images, they often appear more blurred, resulting in limited information extraction from small target vessels and the loss of critical details, which negatively impacts detection performance. Additionally, infrared images frequently contain significant noise and numerous distant objects. As the feature map size decreases through the network, fine features of distant targets may be lost due to downsampling, hindering the effective capture of small target details this poses particular challenges for densely packed targets in infrared images. The ADown module addresses this by combining average pooling, which preserves global information, with max pooling, which captures important local features. This dual-pooling strategy enables feature extraction at multiple scales. Therefore, this study in-

tegrates the ADown module into the backbone network of YOLOv11, replacing several conventional convolutional downsampling operations to enhance the model's ability to capture image features more effectively.

As shown in Figure 4., in the ADown downsampling module, the input feature map X first undergoes an average pooling operation. The feature map is sliced to halve its size and divided into two parts along the channel dimension, as shown in Eqs (1)(2). The feature map is then divided into two sub-feature maps along the channel dimension and . for, a 3×3 convolutional layer Conv1 operation with a step size of 2 and padding of 1 is used. And is first subjected to a maximum pooling operation with the aim of retaining important features. Then, a 1×1 convolutional layer Conv2 with step size 1 and padding 0 is used to capture the features, as shown in Eqs(3)-(5).

$$X' = AvgPool(\mathrm{X}, 2, 1, 0)\,. \tag{1}$$

$$\mathrm{X}_1, \mathrm{X}_2 = chunk\,(X', 2, 1)\ . \tag{2}$$

$$\mathrm{X}_1' = Conv\,(\mathrm{X}_1, kernel = 3, stride = 2, padding = 1) \tag{3}$$

$$\mathrm{X}_2' = MaxPool\,(\mathrm{X}_2, kernel = 3, stride = 2, padding = 1) \tag{4}$$

$$\mathrm{X}_2'' = Conv\,(\mathrm{X}_2', kernel = 1, stride = 1, padding = 0) \tag{5}$$

Finally, the two sub feature maps, X_1 and X_2, are concatenated to produce the output feature map Y of the ADown module.

$$Output = Concat\,(\mathrm{X}_1, \mathrm{X}_2, dim = 1) \tag{6}$$

ADown significantly enhances target detection accuracy while reducing false positives and missed detections by effectively suppressing background noise and redundant information. Small targets often suffer from poor localization accuracy and are prone to being overlooked; the ADown module addresses this by extracting rich feature information and fusing multi-scale features to produce more representative feature maps. The introduction of the ADown's multi-branch structure enables greater feature combination and information interaction, preserving more contextual details and preventing the loss of small target features. Whether dealing with images of varying resolutions or complex, diverse backgrounds, ADown allows the network to flexibly adjust its focus, thereby improving overall detection performance and boosting model robustness.

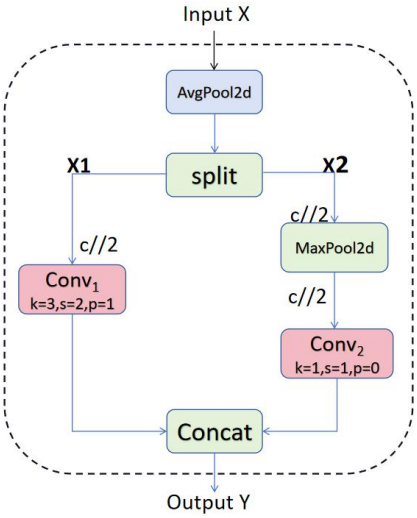


Fig. 4. ADown structure

Dysample DySample introduces an innovative approach that replaces the traditional dynamic convolution based upsampling with a point sampling technique, significantly improving resource efficiency. In the original YOLOv11 model, the UpSample operation imposes substantial computational demands and a large number of parameters, limiting the model's lightweight performance in ship detection tasks. In practical infrared ship monitoring scenarios, infrared images often suffer from pixel distortion, resulting in the loss of fine details that hinder effective feature learning. To address this, DySample was developed as a lightweight and efficient dynamic upsampling module designed to replace the conventional UpSample method. This point-based sampling technique generates content-aware upsampled features concisely and efficiently, without requiring additional high-resolution inputs.DySample reduces model complexity and computational overhead while maintaining high accuracy. Through an appropriate loss function and forward propagation training, it adaptively adjusts sampling point positions and weights, significantly enhancing upsampling accuracy and efficiency. This makes DySample particularly well-suited for the complex target detection tasks addressed in this paper. The flowchart of DySample upsampling is shown in Figure 5. Given an input feature map of size C × H × W and a sampling coordinate set of size 2 × H × W (where the first two dimensions correspond to x and y coordinates), the feature map is resampled using the $grid_sample$ function. Bilinear interpolation is applied to generate a new feature map of size $C \times H_1 \times W_1$, as mathematically expressed in (7).

$$X = grid_sample(X, \delta) \tag{7}$$

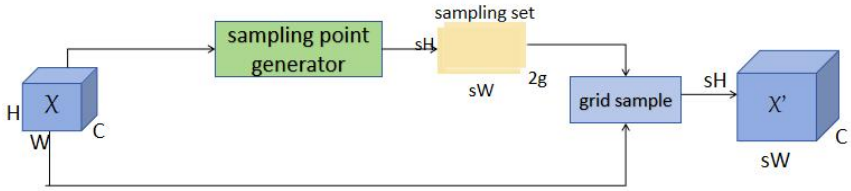


Fig. 5. Sampling process on DySample

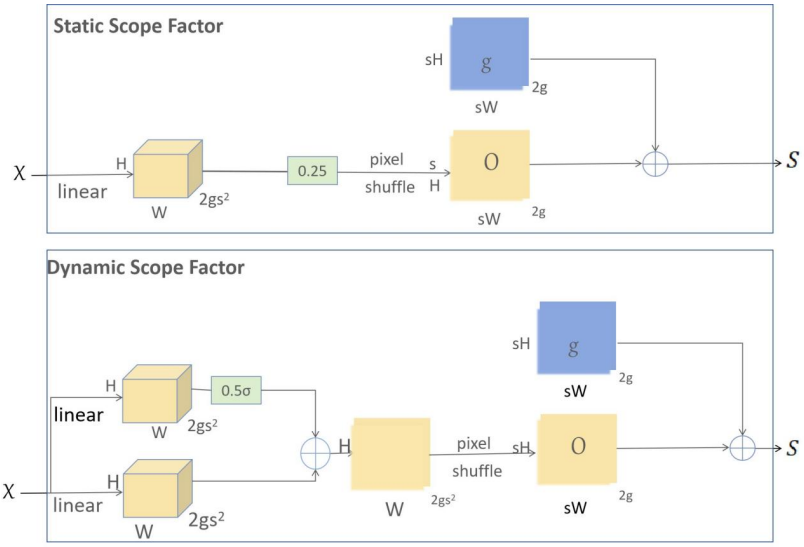


Fig. 6. Point sampling set generation process

As shown in Figure 6, given an upsampling scaling factor s and a feature map X of size $C \times H \times W$, a linear layer (with the number of input and output channels C and $2s^2 \times H \times W$, respectively) is used to generate an offset O of size $2 \times sH \times sW$, which is then reshaped to 2 × sH × sW by Pixel Shuffling [33] . through Pixel Shuffling, each pixel of the original feature map is mapped to a higher resolution and the pixels are aligned by offsets. The sampling set δ is the sum of the offset O and the original sampling grid G. The sampling set in this case represents the "offset" operation on the original grid, which results in a more accurate position in high resolution space. G is a grid, usually consisting of pixel coordinates or positions. O By learning the offset of each pixel, the position of each pixel can be adjusted so that features can be more accurately localized at higher resolutions. That is, the reshaping operation is omitted. Finally, an upsampled feature map X' of size $C \times sH \times sW$ is generated and produced by the sampling set function, which is shown in (8)-(9) as follows.

$$O = linear(X), \tag{8}$$

$$\delta = G + O. \tag{9}$$

The core function of the point sampling process is to accurately upsample feature maps through offset generation, pixel shuffling, and sampling set adjustment. This enhances the model's ability to capture fine details and improves overall accuracy while maintaining computational efficiency. By increasing the resolution of feature maps, this process boosts the network's performance, particularly in preserving image details and detecting small objects. As a result, the model's perceptual capability is strengthened, enabling more precise localization and identification of true vibration sites.

NWD-Ciou In YOLOv11, the CIoU metric is used to evaluate the overlap between predicted and ground truth bounding boxes. Compared to the traditional IoU (Intersection over Union) metric, CIoU considers not only the overlap area but also the normalized distances between box centers, as well as differences in width and height, making it more robust to variations in bounding box location and size[34][35] .In ship detection, targets vary greatly in size, posing specific challenges. For small ships, the bounding boxes are usually very small, so even minor positional deviations can cause significant fluctuations in IoU values. While the CIoU loss improves IoU stability by incorporating center distance and aspect ratio, it remains highly sensitive to slight bounding box deviations for small targets. This sensitivity can lead to unstable training and ultimately degrade model performance, as the model becomes overly sensitive to the exact position of small targets.Furthermore, CIoU's sensitivity to size changes is especially pronounced for small targets, which may cause the model to overemphasize size adjustments at the expense of other important features like appearance and contextual information.To address these limitations, NWD (Normalized Wasserstein Distance) has been introduced as a novel approach for small target detection[36][37] . Instead of relying on IoU, NWD models each bounding box as a two-dimensional Gaussian distribution and uses the Wasserstein distance to measure the similarity between these distributions. Unlike IoU, Wasserstein distance can effectively measure similarity even when there is little or no overlap. This method is inherently more adaptable to multi-scale objects and is particularly well-suited for measuring similarity among small targets.

In this chapter, NWD is combined with the CIoU method and integrated into the regression loss function to fully utilize the advantages of both in ship detection. CIoU excels in measuring spatial location and overlap, and is particularly suitable for medium- to large-scale ship detection, where the alignment accuracy of the bounding box is critical. On the other hand, NWD exhibits remarkable robustness in dealing with small-scale ships, capturing the distributional features of targets effectively and being insensitive to scale variations. The advantage of NWD lies in its ability to recognize small distributional differences of small targets, details that are often overlooked by conventional methods. Therefore, by combining CIoU's precision in spatial alignment with NWD's sensitivity to distributional features, the robustness and accuracy of the model at different scales are significantly improved. This fusion captures the location and features of pedestrians in a more comprehensive and detailed way, significantly improving the detection performance, especially in scenes with large variations in target size. The formulation of this regression loss function is shown below:

$$L_{reg} = (1 - \gamma)L_{NWD} + \gamma L_{CIoU}, \tag{10}$$

where is the regression loss function, is a weighting factor to balance the contribution of NWD and CIoU in the regression loss. The is usually between 0 and 1. It has been verified that optimal results are obtained when = 0.7. The CIoU loss function is defined as follows:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v, \tag{11}$$

Wherein, IoU is used to assess the degree of overlap between the predicted bounding box and the real bounding box . by calculating the Euclidean distance between the center point of the predicted box and the center point of the real box. denotes the diagonal length of the smallest closed region that can enclose the predicted and true boxes. is the trade-off parameter used to apply the importance of the center distance in the loss function. is a function of the quantized aspect ratio metric.

$$v = \frac{4}{\pi^2}\left(\arctan\frac{wgt}{hgt} - \arctan\frac{w}{h}\right)^2 \tag{12}$$

In(12) , w and h denote the height and width of the predicted frame, respectively, and denote the height w_{gt} and h_{gt} width of the true frame, respectively. The NWD loss function is defined as follows:

$$L_{NWD} = 1 - NWD(N_a, N_b), \tag{13}$$

$$NWD(N_a, N_b) = \exp\left(-\frac{\sqrt{w_2^2(N_a, N_b)}}{C}\right), \tag{14}$$

$$W_2^2(N_a, N_b) = \left\| \left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2}\right]^T, \left[cx_b, cy_b, \frac{w_b}{2}, \frac{h_b}{2}\right]^T \right) \right\|_2^2 \tag{15}$$

c is a constant with strong correlation to the dataset and $W_2^2(N_a N_b)$ is a distance metric. $\mathbb{N}_a$ and $\mathbb{N}_b$ are Gaussian distributions modeled by the bounding boxes $A(cx_a, cy_a, w_a, \hat{h}_a)$ and $B(c_{X_b}, c_{Y_b}, w_b, \hbar_b)$ their centers.

4. Experiments and discussions

This section primarily describes the composition of the dataset used in the experiments, along with relevant details and evaluation metrics. To thoroughly validate the effectiveness of the proposed improved algorithm, a series of comparative experiments have been designed and conducted. To ensure fairness and reproducibility, all experiments are performed without using pre-trained weights, and consistent parameter settings and training strategies are strictly maintained throughout.

4.1. Introduction to the data set

The dataset used in this study is primarily sourced from the InfiRay infrared open platform , which provides real-scene infrared ship image data designed to support the

validation and evaluation of infrared target detection algorithms. The dataset is captured using infrared imaging devices with varying resolutions and focal lengths, covering a wide range of typical ship types. It contains a total of 8,402 infrared images labeled with seven categories of ship targets: liner, bulk carrier, warship, sailboat, canoe, container ship, and fishing boat. The image resolutions vary widely, with some images at 384×288 and 1280×1024 pixels, while the remainder are below 640×640 pixels. During model training, all images are uniformly resized to 640×640 pixels. The dataset is split into training, validation, and test sets at an approximately 8:1:1 ratio, with 6,048 images for training, 1,277 for validation, and 1,077 for testing. The images are mainly captured in coastal ports and active ship areas, where sailboats and canoes constitute a relatively high proportion, and fishing boats are commonly distributed in coastal port zones. The presence of numerous small targets in the dataset adds to the challenge of infrared ship detection and realistically reflects the difficulties encountered in maritime environments. To further validate the generalization ability and robustness of the proposed method, the high-resolution satellite image dataset HRSID[38] is also introduced in this paper. This dataset is widely used in remote sensing image processing and computer vision research, and contains 99 images from Sentinel-1B, 36 images from TerraSAR-X, and 1 image from TanDEM-X, with the image resolution ranging from 1 m to 15 m, and unified in 800×800 image format. A total of 16951 vessels are labeled in the whole dataset, distributed in 5604 sliced SAR images, which provides sufficient samples for the robust training and performance evaluation of the model. In the experiments, the HRSID dataset is used as 70% for training, 20% for testing, and 10% for validation in order to comprehensively measure the detection effect and adaptability of the model in different types of images.

The detailed parameters of the two datasets are presented in Table 1., which outlines the dataset sources, sizes, the number of images in the training, validation, and test sets, as well as the total number of vessels included in each dataset. Figure 7. showcases sample images from both datasets. The InfiRay dataset contains infrared images captured from various angles, featuring different vessel types and diverse backgrounds. In contrast, the HRSID dataset primarily consists of small vessel targets, making it particularly suitable for evaluating the model's performance in small object detection.

Table 1. Parameters of the HRSID dataset and InfiRay dataset

Parameters	HRSID	InfiRay
Data source	Sentinel-1B,TerraSAR-X,TanDem-X	InfiRay
Image size/pixels	800 × 800	384x2881280x1024640x640
Number of training set images	3922	6048
Number of test images	1121	1077
Number of val images	561	1277
Total number of ship target	16951	31307

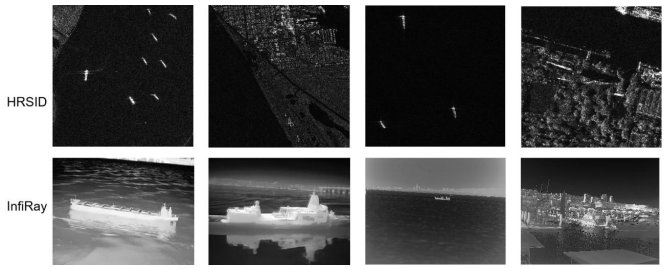


Fig. 7. Example of HRSID training set and InfiRay training set

4.2. Experimental environment

The experiments in this study were conducted on a computer running Windows 10. The hardware configuration includes a 12th-generation Intel® Core™ i7-4214R processor with a base clock speed of 2.40 GHz, paired with an NVIDIA GeForce RTX 3080 Ti graphics card. The software environment was built using Python 3.8 and PyTorch 2.2.1. Detailed specifications of the experimental platform are provided in Table2.

Table 2. Configuration information for the experimental platform

Configuration	Versions
Operating system	Windows 10
CPU	12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz
GPU	NVIDIA GeForce RTX 3080Ti
RAM	32GB
Toolkit	CUDA 12.1
Compiler	Python 3.8
Framework	PyTorch 2.2.1

The experimental parameters for training the model are shown in Table3. The experiments were conducted using an input image size of 640 × 640 pixels over 150 training cycles. The initial and final learning rates were set at 0.01.

Table 3. Experimental parameters of the training model

Name	Configuration
Momentum	0.937
Data enhancement	Mosaic
Epochs	150
Batch size	16
Workers	16
Image size	640×640
Initial learning rate	0.01
Final learning rate	0.01

4.3. Performance evaluation

In this study, to comprehensively evaluate the model's performance, we employ the COCO evaluation metrics, including Precision (P), Recall (R), mean Average Precision (mAP), GFLOPS, and the number of parameters.Precision (P) represents the probability that a detected target is correctly identified and is defined as follows.

$$P = \frac{TP}{TP + FP} \times 100\% \tag{16}$$

In the target detection task, TP (True Positive) denotes the number of correctly identified positive samples, while FP (False Positive) refers to the number of incorrectly detected negative samples.Recall (R) represents the probability of correctly identifying positive samples among all actual positive samples and is defined as follows:

$$R = \frac{TP}{TP + FN} \times 100\% \tag{17}$$

Here, TP (True Positive) refers to the number of correctly identified positive samples, while FN (False Negative) represents the number of missed positive samples. Average Precision (AP) indicates the average detection accuracy for a single category and is defined as follows.

$$AP = \int_0^1 P \, dR \tag{18}$$

mAP (mean Average Precision) is used as the core index to evaluate the performance of the target detection model, and its higher value indicates the better detection accuracy of the model, which is defined as follows.

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N}. \tag{19}$$

GFLOPS (Giga Floating-point Operations Per Second) quantifies the computational capacity of the model, representing the number of billions of floating-point operations the model can perform per second. Meanwhile, model parameters—comprising the network's weights and biases—directly impact the model's computational complexity.

4.4. Ablation experiments

The proposed ADN-YOLO model demonstrates enhanced detection performance across all categories in the infrared ship dataset (infiRay). It significantly improves the detection of small targets in multi-scale scenarios and performs robustly under conditions of dense occlusion. This study introduces three innovative modules built upon the YOLOv11 framework: ADown, DySample, and NWD-CIoU. As shown in Table 4.,Table 5.. eight ablation experiments were conducted to rigorously assess the contribution of each module to the overall detection accuracy. The ADown module combines average pooling and max pooling to enable multi-scale feature extraction in infrared ship images. This approach led to a 0.9% increase in Recall, a 0.5% improvement in mAP@0.5, a reduction of 0.46M in parameters, and a decrease of 1.2 GFLOPs. DySample replaces traditional

dynamic convolution with a point-based sampling technique, yielding a 0.7% increase in mAP@0.5 and a 0.8% improvement in Recall. The NWD-CIoU module enhances the model's sensitivity to subtle distributional differences in small targets by integrating Normalized Wasserstein Distance (NWD) with Complete IoU (CIoU). This integration resulted in a 1.2% gain in Recall and a 0.8% increase in mAP@0.5.Collectively, these enhancements significantly boost the model's performance across all key evaluation metrics.

Table 4. Comparison tests on the InfiRay dataset

ADown	Dysample	NWD-CIoU	Precision (%)	Recall (%)	mAP@0.5 (%)
			90.1	83.1	88.7
✓			89.7	84.0	89.2
	✓		90.0	83.9	89.4
		✓	89.5	84.3	89.5
✓	✓		90.8	84.3	89.8
✓		✓	90.4	85.1	90.0
	✓	✓	90.7	84.4	89.9
✓	✓	✓	91.1	85.0	90.6

Table 5. Comparison tests on the InfiRay dataset

ADown	Dysample	NWD-CIoU	mAP@0.5:0.95 (%)	Params (M)	FLOPs (G)
			62.1	2.56	6.3
✓			61.8	2.10	5.1
	✓		62.4	2.59	6.3
		✓	62.0	2.56	6.3
✓	✓		62.5	2.14	5.2
✓		✓	61.9	2.10	5.3
	✓	✓	62.3	2.62	6.5
✓	✓	✓	62.8	2.14	5.2

PR curves are generated by computing precision and recall at various threshold levels. The horizontal axis represents recall, indicating the proportion of actual positive cases that the model correctly identifies. The vertical axis represents precision, indicating the proportion of predicted positive cases that are truly positive. PR curves provide a comprehensive view of model performance across different thresholds. Typically, curves that are closer to the upper right corner reflect a model's ability to achieve both high precision and high recall. A larger area under the curve (AUC) signifies better overall performance, with the blue curve representing the average performance across all categories.Figure 8 illustrates the PR curves for both ADN-YOLO and YOLOv11. In this figure, the proximity of a curve to the upper right corner indicates superior detection performance. Figure 8.(a) and Figure 8.(b) show the mAP@0.5 values for various categories in the dataset at the lower left. On the infrared ship dataset, the results in Figure 8.(b) demonstrate notable improvements across all categories compared to Figure 8.(a), with particularly significant

gains for the 'liner' and 'fishing boat' categories. This highlights the enhanced ability of ADN-YOLO to capture target features in complex maritime environments.Overall, the PR curves confirm that ADN-YOLO achieves a well-balanced trade-off between precision and recall.

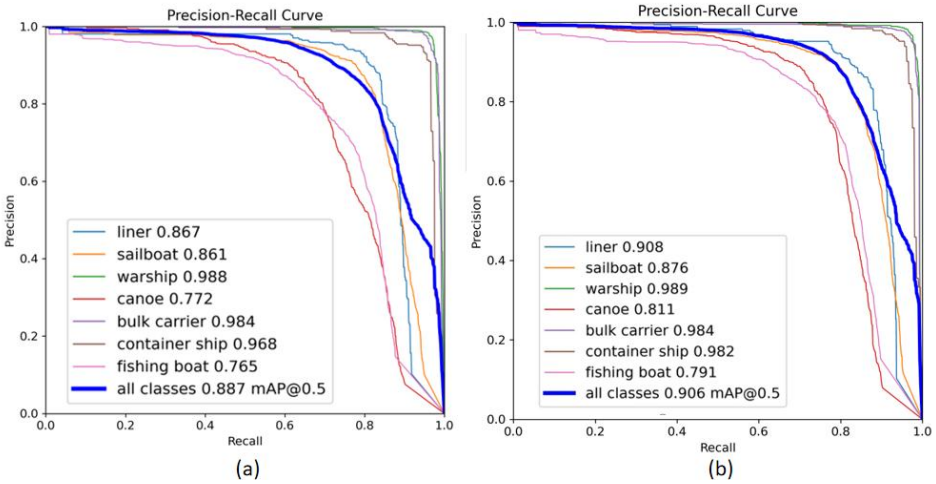


Fig. 8. Comparison of PR curves for the test set models: the PR curves of (a) YOLOv11 and (b) ADN-YOLOv11

A confusion matrix is a visualization tool that presents the performance of a classification algorithm in a matrix format, offering a clear view of how the model performs across different categories. Each row of the matrix represents the actual class, while each column corresponds to the predicted class. In this context, the horizontal axis denotes the true labels, and the vertical axis denotes the predicted precision. Each horizontal coordinate of the matrix represents a category in the dataset, including seven types of ship targets: liner, bulk carrier, warship, sailboat, canoe, container ship, and fishing boat.The elements along the main diagonal represent the number of correctly classified samples. The lower left triangular region indicates missed detections (false negatives), whereas the upper right triangular region reflects false detections (false positives).When values are high in the lower-left region, it suggests that the model has missed many detections. Conversely, high values in the upper-right region indicate a greater number of false detections. As shown in Figure 9., the confusion matrix of ADN-YOLO (after improvements to YOLOv11) reveals a notable enhancement in detection accuracy. Compared to Figure 9.(a), the reduced values in the upper-right region of Figure 9.(b) indicate a significant decrease in false detections. Similarly, the lower values in the lower-left region of Figure 9.(b) demonstrate a reduction in missed detections.These results confirm that the improved model is more effective in detecting targets within complex maritime environments, with marked improvements in both false and missed detections.

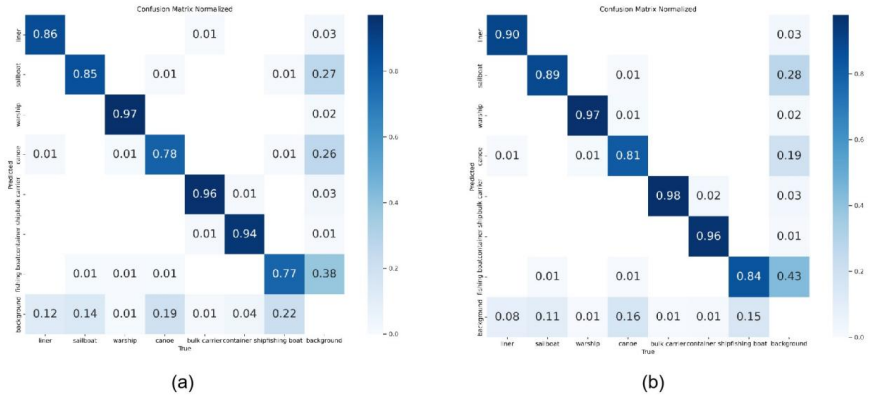


Fig. 9. Comparison of the confusion matrices of the test set models: This figure presents a visual comparison between the confusion matrices of two models evaluated on the test set: (a) the baseline YOLOv11 and (b) the improved ADN-YOLOv11. The comparison highlights differences in classification performance, including true positives, false positives, and false negatives across different categories

Figure 10. presents the visual results of ship detection using HiResCAM for both YOLOv11 and ADN-YOLO. HiResCAM is a model interpretability technique that visualizes attention by capturing gradient information to generate heatmaps. Compared to Grad-CAM—which tends to spread attention and may be better suited for segmentation tasks—HiResCAM offers a more faithful and class-specific interpretation of model decisions [39]. This method provides an intuitive means of visualizing how and where the model focuses during inference, helping to explain its decision-making process. In Figure 10.(a), the detection region of the ship in the original image is annotated to clarify the ship's position. This baseline aids in comparing the attention maps produced by the two models. As seen in Figure 10.(c), ADN-YOLO generates a heatmap with concentrated, darker regions, indicating a strong and focused response to the actual target. This suggests that the model effectively concentrates on critical features related to the ship, improving its precision in infrared ship detection.In contrast, Figure 10.(b) shows that YOLOv11 produces a more dispersed heatmap, with attention distributed over a broader area, including potentially irrelevant background regions. This contrast highlights the superior feature extraction and localization capabilities of ADN-YOLO, particularly in detecting occluded or small ship targets.Overall, compared to the original algorithm, ADN-YOLO demonstrates enhanced focus on ship features and better suppression of background noise. The highlighted areas in the heatmaps more closely align with the true locations of ships in the original image, underscoring the model's improved accuracy, robustness, and interpretability in complex maritime environments.

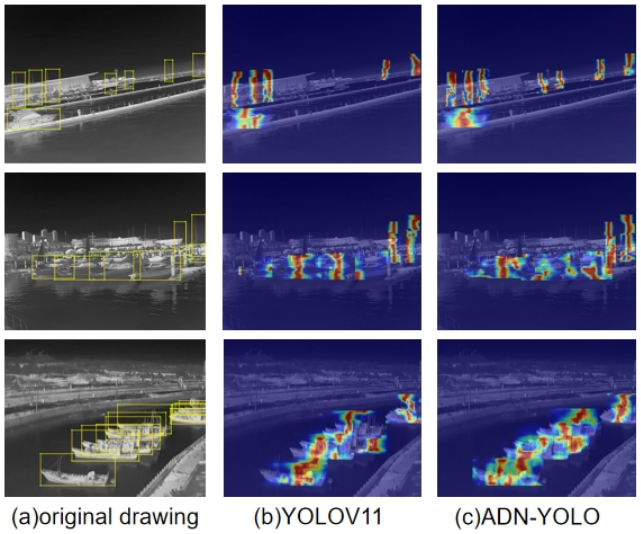


Fig. 10. Visualization of the infiRay dataset.(a) shows the original annotation boxes of the ships, (b) represents the heatmap of YOLOv11, and (c) represents the heatmap of ADN-YOLO

In complex maritime environments, the wide variation in ship scales, combined with the challenges of infrared imaging,such as low-light conditions and target occlusion, makes accurate detection particularly difficult [40,41]. Small targets, in particular, often exhibit weak texture features, and the limited feature extraction capability of YOLOv11 may result in issues such as false detections or missed targets. As illustrated in Figure 11.(a) comparison of the prediction results between YOLOv11 and ADN-YOLO reveals that small targets like canoes are not detected by YOLOv11 in Figure 11.(a), largely due to the reduced visibility inherent in infrared imaging. In densely occluded scenes, or when the target appears at a small scale, YOLOv11 is prone to interference, which negatively impacts detection and localization accuracy. In contrast, Figure 11.(b) demonstrates the results obtained using the proposed ADN-YOLO model. Even under low-visibility conditions, ADN-YOLO is capable of accurately detecting small targets such as canoes. Compared to YOLOv11, ADN-YOLO exhibits significantly enhanced performance, particularly in scenes with heavy occlusion. It achieves superior results in terms of detection accuracy, robustness, and reliability. These improvements make ADN-YOLO better suited for real-world applications in ship detection, especially in challenging infrared imaging scenarios.

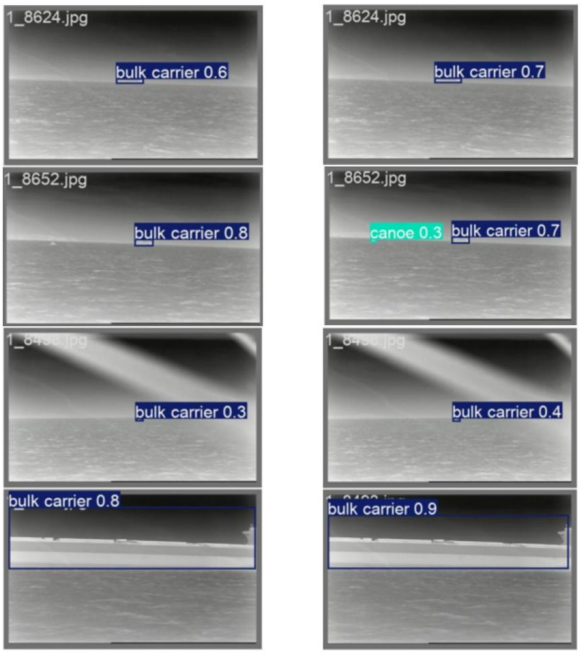


Fig. 11. Comparison of infrared ship detection.(a) shows the ship detection results of YOLO11, and (b) shows the ship detection results of ADN-YOLO

4.5. Comparisons with other methods

In this section, to qualitatively evaluate the detection performance of the proposed ADN-YOLO model on the InfiRay dataset, we compare it against several mainstream YOLO models, including YOLOv3-tiny[42] , YOLOv5 [43], YOLOv6[44] , YOLOv8[45] , YOLOv9-tiny[46] , YOLOv10[47] , YOLOv11[48] , and YOLOv12[49] , using the test set for ship image detection. The comparison results are presented in Table 6.Among the models, YOLOv8 achieves the highest average accuracy at 89.9%. However, its large number of parameters and high FLOPs make it less suitable for real-time applications. In contrast, the proposed ADN-YOLO significantly outperforms the baseline YOLOv11 model, achieving improvements of 1.0%, 1.9%, and 1.9% in precision, recall, and mAP@0.5, respectively reaching 91.1%, 85.0%, and 90.6%.Furthermore, ADN-YOLO reduces the number of parameters by 0.4 million compared to YOLOv11, and by 0.9 million when compared to YOLOv8, demonstrating its superior efficiency. These results confirm that ADN-YOLO is more suitable for real-time ship detection in infrared images, offering a favorable balance between detection accuracy and computational efficiency. It achieves strong detection performance while maintaining a lightweight model architecture, making it highly effective for practical deployment in complex maritime scenarios.

Table 6. Comparison tests on the InfiRay dataset

Model	Precision/%	Recall/%	Map@0.5/%	map@0.5:0.95/%	Params/M	FLOPs/G
Yolov3-tiny	89.8	79.1	85.2	59.4	12.1	18.9
Yolov5	88.8	82.7	88.9	61.3	2.5	7.1
Yolov6	89.8	82.2	88.2	61.5	4.2	11.8
Yolov8	90.1	85.3	89.9	62.5	3.0	8.1
Yolov9-tiny	90.6	84.3	89.8	62.7	1.9	7.6
Yolov10	86.6	81.1	88.8	61.2	2.7	8.2
Yolov11	90.1	83.1	88.7	62.1	2.6	6.3
Yolov12	90.3	84.4	89.0	61.4	2.55	6.3
ADN-YOLO	91.1	85	90.6	62.8	2.1	5.2

presents the evaluation results of the proposed model in terms of Figure 12mAP@0.5 and Recall after 150 training epochs. Due to challenges such as low resolution, insufficient illumination, and noise interference inherent in infrared imagery, vessel targets are often missed during detection. As shown in Figure 12(b), the proposed ADN-YOLO model significantly outperforms the comparative models in Recall, demonstrating a clear advantage in reducing missed detections. This indicates the model's effectiveness in mitigating the adverse effects of complex backgrounds, poor lighting conditions, and other interference factors common in infrared images.The mAP@0.5 metric is a critical indicator of object detection performance—higher values signify better model capability in distinguishing targets from background noise and reducing both missed and false detections. As illustrated in Figure 12.(a), the mAP@0.5 curve of the proposed model is consistently higher than those of other models, underscoring its superior detection accuracy. This suggests that ADN-YOLO is more effective in recognizing vessel targets within infrared imagery and is capable of detecting a greater number of valid targets.In summary, ADN-YOLO demonstrates higher precision and recall in infrared ship detection tasks, reflecting its robustness and overall performance advantages in complex maritime environments.

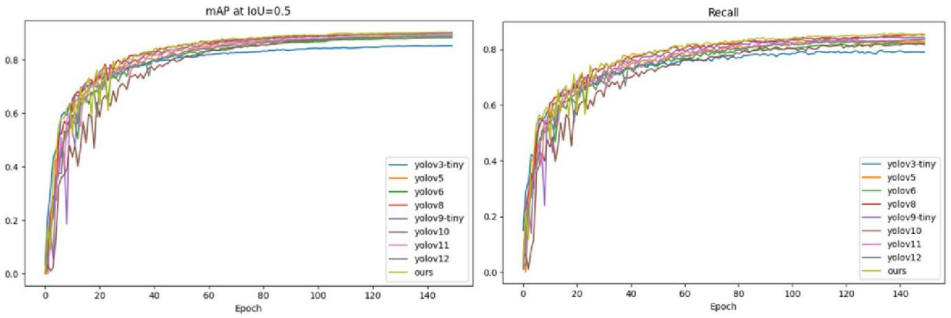


Fig. 12. mAP for each model in the InfiRay dataset and Recall.(a) shows the curves of each model with respect to the mAP@0.5 metric, and (b) shows the curves of each model in terms of Recall

In this study, comparative experiments were conducted on the HRSID dataset to further validate the stability and generalization ability of the proposed model. The performance of ADN-YOLO was compared with several mainstream detection algorithms, including YOLOv3-tiny [42], YOLOv5 [43], YOLOv6 [44], YOLOv8 [45], YOLOv9-tiny [46], YOLOv10 [47], YOLOv11 [48], and the latest YOLOv12 [49]. The comparison focuses on the models' performance in ship image detection on the validation set. The specific experimental results are presented in Table 7.As shown in the table, YOLOv9-tiny has the smallest number of parameters. While it offers certain advantages in terms of lightweight design, its mAP@0.5 and Recall scores are 0.7% and 2.0% lower, respectively, than those of the proposed ADN-YOLO, indicating some limitations in detection performance. Compared with the baseline model YOLOv11, ADN-YOLO achieves improvements of 1.9%, 2.1%, and 2.9% in Precision, Recall, and mAP@0.5:0.95, respectively, significantly enhancing overall detection capability.This performance improvement is particularly meaningful in practical applications especially in complex maritime scenarios where small vessels are more prone to being missed or misidentified. Such advancements contribute to enhanced system robustness and safety. In summary, ADN-YOLO not only demonstrates superior detection performance for ship targets in noisy, low-light infrared images but also shows strong adaptability in small-object detection tasks, highlighting its broad application potential.

Table 7. Comparison tests for the HRSID dataset

Model	Precision/%	Recall/%	Map@0.5/%	map@0.50.95/%	Params/M	FLOPs/G
Yolov3-tiny	90.4	70.4	81.6	58.2	12.1	18.9
Yolov5	89.2	82.3	88.3	63.1	2.50	7.1
Yolov6	88.1	81.0	86.7	60.7	4.23	11.8
Yolov8	89.8	83.6	88.5	66.5	3.01	8.1
Yolov9-tiny	92.3	81.5	90.8	66.4	1.97	7.6
Yolov10	91.6	80.6	90.3	66.0	2.69	8.2
Yolov11	89.7	81.4	91.0	63.3	2.58	6.3
Yolov12	89.4	79.9	88.4	63.1	2.56	6.3
ADN-YOLO	91.6	83.5	91.5	66.2	2.25	5.2

Figure 13 presents various evaluation metrics after 150 training epochs on the HRSID dataset. The vertical position of each circle represents the mAP@0.5 higher value placement indicates better detection performance, especially for small vessel targets at sea. The size of each circle intuitively reflects the number of model parameters, which corresponds to the model's complexity and learning capacity.From the figure, it is evident that our model achieves superior performance in terms of parameter efficiency compared to other models. Horizontally, the position of each circle indicates the GFLOPs, with a greater distance to the right representing a heavier computational load. As seen in the figure, YOLOv3-tiny lies farthest to the right, indicating a higher computational burden. In contrast, our model is positioned closest to the left edge of the horizontal axis, reflecting its low computational cost and faster inference speed—making it well suited for real-time detection of small vessel targets.In summary, the proposed model not only offers faster inference speed than competing models but also achieves a higher mAP@0.5 score. It ef-

fectively detects small vessel targets while preserving detailed features, even under rapid inference conditions. Additionally, it significantly reduces both false positives and missed detections, highlighting its practical advantage in real-world applications.

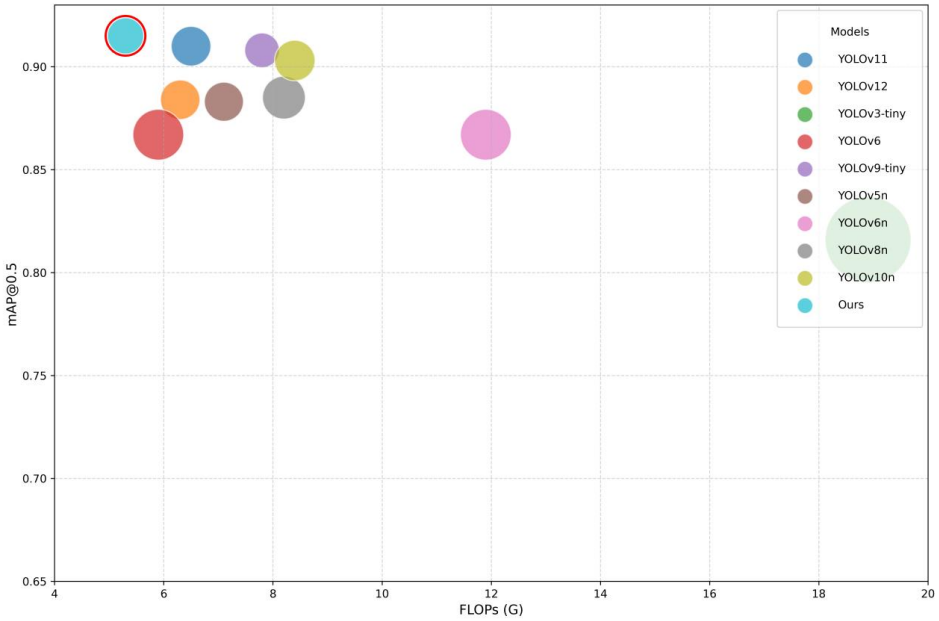


Fig. 13. Bubble plots for each model in the HRSID dataset

In complex marine environments, noise interference is common due to the structural similarities among vessels. Detecting small targets often encounters challenges such as overlapping objects and weak responses on heat maps. To better illustrate these issues, we visualized detection results on the HRSID dataset[50][51][52], enabling direct observation of missed and misdetected vessels. To compare our improved model with the baseline, we selected three representative images from the HRSID dataset depicting complex sea scenes, as shown in Figure 14 These images contain numerous small vessel targets. In our visualization, different colored bounding boxes are used to distinguish detection outcomes: yellow boxes indicate ground-truth annotations from the original dataset, blue boxes represent correctly identified vessels by the model, orange boxes mark vessels missed by the model, and purple boxes denote falsely detected vessels. This color scheme facilitates a clearer understanding and evaluation of the model's accuracy and errors. As shown in Figure 14.(a), small vessels are densely clustered, making detection heavily influenced by the surrounding environment. In Figure 14.(b), the baseline model exhibits both false positives and missed detections, with environmental interference exacerbating these errors. In contrast, Figure 14.(c) displays results from our improved model, demonstrating significant reductions in missed detections of small vessels and accurate recognition even in complex scenarios.

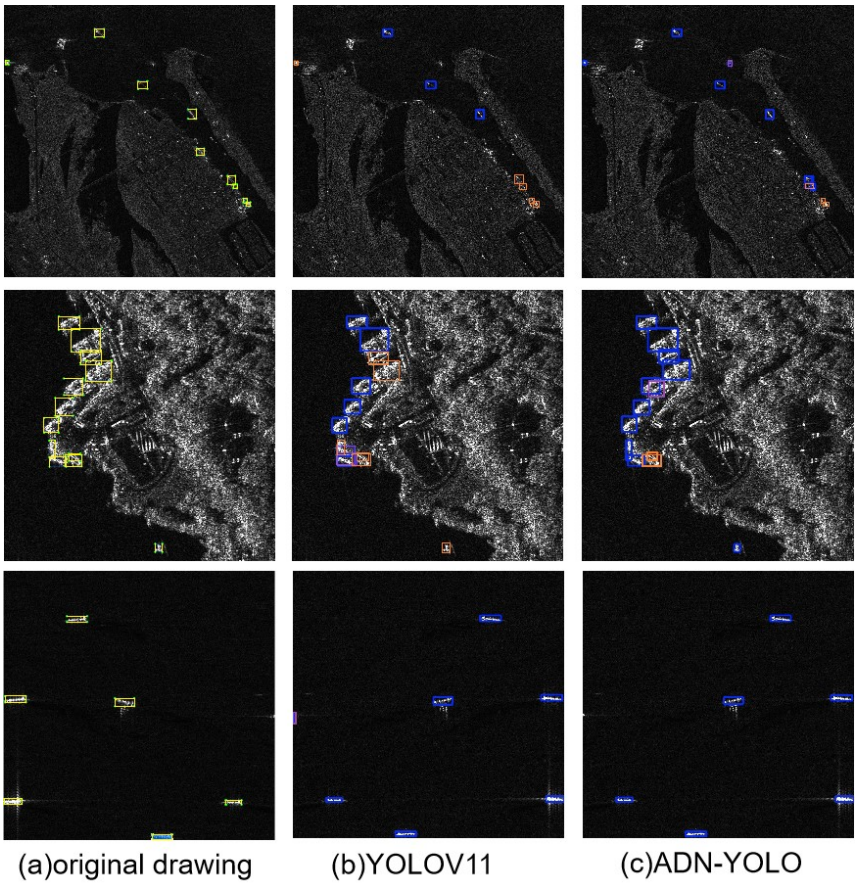


Fig. 14. Heatmaps on the HRSID dataset (a) Vessel distribution in the original image, with yellow boxes representing the ground-truth annotations (b) Detection results from YOLOv11: blue boxes indicate correctly detected vessels, orange boxes indicate missed detections, and purple boxes indicate false detections. (c) Detection results from ADN-YOLO: blue boxes denote correctly detected vessels, and orange boxes denote missed detections

5. Conclusion

In this study, we propose an improved infrared ship detection model, ADN-YOLO, designed to address the challenges inherent to infrared scenes, such as the wide distribution of small target ships, ambiguous features, and susceptibility to background interference. The model is structurally optimized based on the YOLOv11 framework and incorporates dynamic sampling, attention mechanisms, and a novel loss function to comprehensively enhance detection accuracy and robustness for infrared small targets.Specifically, ADN-YOLO introduces a lightweight and efficient ADown downsampling module into

the backbone network, replacing the traditional CBS structure. This module enhances the extraction of salient features for small targets while reducing the model's parameter count and computational complexity. Additionally, the Dysample dynamic upsampling module reconstructs the upsampling process from a point-sampling perspective, effectively preserving shallow features and spatial location information. This further improves the model's accuracy and robustness in detecting small infrared targets, especially under complex backgrounds and weak target conditions.Moreover, to tackle the instability of bounding box regression using the traditional IoU loss function in small target detection, this study proposes the NWD-CIoU metric. By modeling the predicted and ground-truth boxes as two-dimensional Gaussian distributions and employing the normalized Wasserstein distance to measure their differences, NWD-CIoU significantly enhances the model's generalization across targets of varying scales and shapes.Comprehensive experimental results demonstrate that ADN-YOLO achieves superior detection accuracy and recall in infrared image target detection, particularly for small targets, validating the effectiveness and practical value of the proposed method.

However, the method proposed in this paper still has certain limitations in practical applications. While the model has achieved significant improvements in detection accuracy, there remains room for optimization in inference efficiency, as the current running speed and resource consumption do not fully meet the demands of extreme real-time scenarios. Additionally, the model has yet to be thoroughly validated across a broader range of infrared detection tasks, and its adaptability to complex environments and cross-scene generalization requires further investigation. In future work, we plan to extend the model to other infrared target detection applications such as night surveillance and border defense warning to further verify its generality and stability.

Acknowledgments. This work was Supported by Key Lab of Information Network Security, Ministry of Public Security,the National Natural Science Foundation of China under Grants 61672338, the Natural Science Foundation of Shanghai under Grant 21ZR1426500.

References

1. Shuhong Wang, Weiyao Li, and Lu Xing. A review on marine economics and management: How to exploit the ocean well. *Water*, 14(17):2626, 2022.
2. Mingming Cui, Dezhi Han, Han Liu, Kuan-Ching Li, Mingdong Tang, Chin-Chen Chang, Ferheen Ayaz, Zhengguo Sheng, and Yong Liang Guan. Secure data sharing for consortium blockchainenabled vehicular social networks. *IEEE Transactions on Vehicular Technology*, 2024.
3. Liyuan Li, Linyi Jiang, Jingwen Zhang, Siqi Wang, and Fansheng Chen. A complete yolo-based ship detection method for thermal infrared remote sensing images under complex backgrounds. *Remote Sensing*, 14(7):1534, 2022.
4. Chongqing Chen, Dezhi Han, and Xiang Shen. Clvin: Complete language-vision interaction network for visual question answering. *Knowledge-Based Systems*, 275:110706, 2023.
5. Renke Kou, Chunping Wang, Zhenming Peng, Zhihe Zhao, Yaohong Chen, Jinhui Han, Fuyu Huang, Ying Yu, and Qiang Fu. Infrared small target segmentation networks: A survey. *Pattern recognition*, 143:109788, 2023.
6. Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2007.

7. Chongqing Chen, Dezhi Han, and Xiang Shen. Clvin: Complete language-vision interaction network for visual question answering. *Knowledge-Based Systems*, 275:110706, 2023.
8. Chongqing Chen, Dezhi Han, Zihan Guo, and Chin-Chen Chang. Towards bias-aware visual question answering: Rectifying and mitigating comprehension biases. *Expert Systems with Applications*, 264:125817, 2025.
9. Landan Zhang and Zhenming Peng. Infrared small target detection based on partial sum of the tensor nuclear norm. *Remote Sensing*, 11(4):382, 2019.
10. Dezhi Han, Nannan Pan, and Kuan-Ching Li. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. *IEEE Transactions on Dependable and Secure Computing*, 19(1):316–327, 2020.
11. GU Jiaojiao, LI Bingzhen, LIU Ke, and JIANG Wenzhi. Infrared ship target detection algorithm based on improved faster r-cnn. *Infrared Technology*, 43(2):170–178, 2021.
12. Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 821–830, 2019.
13. Yang Zheng, Izzat H Izzat, and Shahrzad Ziaee. Gfd-ssd: Gated fusion double ssd for multispectral pedestrian detection. *arXiv preprint arXiv:1903.06999*, 2019.
14. Shasha Li, Yongjun Li, Yao Li, Mengjun Li, and Xiaorong Xu. Yolo-firi: Improved yolov5 for infrared image object detection. *IEEE access*, 9:141861–141875, 2021.
15. Xiao Luo, Hao Zhu, and Zhenli Zhang. Ir-yolo: Real-time infrared vehicle and pedestrian detection. *Computers, Materials & Continua*, 78(2), 2024.
16. Jihao Si, Binbin Song, Jixuan Wu, Wei Lin, Wei Huang, and Shengyong Chen. Maritime ship detection method for satellite images based on multiscale feature fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:6642–6655, 2023.
17. Bowen Xing, Wei Wang, Jingyi Qian, Chengwu Pan, and Qibo Le. A lightweight model for real-time monitoring of ships. *Electronics*, 12(18):3804, 2023.
18. Nan Wang, Bo Li, Xingxing Wei, Yonghua Wang, and Huanqian Yan. Ship detection in spaceborne infrared image based on lightweight cnn and multisource feature cascade decision. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4324–4339, 2020.
19. Jiatao Li, Dezhi Han, Tien-Hsiung Weng, Huafeng Wu, Kuan-Ching Li, and Arcangelo Castiglione. A secure data storage and sharing scheme for port supply chain based on blockchain and dynamic searchable encryption. *Computer Standards & Interfaces*, 91:103887, 2025.
20. K Zhao and X Kong. Background noise suppression in small targets infrared images and its method discussion. *Optics and Optoelectronic Technology*, 2(2):9–12, 2004.
21. TS Anju and NR Nelwin Raj. Shearlet transform based image denoising using histogram thresholding. In *2016 International Conference on Communication Systems and Networks (ComNet)*, pages 162–166. IEEE, 2016.
22. P Jiao. Research on image classification and retrieval method based on deep learning and sparse representation. *Xi'an University of Technology: Xi'an, China*, 2019.
23. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
24. Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
25. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
26. Debasmita Ghose, Shasvat M Desai, Sneha Bhattacharya, Deep Chakraborty, Madalina Fiterau, and Tauhidur Rahman. Pedestrian detection in thermal images using saliency maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

27. Chaitanya Devaguptapu, Ninad Akolekar, Manuj M Sharma, and Vineeth N Balasubramanian. Borrow from anywhere: Pseudo multi-modal object detection in thermal imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
28. Chongqing Chen, Dezhi Han, and Chin-Chen Chang. Caan: Context-aware attention network for visual question answering. *Pattern Recognition*, 132:108980, 2022.
29. Yangyang Li, Qin Huang, Xuan Pei, Yanqiao Chen, Licheng Jiao, and Ronghua Shang. Cross-layer attention network for small object detection in remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:2148–2161, 2020.
30. Hao Guo and Dongbing Gu. Closely arranged inshore ship detection using a bi-directional attention feature pyramid network. *International Journal of Remote Sensing*, 44(22):7106–7125, 2023.
31. Chongqing Chen, Dezhi Han, and Chin-Chen Chang. Mpcct: Multimodal vision-language learning paradigm with context-based compact transformer. *Pattern recognition*, 147:110084, 2024.
32. Xiang Shen, Dezhi Han, Chin-Chen Chang, Ammar Oad, and Huafeng Wu. Gfsnet: Gaussian fourier with sparse attention network for visual question answering. *Artificial Intelligence Review*, 58(6):1–30, 2025.
33. Wenji Yang and Xiaoying Qiu. A lightweight and efficient model for grape bunch detection and biophysical anomaly assessment in complex environments based on yolov8s. *Frontiers in Plant Science*, 15:1395796, 2024.
34. Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.
35. Yufei Zhao, Yong Liang Guan, Dong Chen, Afkar Mohamed Ismail, Xiaoyan Ma, Xiaobei Liu, and Chau Yuen. Exploring rcs diversity with novel oam beams without energy void: An experimental study. *IEEE Transactions on Vehicular Technology*, 2024.
36. Liliang Zhou, Huaming Ran, Rongling Xiong, and Ruijie Tan. Nwd-yolov5: A yolov5 model for small target detection based on nwd loss. In *2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, pages 542–546. IEEE, 2024.
37. Yufei Zhao, Ziyang Wang, Yilong Lu, and Yong Liang Guan. Multimode oam convergent transmission with co-divergent angle tailored by airy wavefront. *IEEE Transactions on Antennas and Propagation*, 71(6):5256–5265, 2023.
38. Xiao Tang, Jiufeng Zhang, Yunzhi Xia, and Huanlin Xiao. Dbw-yolo: A high-precision sar ship detection method for complex environments. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.
39. Tobias Clement, Truong Thanh Hung Nguyen, Mohamed Abdelaal, and Hung Cao. Xai-enhanced semantic segmentation models for visual quality inspection. In *2024 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4. IEEE, 2024.
40. Dun Li, Dezhi Han, Noel Crespi, Roberto Minerva, Syed Mohsan Raza, Reza Farahbakhsh, Wei Liang, and Zibin Zheng. Blockchain in the digital twin context: A comprehensive survey. *ACM Computing Surveys*, 2025.
41. Rui Jiang, Hang Shi, Jiahong Ni, Jiatao Li, Yi Feng, Xinqiang Chen, and Yinlin Li. Lsdformer: Lightweight sar ship detection enhanced with efficient multi-attention and structural reparameterization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–20, 2025.
42. Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
43. Bharat Mahaur and KK Mishra. Small-object detection based on yolov5 in autonomous driving systems. *Pattern Recognition Letters*, 168:115–122, 2023.

44. Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.
45. Muhammad Hussain. Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo. *IEEE access*, 12:42816–42833, 2024.
46. Jialin Zou and Hongcheng Wang. Steel surface defect detection method based on improved yolov9 network. *IEEE Access*, 2024.
47. Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011, 2024.
48. Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.
49. Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.
50. Jiatao Li, Dezhi Han, Shuxin Shi, Xiaoqi Xin, Kuan-Ching Li, and Chin-Chen Chang. An active client selection scheme based on blockchain for federated learning in shipping. *IEEE Trans. Intell. Transp. Syst.*, 26(11):20669–20684, Nov. 2025.
51. Rui Jiang, Ruixiang Zhu, Hu Su, Yinlin Li, Yuan Xie, and Wei Zou. Deep learning-based moving object segmentation: Recent progress and research prospects. *Int. J. Automat. Comput.*, 20(3):335–369, Jun 2023.
52. Jiatao Li, Dezhi Han, Tien-Hsiung Weng, Huafeng Wu, Kuan-Ching Li, and Arcangelo Castiglione. A secure data storage and sharing scheme for port supply chain based on blockchain and dynamic searchable encryption. *Comput. Stand. Interfaces*, 91, Jan. 2025. Art. no. 103887.

Tao Li is currently pursuing a master's degree in the School of Information Engineering at the China University of Shipping in Pudong. His current research interests are in network security.

Dezhi Han received his Bachelor of Science in applied physics from Hefei University of Technology in 990, and his Master of Science and Doctor of Philosophy in computer science from Wuhan University of Science and Technology in China in 2001 and 205, respectively. He is currently a professor in the Department of Computer Science at the China University of Shipping in Shanghai Pudong, where he has been employed since 200. His current research interests include cloud computing and security, blockchain, wireless communication security, network, and information security.

Songyang Wu is a researcher and director of the Security Center at the Third Research Institute of Ministry of Public Security, and also serves as the deputy director of the National Engineering Research Center for Network Security Grade and Security Technology and the deputy director of the Key Laboratory of Information Network Security of Ministry of Public Security. He received his Bachelor of Science in computer science and technology from Tongji University in 200 and his Doctor of Philosophy in computer application from Tongji University in 2011. The same year, he joined the Ministry of Public Security Center. He received his Doctor Philosophy in computer application from Tongji University in 2011. The same year, he joined the Network Security Center at the Third Research Institute of Ministry of Public Security. research interests include cybercrime investigation, electronic data forensics, big data security, and artificial intelligence security.

Xiang Shen received his Master of Science in computer science technology from the Shanghai Maritime University in Shanghai Pudong, China, in 2022 and is currently pursuing a doctoral degree in the School of Information Engineering. current research interests are in network security.

Liqi Zhu received his Master of Science in computer science and technology from the Shanghai Maritime University in Shanghai Pudong,, in 2025 and is currently pursuing a doctoral degree in the School of Information Engineering. His current research interests are in blockchain federated learning.

Wenqi Sun is an associate researcher and research engineer at the Network Security Center at the Third Research Institute of Ministry of Public Security; she received her computer science and technology degree from Northeast University 2010 and her doctorate in computer science and technology from Tsinghua University in 2016. In 2018, she joined the Network Center at the Third Research Institute of Ministry of Public Security; her current research interests are in network investigations.

Received: June 13, 2025; Accepted: December 14, 2025.

Computer Science and Information Systems 23(1):165–184 https://doi.org/10.2298/CSIS250710074H

A method for representing stock time series features based on trend and inclination angle turning points

Lei Han and Xuedong Gao

School of Economics and Management, University of Science and Technology Beijing
100083 Beijing,China
D202210486@xs.ustb.edu.cn
gaoxuedong@manage.ustb.edu.cn

Abstract. Stock time series data mining faces significant challenges in terms of time and space. Time series feature representation is an important means of reducing the dimensionality of time series. This study proposes a stock time series feature representation method based on a combination of slope angle changes and trend turning point screening. This method alleviates the limitations of single methods, such as local feature overfitting, loss of global trends, and sparse long-segment features, through a multi-level collaborative mechanism of screening, filtering, and supplementation. Experimental results show that compared with four other methods, the proposed method effectively preserves the original features of stock time series, achieving good results in both similarity metrics and fitting errors.

Keywords: stock time series, feature representation, inclination angle turning point, trend analysis.

1. Introduction

With the emergence of new concepts and technologies such as cloud computing, big data, the Internet of Things, and mobile cloud computing, the rate of data growth is rapidly increasing [7]. At the same time, the rapid advancement of computing power and massive storage technology has enabled the systematic acquisition and storage of the complex and heterogeneous data generated by human society in its daily operations and scientific exploration. In this context, time series data, which is organized strictly according to time sequence, has become an important research subject in the field of data analysis because it can describe the dynamic evolution of the state or phenomenon of an object. Efficient processing and in-depth mining of time series data are key means of revealing its internal patterns, correlation laws, and predicting future trends, and have significant practical value. The stock market is a typical application scenario for time series data. Compared with traditional economic data, stock prices exhibit significant volatility and high complexity, which poses unique challenges for the analysis and prediction of stock time series. How to effectively extract useful information from stock time series data and provide scientific guidance to investors has become a hot topic in scientific research.

Due to the inherent high-dimensional characteristics and strong sequence dependency of stock time series, it is very difficult to perform data mining or similarity measurement directly on the original time series. First, at the storage level, uncompressed high-dimensional data occupies a large amount of physical space, resulting in significant storage costs. Second, at the computing level, the computational complexity of traditional

algorithms in high-dimensional spaces often increases exponentially, leading to a sharp decline in processing efficiency. More importantly, raw data usually contains observational noise and measurement errors, and direct processing can easily cause the model to overfit noise patterns, damaging the robustness and generalization ability of the results. The benefits of time series feature representation in improving the efficiency and effectiveness of time series mining have received widespread attention. For example, in areas such as short-term electricity load forecasting [28], carbon price forecasting [39], and electroencephalographic signal detection [2], the high dimensionality of the data and the complexity of the features often result in unsatisfactory prediction results that fail to meet expectations.

By performing feature representation on stock time series, the dimension can be significantly reduced. Using data mining methods, it becomes easier to discover potential patterns in the data, thereby improving data interpretability and prediction accuracy.

The core idea of time series feature representation is to map the original sequence from a high-dimensional space to a low-dimensional feature space or symbolic sequence through dimensionality reduction transformation or symbolic abstraction, thereby achieving the effects of compressing data scale, filtering high-frequency noise, and extracting discriminative features. This preserves the key dynamic patterns and structural information of the time series and enhances the ability to represent real information.

When dealing with stock time series feature representation issues, segmented linear representation, extreme point extraction, and trend analysis have always been the core technical approaches that attract the most attention. However, existing methods often face limitations such as local feature overfitting, loss of global trends, and sparse long segment features. Therefore, this study proposes a new method for feature representation of stock time series data using a multi-level collaborative mechanism. This method initially screens important change points in stock time series using angle changes, then further screens trend turning points using distance metrics, and finally supplements the turning points with error thresholds to ensure trend continuity. Experimental results demonstrate that the method proposed in this study can significantly reduce the volume of raw data, enhance sensitivity to fluctuations in raw stock time series, and better characterize stock price trends, providing important reference value for stock investors.

2. Related work

Two key aspects of time series data mining are time series feature representation and time series similarity measurement. With the iterative evolution of information technology and the deep penetration of machine learning methods, related research paradigms have shown significant disciplinary transfer effects, gradually expanding to diverse application scenarios such as financial modelling [23], eco-driving recognition [40], emotion recognition [36], and biomedical diagnosis [1].

Given the high-dimensional, massive, and complex nature of time series data, Faloutsos et al. [9] propose constructing a low-dimensional, robust feature representation space and applying lossy dimensionality reduction compression to the original sequence, which yields more significant effects on data mining. In the process of time series feature representation, the core lies in mapping the multidimensional observations of the original sequence to a low-dimensional space composed of discriminative feature vectors. This

mapping must achieve two objectives: (1) significantly compress the data scale to reduce storage and computational overhead; (2) maximize the preservation of the global evolutionary trends and local structural patterns of the sequence during the compression process.

Therefore, the design of feature representation methods must satisfy three evaluation criteria:

- High dimensionality reduction efficiency: significantly reduce the size of the original sequence data;
- Strong feature fidelity: retain the overall trend pattern and key local features of the sequence;
- Balance between computational efficiency and representation accuracy: achieve high-precision feature encoding with limited computational resources.

In recent years, many excellent methods have been proposed to address the problem of time series feature representation. These methods can be broadly categorized into segment-based feature representation methods, domain transformation-based feature representation methods, and model-based feature representation methods, as shown in Fig1. To assess the similarity of time series or their segmented representations, various metrics have been applied to time series similarity measurement, including Euclidean distance, triangular distance, envelope-based metrics, distance based on dynamic time warping, and distance calculated based on the longest common subsequence of two time series.

2.1. Time series feature representation methods

2.1.1 Segment-Based Feature Representation Methods The Piecewise Linear Representation (PLR) method was first proposed by Wu et al. in 2004. They used the results of PLR to study data similarity and stock price prediction [38]. In this method, the selection of segment points is particularly critical [22]. The PLR method is simple in form and intuitive in interpretation, but early algorithms had large fitting errors. As the scale of time series data continues to expand, researchers are paying more attention to segmented linear representations based on important point strategies, and trend research has gradually become a hot topic. Wen et al. [32] proposed a bidirectional segmented linear representation method, using a set of linear fitting functions to represent the original time series, which can achieve dimension reduction while maintaining its dynamic characteristics. Hung et al[13]. used the mean and slope to perform piecewise approximation, but this evenly spaced piecewise result easily distorts the overall trend of the time series. Sun et al.[33] combined the trend distance index with the mean to construct a symbolic description, in order to more comprehensively characterize the overall evolutionary trend of the sequence segment. Zhang et al. [41] designed a symbol representation method based on trend feature enhancement, which retains the mean while integrating trend distance factors and trend pattern factors to jointly encode the dynamic characteristics of the sequence. Zhou et al. [45] proposed a segmented linear representation combining turning points and maximum absolute deviation in stock time series to extract sequence features. Liu et al.[43] proposed a turning point extraction strategy based on angle change thresholds, which calculates the directional angle change between adjacent segments and combines trend segment features to evaluate and screen for significant turning points. Liao et al.[20]

designed a distance-driven threshold discrimination method, which determines the importance of candidate points by comparing them with a preset distance threshold. However, its underlying basic inflection point screening relies solely on the numerical relationship between three adjacent points, resulting in limited robustness. Chen et al.[4] proposed a linear representation method for time series key points, which comprehensively considers the time span and amplitude changes and can efficiently extract key points from time series.

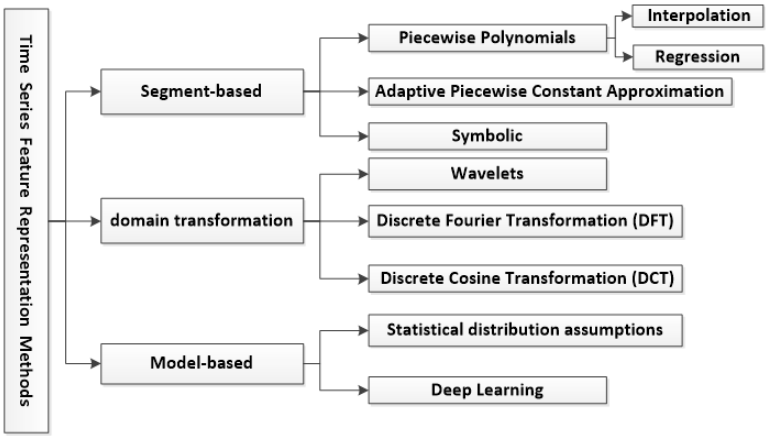


Fig. 1. Time Series Feature Representation Methods

Based on PLR, several classic algorithms such as piecewise aggregate approximation (PAA) [16], top-down (TD) [19], and bottom-up (BU) [30] piecewise linear representation methods have been proposed. The PAA algorithm achieves dimension reduction by dividing time series into equal intervals and using the average value of each segment to represent the original sequence. This method does not take into account the differences between adjacent samples, and the results are often unsatisfactory. Without addressing this shortcoming, optimization research on PAA has primarily focused on two directions: feature representation and segmentation strategies. On one hand, Huang et al. [14] enhanced the mean representation by integrating auxiliary features such as slope and variance. On the other hand, Keogh et al. [17] proposed the Adaptive Partitioning Constant Approximation (APCA), which utilizes dynamic programming to achieve sequence-adaptive optimal segmentation, resulting in non-uniformly partitioned segments. Additionally, to simultaneously improve dimension reduction efficiency and enhance the representation of sequence macro trends, researchers proposed the Symbolic Aggregate Approximation (SAX) method [21] and its improved frameworks[15,12,44].

Information granularity, as a key extension of piecewise linear representation, focuses on abstracting sequences into granular structures with semantic meaning. Based on information granularity theory [31], Duan et al. [8] developed a time series granularity model and constructed a granularity-driven clustering framework based on this model. Guo et al. [11] designed a trend feature-guided granularity method and verified its representational

validity in clustering tasks, confirming its advantages in representing time series structural features.

2.1.2 Feature representation methods based on domain transformation The core of feature representation based on domain transformation lies in mapping time series to a transformation domain independent of the original data, where the transformation operator and feature selection do not depend on the inherent characteristics of the data. Typical transformation spaces include the frequency domain, spectral space, etc. Commonly used methods include the Discrete Fourier Transform (DFT)[18] and the Discrete Wavelet Transform (DWT) [3].

DFT converts time-domain sequences into complex frequency-domain representations that characterize periodicity and frequency structure using trigonometric basis functions. DFT preserves data length and is suitable for smooth sequences. However, it has weak ability to capture local sudden changes. DWT retains local characteristics in both time and frequency through multi-scale decomposition, and the wavelet coefficients it generates can accurately describe the transient modes of non-stationary signals. Compared with DFT, DWT demonstrates superior feature representation robustness in non-stationary time series such as highly volatile financial data. Panda et al[29]. achieved good results in tumor classification by constructing models using wavelet-based feature learning and deep learning-based feature learning. De et al[37]. proposed an end-to-end framework for stock price prediction called the Multi-level Wavelet Decomposition Interaction Network (MWDINet). MWDINet employs multi-scale wavelet decomposition interaction modules (MWDI-Block) and Heron moving average modules (HMA-Block) to extract frequency-domain and time-domain information from data, respectively, demonstrating significant potential in stock price prediction.

2.1.3 Model-based feature representation methods Model-based feature representation methods assume that time series are generated by specific mathematical models, and extract features by constructing corresponding models. Their advantages lie in the flexibility of model selection and their derivative capabilities, but they have limitations such as strong assumption dependence, high domain knowledge thresholds, and high computational complexity.

Based on statistical distribution assumptions, the current mainstream methods include autoregressive models (ARM), orthogonal polynomial regression models [10], hidden Markov models (HMM)[24], and kernel models. Such methods require domain-specific prior knowledge to support model selection and are difficult to uniformly capture the coupled effects of long-term dependencies, cyclical patterns, and random fluctuations. Methods based on neural network models utilize deep networks to automatically learn feature representations, giving rise to methods such as CNN-RNN hybrid architectures [26], WDCNN-LSTM[5], multi-channel deep convolutions, and hierarchical bidirectional learning [34]. Lu et al[25]. developed a time series-based recurrent neural network (TRNN) model specifically optimized for the characteristics of financial time series data. The model achieves efficient data dimensionality reduction by intelligently extracting key points of market trends and turning signals, and reinforces the impact of recent trading volume on prices through a dynamic weighting mechanism, significantly improving prediction accuracy while ensuring computational efficiency. Wang et al[35].

proposed an enhanced interval value decomposition integral model based on comprehensive feature extraction and optimized deep learning, utilizing interval variational modal decomposition (FIVMD) with a feedback mechanism to extract internal features, achieving satisfactory prediction results. Although these methods have overcome the assumption constraints of statistical models and enhanced the representational capabilities of big data, the low interpretability, high computational overhead, and weak generalization performance of neural networks in financial time series data remain critical issues that require optimization.

2.2. Time series similarity measurement methods

After time series feature representation, time series similarity measures can be used to obtain the degree of similarity between sub-sequences and between sub-sequences and the original sequence. Such measures can usually be defined by a difference function, also known as a distance function[27]. Euclidean distance has very efficient computational performance, making it a commonly used method for measuring the distance between time series. Euclidean distance metrics require sequences to be of equal length to achieve "one-to-one" matching calculations between data, which also limits its scope of application. Dynamic Time Warping (DTW) is a highly accurate and robust time series distance measurement method originally applied in the field of speech recognition. Compared to traditional Euclidean distance, DTW can match data points in time series by warping the time domain of the time series, not only achieving better shape measurement effects but also enabling the measurement of two time series of unequal length.

In recent years, to alleviate the computational bottleneck of early time series similarity measurement methods, Choi et al. [6]proposed the fast Sakoe-Chiba DTW (SC-DTW) and the fast incremental DTW (I-DTW), which achieved a significant improvement in computational speed compared to the traditional DTW. Zhang et al. [42] proposed a time series similarity measurement method that combines sequence decomposition and DTW. This method has lower computational complexity than DTW and demonstrates significant efficiency in classification tasks. Additionally, for the granular representation of time series information, some scholars have adopted the integral area method to measure the distance between the same linear segments.

In the highly complex stock market, stock price sequences are influenced by the coupled effects of multiple heterogeneous factors, coupled with significant interference from high-frequency noise, resulting in their pronounced volatility and inherent instability. To more effectively identify and analyze the underlying trend patterns within such sequences, existing research primarily relies on linear transformation methods and feature representation techniques such as piecewise aggregation approximation. However, the effectiveness of these methods is highly dependent on pre-set fixed or adaptive parameter thresholds. This parameter dependency not only introduces significant subjective risks but may also lead to information loss or false pattern recognition due to improper threshold selection, thereby facing adaptive limitations when handling highly complex, non-stationary financial time series. Therefore, this study focuses on how to retain the main trend features of stock time series while mitigating the limitations caused by segmented linear fitting, such as local feature overfitting, global trend loss, and long segment feature sparsity. It also employs appropriate similarity measurement methods to compare the advantages and disadvantages of different feature representation methods.

3. Preliminary

This section provides definitions related to the methods proposed in this study.

Definition 1 (time series): A time series is an ordered set of observations collected in chronological order. Recorded as $X = \{(x_1, t_1), (x_2, t_2), \cdots, (x_n, t_n)\}$.Where$(x_i, t_i)$ represents the observed value at time t_i and satisfies $t_1 < t_2 < \cdots < t_n$. If the sampling interval is constant, i.e., $\Delta t = t_{i+1} - t_i$ is a constant, then the time series can be simplified to a numerical sequence form: $X = \{x_1, x_2, \cdots, x_n\}$. For stock time series, stock price data comes from every transaction in the stock market, and there is a fixed time interval between each transaction within the specified trading time. In the stock market, daily transaction data is not continuous in time, but due to the fixed trading hours, stock price data at different time granularities can be represented as:$P = \{p_1, p_2, \cdots, p_n\}$.

Definition 2 (Segmented Linear Representation of Stock Time Series): Divide the stock time series P into m segments, each of which is approximated by a straight line between the endpoints. The formal definition is:

$$P_{PLR} = \{(p_{s1}, p_{e1}), (p_{s2}, p_{e2}), \cdots, (p_{sm}, p_{em})\} \tag{1}$$

Where, $p_{sk}, p_{ek} \in P$,$ek = sk + 1$. Since the segment points are interconnected, the sequence represented by the segments can be expressed as: $P_{PLR} = \left\{p_1^*, p_2^*, \ldots, p_{m+1}^*\right\}$.

Definition 3 (fitting error): The fitting error is obtained by linearly interpolating line P_{PLR} to obtain the approximate value $P' = \{p_1{}', p_1{}', \cdots, p_n{}'\}$ of the entire sequence.

$$Error = \sqrt{\sum_{i=1}^{n} (p_i - p_i{}')^2} \tag{2}$$

Definition 4 (inclination angle turning points): Given a stock price time series $P = \{p_1, p_2, \cdots, p_n\}$, its slope turning point series $P^s = \{p_1^s, p_2^s, \ldots, p_m^s\}$. The absolute angle change value α_i at point p_i must satisfy:

$$\alpha_i = \begin{cases} \pi - (|\theta^-| - |\theta^+|), \text{if } \theta^- \times \theta^+ < 0 \\ |\theta^- - \theta^+|, \text{otherwise} \end{cases} \geq \delta \tag{3}$$

Where, $\theta^- = \arctan\left(\frac{p_i - p_{i-1}}{\Delta t}\right)$ is the direction angle of the front line segment, $\theta^+ = \arctan\left(\frac{p_{i+1} - p_i}{\Delta t}\right)$ is the direction angle of the rear line segment, and δ is the angle change threshold.

As defined in Definition 4, the smaller the angular change value of a point, the closer the lines connecting the three points before and after that point as the midpoint approach a straight line, indicating a smaller trend change. Consequently, the probability of that point being a trend inflection point is smaller, while the opposite is true for a larger angular change value. As shown in Fig2, Fig2(a) displays the K-line chart and closing prices of the stock (SZ300059) from February 6, 2025, to March 5, 2025. Fig2(b) shows the sequence of angle change values for the stock's closing prices. As can be seen, the stock price time series exhibits significant volatility, and angle change values can be used to efficiently select segmentation points and achieve the preliminary goal of reducing the number of time series.

Definition 5 (Stock Time Series Trend): The stock time series trend is determined by connecting adjacent inclination angle turning points. Given a stock time series inclination angle turning point sequence $P^s = \{p_1^s, p_2^s, \ldots, p_m^s\}$, the trend is defined as $T = \{t_1, t_2, \ldots, t_j, \ldots t_{m-1}\}$. Here, t_j represents the connecting line segment between turning points $p_j$andp_{j+1}.

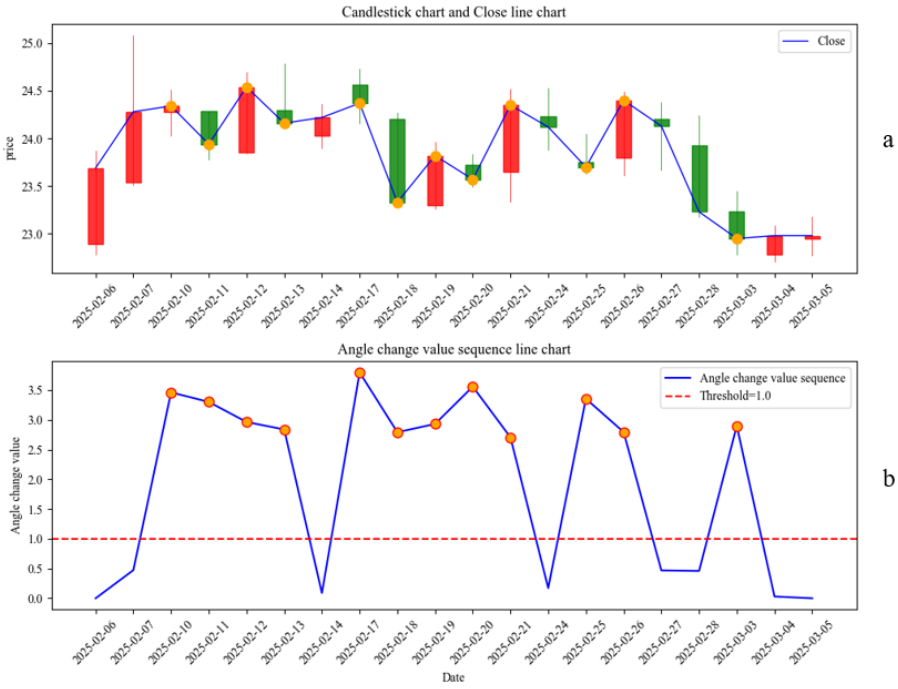


Fig. 2. Example diagram of angle change values

Definition 6 (Stock Time Series Trend Turning Point): For the inclination angle turning point sequence $P^s = \{p_1^s, p_2^s, \ldots, p_m^s\}$, take the three consecutive points p_{j-1}^s, p_j^s and p_{j+1}^s as the minimum pattern for trend formation. Use the vertical distance D_v between the line segments connecting p_j^s to p_{j-1}^s and p_{j+1}^s as the standard for measuring its importance. The greater the distance, the greater the importance of p_j^s in the trend turning point, and the more likely it is to become an important node in the trend turning point. Form the trend turning point sequence: $P^t = (p_1^t, p_2^t, \ldots, p_x^t)$.

$$D_v = \left| p_{j-1}^s + \frac{\left(p_{j+1}^s - p_{j-1}^s\right)(t_2 - t_1)}{t_3 - t_1} - p_j^s \right| \tag{4}$$

Definition 7 (maximum error point): For sub-sequences formed by trend turning points in stock time series, if the length of the sub-sequence segment exceeds the threshold or the fitting error exceeds the threshold, calculate the absolute error between the fitted line

and the original data points: $E = max\left(|p_i - f\left(x_i\right)|\right)$. Select the maximum error point as the supplementary trend turning point.

4. A method for representing stock time series characteristics based on trends and inclination angle turning points

4.1. Method description

The stock time series feature representation method based on trends and inclination angle turning points proposed by this research institute uses a multi-level collaborative mechanism supplemented by inclination angle turning points, trend turning points, and maximum error points to greatly reduce the amount of data while retaining the trend characteristics of the original time series data to the greatest extent possible, laying the foundation for time series analysis. The specific algorithm steps are as follows:Given a stock time series $P = \{p_1, p_2, \cdots, p_n\}$, angle change threshold δ, distance threshold D, maximum error threshold η, and maximum segment length L.

I. Calculate the angle turning points of the stock time series. According to Definition 4, calculate the absolute angle change value of each data point in order, filter out the data points greater than the angle change threshold, add the point to the inclination angle turning point sequence $P^s = \{p_1^s, p_2^s, \ldots, p_m^s\}$, and form the trend segment $T = \{t_1, t_2, \ldots, t_j, \ldots t_{m-1}\}$ according to Definition 5.
II. Based on Definition 6, calculate the perpendicular distance between A and B and the line connecting C, and select points greater than the distance threshold D as trend turning points.
III. Based on the sequence length threshold L select the sequence segment greater than L in II or the sequence segment with a fitting error greater than the maximum error threshold, calculate the absolute error E between the fitting line and the original data points in the sequence segment, and select the largest error point as the supplementary trend turning point.
IV. Trend continuity test. For the trend turning point sequence formed in II and III, calculate the slope of adjacent trend segments, merge trend segments with the same change trend, and ensure that the trend segments are always in a state of alternating rises and falls.

The algorithm flowchart is shown in Fig3. The algorithm pseudocode is shown in Fig4.

Parameter Description: The algorithm in this paper involves four key parameters: angle change threshold δ , distance threshold D, maximum error threshold η, and maximum segment length L. δ and D jointly control the sensitivity of trend inflection point detection. Setting the threshold too low may cause redundant local fluctuations to be misjudged as trend changes, while setting it too high may result in important trend inflection points being missed. η and L jointly regulate the granularity distribution of trend segments. By constraining the upper limit of segmentation error and the maximum interval length, they ensure feature density while avoiding over-segmentation issues. When optimizing parameters, it is recommended to prioritize balancing the values of δ and D , while η and L can be appropriately relaxed to improve the continuity of long trends.

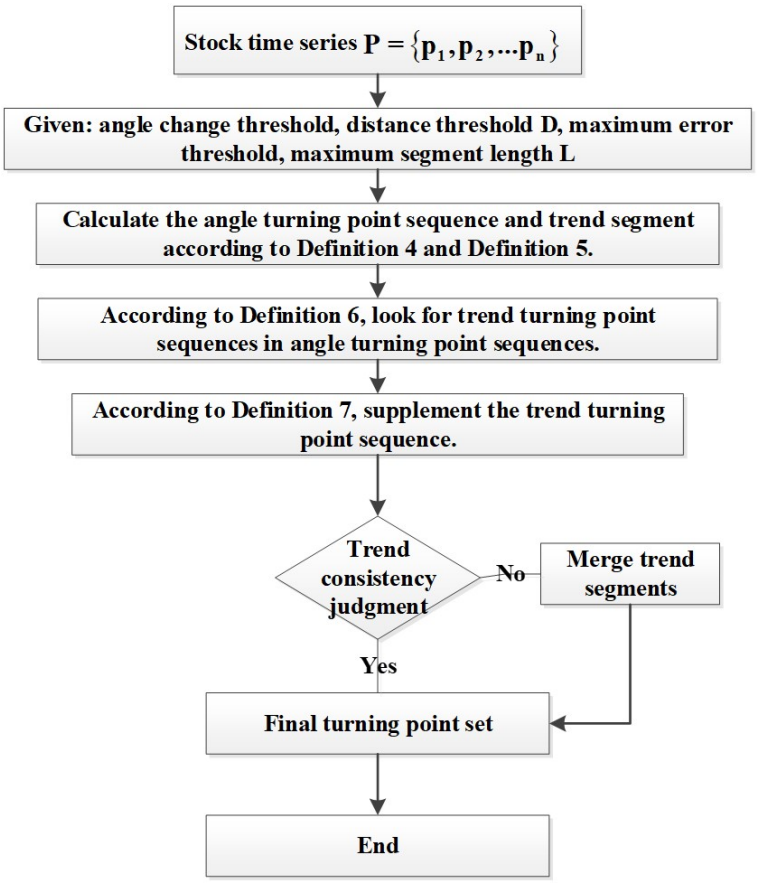


Fig. 3. Algorithm flowchart

4.2. Evaluation indicators

Similarity measurement is an important part of evaluating the quality of time series feature representation methods. This study uses Euclidean distance and integral distance to evaluate the fitting error and morphological similarity between the feature-represented sequence and the original sequence data.

4.2.1 European distance European distance refers to the distance between two objects in m-dimensional space. Due to its highly efficient computational performance, it is commonly used to measure the distance between time series. The calculation formula is shown in Eq5:

European distance in two-dimensional space:

$$Dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{5}$$

Input: A set of stock time series data P; angle change threshold; distance threshold D; maximum error threshold; maximum segment length L.
Output: Stock Time Series Trend Turning Point Sequence X

P=p1, p2, … , pi
Define set X to store trend turning points, initialized as empty.
Define error set E, used to store the fitting error of the trend turning point, initialized as empty.
Definition function find_Angular inflection point (P, δ) P^s // Searching for angle turning point sequences
Definition function find_Trend turning points (P^s) P^t // Finding trend turning point sequences
Definition function Additional turning point （ P^t ） E // Supplemental turning point sequence。
while True
  if $E = \max(|p_i - f(x_i)|) > \eta$
    P^t append p_i
      if K(p_{i-1}^t, p_i^t)*K(p_i^t, p_{i+1}^t)>0
        P^t=P^t （p_{i-1}^t, p_{i+1}^t）
      Else
        P^t=P^t
  Else
    P^t=P^t
    X=P^t
  Break
Return X

Fig. 4. Algorithm pseudocode

4.2.2 Area distance Area-based similarity measurement methods can measure the morphological similarity between feature representations and original sequences after linear interpolation. For stock time series, the more similar the morphology, the more important it is for guiding future investment decisions. Therefore, this study uses the area distance method to measure the degree of morphological similarity between feature representations and original sequences.

Given the original stock time series A, the sequence B obtained by linear interpolation after feature representation is used, where$n = m$. There are two types of relationships between the two sequences: intersecting and non-intersecting, as shown in Fig5. As shown in Fig5(a), the integral distance when two sequences do not intersect is:

$$Dist = \int_{t_1}^{t_2} [(k_1 t + b_1) - (k_2 t + b_2)]\, dt \tag{6}$$

As shown in Fig5(b), the integral distance between the two sequences relative to $t^\star$ is:

$$Dist = \int_{t_1}^{t^\star} [(k_1 t + b_1) - (k_2 t + b_2)]\, dt + \int_{t^\star}^{t_2} [(k_2 t + b_2) - (k_1 t + b_1)]\, dt \tag{7}$$

5. Analysis and evaluation of experimental results

5.1. Experimental Process

5.1.1 Experimental Environment When implementing the algorithm for identifying extreme points in stock time series trends, we selected a series of development components. The specific selection of these components is shown in Table1.

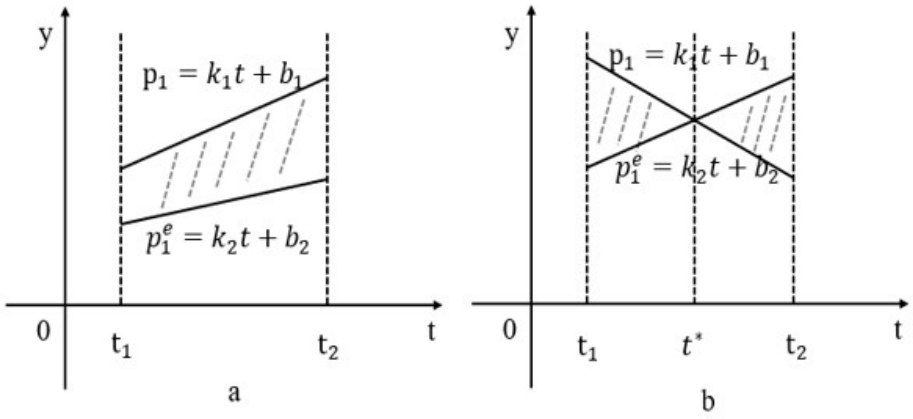


Fig. 5. Relationships between sequences

Table 1. Experimental environment setting

Name	Configurations
Processor CPU	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz
Memory RAM	DDR4 8GB 2400MHz
Hard disk	128GB Solid State Drive (Toshiba)
Operating System OS	Windows 10 Home Edition 19043.1889
Programming Language	Python3.5.0
IDE	JetBrains PyCharm 2018.3.2
Toolkits	Pandas NumPy et.al

5.1.2 Dataset To ensure that the research findings possess strong industry representativeness and external validity, this study adhered to strict industry coverage and representativeness principles when designing the experimental dataset. Specifically, the primary dataset selected leading listed companies from different industry sectors traded on China's two major securities exchanges—the Shanghai Stock Exchange (SSE) and the Shenzhen Stock Exchange (SZSE)—as the research subjects. The time span is from March 1, 2024, to February 28, 2025, which is a relatively complete annualized operational cycle. This avoids the influence of extreme short-term noise while reflecting market dynamics and structural characteristics with a certain degree of continuity. The stock data is at a 30-minute interval, which, compared to traditional daily data, can more precisely reveal intraday price behavior patterns, short-term volatility characteristics, and potential microstructural effects. The data source is Wind Financial Terminal (Wind), which is a widely recognized and authoritative professional financial data and analysis service provider in China's financial markets. Its database boasts high completeness, timeliness, and accuracy, making it a commonly used and reliable data source for empirical financial research in both academic and professional circles domestically and internationally. Specific stock information is shown in Table2.

Table 2. Data set stock information

Num	Code	Name	Data volume	Industry
1	000063SZ	ZTE	1928	Industrial Communication Device, 5G
2	000333SZ	Midea Group	1928	Electrical and Mechanical Industries
3	001979SZ	China Merchants Shekou Industrial Zone Holdings	1928	Real Estate Development Sector
4	002371SZ	NAURA Technology Group	1928	Semiconductor and electronic components sector
5	002594SZ	BYD	1928	Automotive Manufacturing Sector
6	600030SH	Citic Securities	1928	Big finance, securities
7	600519SH	KweichowMoutai	1928	Distillery Industry
8	601138SH	Foxconn Industrial Internet	1928	Industrial Internet
9	601985SH	China National Nuclear Power	1928	Power industry
10	688256SH	Cambricon Technologies Corporation Limited	1928	Artificial Intelligence Chip

Data preprocessing is an important part of data mining. Through analysis of the original dataset, the stock data selected for this study shows no missing values between March 1, 2024, and February 28, 2025. The 30-minute line data consists of eight data points for each trading day, totaling 241 trading days. To fully reflect the original state of the stock time series, this study did not standardize the original series.

5.2. Analysis of experimental results

In order to verify the validity of the stock time series feature representation method based on inclination angle and trend turning points proposed in this study, we selected the stock time series feature representation method based on trend feature points (TFP) [17], the time series trend feature extraction method based on turning points and trend segments (TPTS)[21], PLR-ITTP [15], and PLR-KP [12]for comparison.

5.2.1 European distance After feature representation, the method proposed in this study uses linear interpolation to fill in the time series. The other four methods also use linear interpolation to fill in the gaps, and then calculate the European distance between the filled-in sequence and the original sequence. Table3 shows the European distance of different methods at a compression rate of 70%.

Table 3. European distance for different methods at a compression rate of 70%

	TPTS	PLR-KP	PLR-ITTP	TFP	Our
000063SZ	4.2697	9.7442	5.1305	6.8978	**4.1358**
000333SZ	7.0874	17.013	7.9028	9.5392	**6.6985**
001979SZ	2.2585	3.2936	2.4531	3.1167	**2.0417**
002371SZ	78.7373	99.5045	57.5954	77.276	**57.3243**
002594SZ	40.202	66.0142	35.5037	44.3144	**32.8873**
600030SH	2.6216	6.0502	2.9709	5.8952	**2.3209**
600519SH	263.6387	237.6315	**130.4544**	184.0867	130.567
601138SH	4.265	7.8261	5.1287	7.3047	**4.1947**
601985SH	1.3405	2.0641	1.6546	2.077	**1.2407**
688256SH	176.7896	203.3061	116.9405	155.6698	**114.68**

As can be seen from Table3, except for the 600519SH, where the method proposed in this study is slightly greater than PLR-ITTP in terms of European distance, the method proposed in this study has the smallest value in terms of European distance measurement for other stocks, while the PLR-KP method has the largest value for most stocks. This indicates that considering only the key points and time spans of the time series is insufficient for accurately representing stock time series in an effective linear manner. In contrast, the three-level collaborative method proposed in this study achieves excellent results in the segmented linear representation of stock time series. Because the method proposed in this study uses two parameters, the maximum error threshold η and the maximum segment length L, to split long time series that other methods fail to focus on, the length of the sub-segments after feature representation is relatively uniform. In this case, the measured European distance will be smaller than that of other methods, which also shows that the method proposed in this paper can more accurately characterize the original stock time series.

5.2.2 Area distance Based on European distance measurement, the formula is used to calculate the area distance in order to measure the morphological similarity between the

feature representation sequence and the original sequence. Fig6shows the area distance of different methods at a compression rate of 70%.

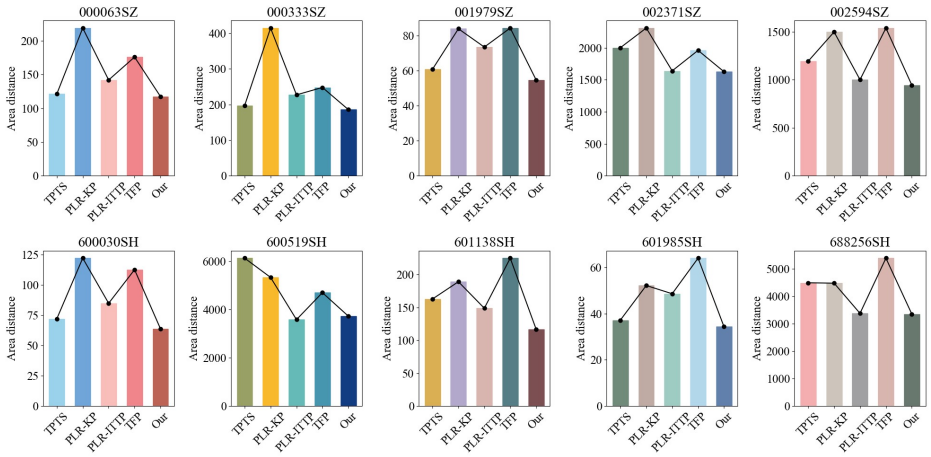


Fig. 6. Area distance of different methods at a compression ratio of 70%

As can be seen from the experimental results in Fig6, there are significant differences among different methods in terms of area distance metrics. The method proposed in this study achieved the minimum area distance for all stocks except 600519SH, indicating that after using the method proposed in this paper for stock time series feature representation, the resulting trend lines of turning points maintain the highest degree of similarity with the original series, better aligning with the original stock time series pattern. While the PLR-ITTP method achieved the minimum area distance for the 600519SH stock, its performance on other stock sequences was unstable. The experimental results indicate that the method proposed in this study has good adaptability and demonstrates robust performance in stock time series feature representation.

5.2.3 Fitting error Based on Eq2, the fitting error between the linear interpolation sequence after feature representation and the original sequence was calculated. The calculation indicators included root mean square error (RMSE) and mean absolute error (MAE). The experimental results are shown in Table 4.

In Table4, bolded data indicate the smallest error. As shown in Table4, among the 10 stock time series, the method proposed in this study achieved the smallest fitting error across all time series, while the PLR-KP method performed poorly in most time series. This indicates that by employing multi-level collaborative screening and supplementation of turning points, it is possible to avoid the premature rejection of important turning points to the greatest extent. By preserving the overall trend of the time series and adequately reducing dimensionality, the method mitigates the limitations of local feature overfitting, loss of global trends, and sparsity of long-term features, thereby reducing fitting errors.

Table 4. Fitting error of different methods at a compression ratio of 70%

	TPTS		PLR-KP		PLR-ITTP		TFP		Our	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
000063SZ	0.0972	0.066	0.2219	0.1174	0.1168	0.0757	0.1571	0.0931	**0.0942**	**0.0633**
000333SZ	0.1614	0.1082	0.3875	0.2221	0.18	0.1212	0.2172	0.1328	**0.1526**	**0.1008**
001979SZ	0.0514	0.0331	0.075	0.0449	0.0559	0.0389	0.071	0.0448	**0.0465**	**0.0291**
002371SZ	1.7932	0.9954	2.2662	1.2308	1.3117	0.8753	1.7599	1.0462	**1.3055**	**0.8684**
002594SZ	0.9156	0.5388	1.5034	0.8026	0.8086	0.5343	1.0092	0.6022	**0.749**	**0.5053**
600030SH	0.0597	0.039	0.1378	0.065	0.0677	0.0449	0.1343	0.0599	**0.0529**	**0.0342**
600519SH	6.0042	3.0027	5.4119	2.8604	2.971	1.9852	4.1925	2.5142	**2.9736**	**1.9845**
601138SH	0.0971	0.065	0.1782	0.1013	0.1168	0.0793	0.1664	0.0939	**0.0955**	**0.0632**
601985SH	0.0305	0.0203	0.047	0.0281	0.0377	0.0258	0.0473	0.0284	**0.0283**	**0.0184**
688256SH	4.0263	2.0858	4.6302	2.4118	2.6632	1.7871	3.5453	1.9661	**2.6118**	**1.7689**

This further demonstrates the effectiveness and superiority of the method proposed in this study.

5.2.4 Fitting error of each method at different compression rates After calculating the European distance, area distance, and fitting error of various methods under the same compression ratio, the effectiveness of the method proposed in this study was verified. In this section, we examined the fitting error of the five methods for stock 002371SZ at compression ratios of 70%, 75%, 80%, 85%, and 90%. The experimental results are shown in Fig7 and Fig8.

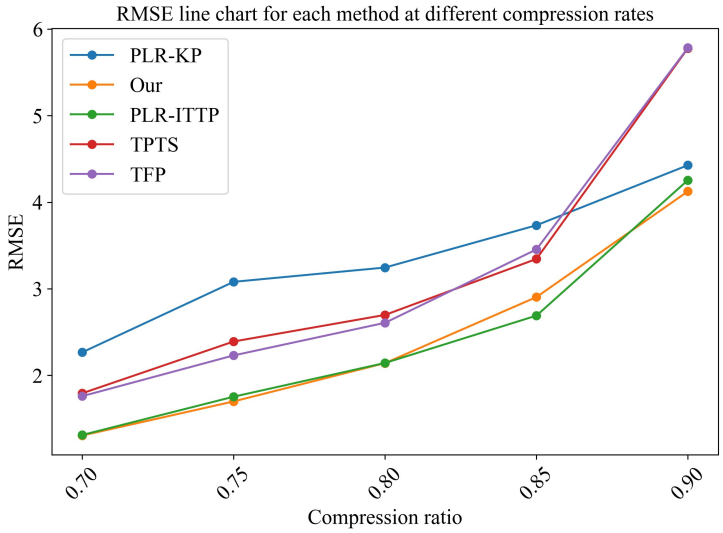


Fig. 7. RMSE line charts for each method at different compression rates

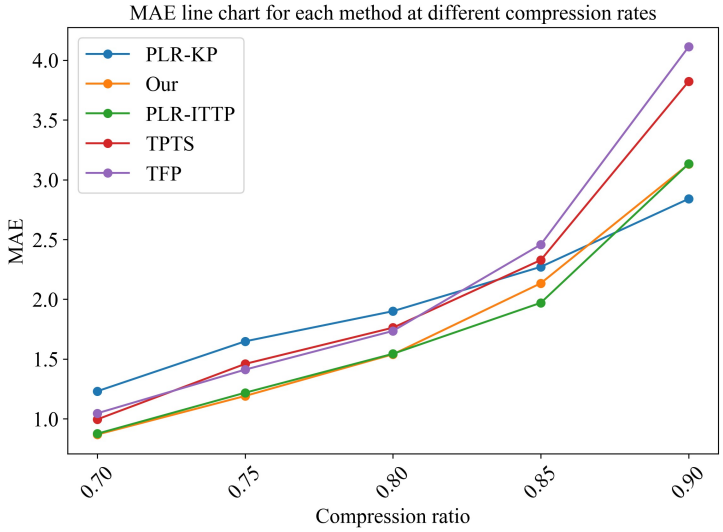


Fig. 8. MAE line charts for each method at different compression rates

As shown in Fig7 and Fig8, the proposed method achieves the minimum RMSE and MAE values before a compression rate of 80%. When the compression rate exceeds 80%, the proposed method still achieves relatively low RMSE and MAE values. PLR-ITTP and PLR-KP perform well at higher compression rates. Meanwhile, as the compression rate increases, the growth of RMSE and MAE is relatively stable, indicating the stability of the proposed method in terms of fitting performance.

6. Conclusions

Feature representation of stock time series data is a process aimed at transforming raw high-dimensional data into a low-dimensional form. Its core objective is to significantly improve the efficiency of subsequent data mining, similarity measurement, and other related research tasks while maximizing the retention of key information from the original sequence. Based on the segmented linear representation of time series, this study proposes a stock time series feature representation method based on a combination of inclination angle changes and trend turning point screening. Through a multi-level collaborative mechanism, this method alleviates the limitations of local feature overfitting, global trend loss, and long segment feature sparsity in a single method. The study reached the following conclusions:

– The stock time series feature representation method proposed in this study, which combines inclination angle changes with trend turning point screening, has achieved good results in terms of European distance, area distance, and fitting error evaluation indicators. It can effectively retain the original features of stock time series and maintain good stability.

– The method proposed in this study can more accurately capture trend features in stock time series analysis, providing investors with more reliable and precise decision-making basis. This research finding has important implications for understanding the stock market and making investment decisions.

However, this study still has some limitations: first, although the proposed multi-level collaborative processing mechanism improves the depth of analysis of the original sequence, the increased algorithm complexity may affect its computational efficiency on large-scale datasets. Second, the current validation is only applicable to specific types of stock time-series data, and the generalization ability of the method needs further investigation.

Future work will focus on the following directions: (1) reducing computational complexity through algorithm optimization to improve processing efficiency; (2) validating the method's adaptability in broader data scenarios (such as high-frequency trading data and cross-market indices); and (3) exploring its integration with deep learning models to enhance its stability and predictive capabilities in noisy market environments. These improvements will help advance the practical application of this method in complex financial scenarios.

References

1. Abineza, C.and Balas, V., Nsengiyumva, P.: Deep learning model for early subsequent copd exacerbation prediction. Studies in Informatics and Control 32(3), 99–107 (2023)
2. Bae, H.J., Park, J.S., Choi, J.H., Kwon, H.Y.: Learning model combined with data clustering and dimensionality reduction for short-term electricity load forecasting. Scientific Reports 15(1), 3575 (2025)
3. Buhler, J., Tompa, M.: Finding motifs using random projections. In: Proceedings of the fifth annual international conference on Computational biology. pp. 69–76 (2001)
4. Chen, S., Lv, X., Qi, R., Wang, L., Yu, L.: Linear representation method based on key points for time series. Computer Science 43(05), 234–237 (2016)
5. Chen, Z., Zhao, J.: Bearing fault diagnosis based on wide deep convolutional neural network and long short term memory. Tehnički vjesnik 30(1), 265–273 (2023)
6. Choi, W., Cho, J., Lee, S., Jung, Y.: Fast constrained dynamic time warping for similarity measure of time series data. IEEE Access 8, 222841–222858 (2020)
7. Dars, M.A., Liu, Q., Chaozhu, Z., Shaikh, S., Farid, G.: A context-aware recommendation system based on hadoop big data. STUDIES IN INFORMATICS AND CONTROL 34(1) (2025)
8. Duan, L., Yu, F., Pedrycz, W., Wang, X., Yang, X.: Time-series clustering based on linear fuzzy information granules. Applied Soft Computing 73, 1053–1067 (2018)
9. Faloutsos, C.and Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. ACM Sigmod Record 23(2), 419–429 (1994)
10. Fuchs, E., Gruber, T., Pree, H., Sick, B.: Temporal data mining using shape space representations of time series. Neurocomputing 74(1-3), 379–393 (2010)
11. Guo, H., Wang, L., Liu, X., Pedrycz, W.: Trend-based granular representation of time series and its application in clustering. IEEE Transactions on Cybernetics 52(9), 9101–9110 (2021)
12. Hu, Y., Guan, P., Zhan, P., Ding, Y., Li, X.: A novel segmentation and representation approach for streaming time series. IEEE Access 7, 184423–184437 (2018)
13. Hung, N., Anh, D.: An improvement of paa for dimensionality reduction in large time series databases. In: Pacific rim international conference on artificial intelligence. pp. 698–707 (2008)

14. Hung, N., Anh, D.: An improvement of paa for dimensionality reduction in large time series databases. In: Pacific rim international conference on artificial intelligence. pp. 698–707. Springer (2008)
15. Ji, C., Liu, S., Yang, C., Wu, L., Pan, L., Meng, X.: A piecewise linear representation method based on importance data points for time series data. In: 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 111–116. IEEE (2016)
16. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems 3(3), 263–286 (2001)
17. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. In: Proceedings of the 2001 ACM SIGMOD international conference on Management of data. pp. 151–162 (2001)
18. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. In: Proceedings of the 2001 ACM SIGMOD international conference on Management of data. pp. 151–162 (2001)
19. Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Kdd. vol. 98, pp. 239–243 (1998)
20. Liao, J., Zhou, Z., Kou, Y., Luo, H.: Time series piecewise linear representation based on trend transition point. Computer Engineering and Applications 46(30), 50–53+81 (2010)
21. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Mining and knowledge discovery 15(2), 107–144 (2007)
22. Liu, S., Jiang, H.: A review on time series representation for similarity-based pattern search. Computer Engineering Apply 27, 53–57 (2004)
23. Liu, Z.and Liu, X., Zhou, L.: A hybrid cnn and lstm based model for financial crisis prediction. Tehnički vjesnik 31(1), 185–192 (2024)
24. Liu, Z., Wang, S.: Decoding chinese stock market returns: Three-state hidden semi-markov model. Pacific-Basin Finance Journal 44, 127–149 (2017)
25. Lu, M., Xu, X.: Trnn: An efficient time-series recurrent neural network for stock price prediction. Information Sciences 657, 119951 (2024)
26. Lu, X., Tian, J., Liao, Q., Xu, Z., Gan, L.: Cnn-lstm based incremental attention mechanism enabled phase-space reconstruction for chaotic time series prediction. Journal of Electronic Science and Technology 22, 100256 (2024)
27. Medrano, C.L., Alonso, G.D.C., Lafarga, O.V., Cervantes, J.L.V.: Mcclusteringsm: An approach for the multicriteria clustering problem based on a credibility similarity measure. Computer Science and Information Systems 21(3), 1147–1177 (2024)
28. Meena, H.K., Sharma, R., Tailor, A., Verma, H., Saini, R.: Detection of epilepsy using graph signal processing of eeg signals with three features. In: Pattern Recognition and Data Analysis with Applications, pp. 569–578. Springer (2022)
29. Panda, N., Muduli, D., Parija, S., Sharma, S.: Enhanced multi-class brain tumor classification model using mri images: A wavelet and deep learned features based approach. In: 2025 International Conference on Microwave, Optical, and Communication Engineering (ICMOCE). pp. 1–5. IEEE (2025)
30. Park, S.and Lee, D., Chu, W.: Fast retrieval of similar subsequences in long sequence databases. In: Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99)(Cat. No. PR00453). pp. 60–67. IEEE (1999)
31. Pedrycz, W., Homenda, W.: Building the fundamentals of granular computing: A principle of justifiable granularity. Applied Soft Computing 13(10), 4209–4218 (2013)
32. Shi, W., Azzopardi, G., Karastoyanova, D., Huang, Y.: Bidirectional piecewise linear representation of time series with application to collective anomaly detection. Advanced Engineering Informatics 58, 102155 (2023)

33. Sun, Y., Li, J., Liu, J., Sun, B., Chow, C.: An improvement of symbolic aggregate approximation distance measure for time series. Neurocomputing 138, 189–198 (2014)
34. Wang, H., Zhang, Y., Liang, J., Liu, L.: Dafa-bilstm: Deep autoregression feature augmented bidirectional lstm network for time series prediction. Neural Networks 157, 240–256 (2023)
35. Wang, J.and Liu, J., Jiang, W.: An enhanced interval-valued decomposition integration model for stock price prediction based on comprehensive feature extraction and optimized deep learning. Expert Systems with Applications 243, 122891 (2024)
36. Wang, X., Chuncao, L., Liu, Y., Liang, W., Kuanching, L., Poniszewska-Maranda, A.: A spatiotemporal graph neural network for eeg emotion recognition based on regional and global brain. Computer Science and Information Systems (00), 53–53 (2025)
37. Wen, D., Zhao, T., Fang, L., Zhang, C., Li, X.: Mwdinet: A multilevel wavelet decomposition interaction network for stock price prediction. Expert Systems with Applications 238, 122091 (2024)
38. Wu, H.and Salzberg, B., Zhang, D.: Online event-driven subsequence matching over financial data streams. In: Proceedings of the 2004 ACM SIGMOD international conference on Management of data. pp. 23–34 (2004)
39. Xu, W., Wang, J., Zhang, Y., Li, J., Wei, L.: An optimized decomposition integration framework for carbon price prediction based on multi-factor two-stage feature dimension reduction. Annals of Operations Research 345(2), 1229–1266 (2025)
40. Yan, L., Jia, L., Guo, J., Lu, S.: A simulation study on the identification of eco-driving behaviour. International Journal of Simulation Modelling 21(3), 489–500 (2022)
41. Zhang, B., Ying, L., Yu, Y.: Novel symbolic aggregate approximation approach for time series based on trend features. Journal of Computer Applications 42(S1), 123–129 (2022)
42. Zhang, Q., Zhang, C., Cui, L., Han, X., Jin, Y., Xiang, G., Shi, Y.: A method for measuring similarity of time series based on series decomposition and dynamic time warping. Applied Intelligence 53(6), 6448–6463 (2023)
43. Zhang, Y., Li, J., Bai, H., Wang, Z.: Trend feature extraction method for time series based on turning point and trend segment. Computer Applications 40(S1), 92–97 (2020)
44. Zhao, D., Liu, S., Miao, Z., Zhang, H., Wei, Y., Xiao, S.: A novel feature extraction approach for mechanical fault diagnosis based on esax and bow model. IEEE Transactions on Instrumentation and Measurement 71, 1–11 (2022)
45. Zhou, M., Yi, J., Yang, J., Sima, Y.: Characteristic representation of stock time series based on trend feature points. IEEE Access 8, 97016–97031 (2020)

Lei Han is a PhD student at the School of Economics and Management, University of Science and Technology Beijing, with research interests in financial time series analysis and data mining.

Xuedong Gao is a professor at the School of Economics and Man- agement, University of Science and Technology Beijing, with research interests in management process optimization, financial time series analysis, and data mining.

Received: July 10, 2025; Accepted: October 16, 2025.

Computer Science and Information Systems 23(1):185–206 https://doi.org/10.2298/CSIS250831075T

An ANN-WSM Hybrid Framework for Sustainable Personalized Learning in Smart Education

Christos Troussas, Akrivi Krouska, Phivos Mylonas, Cleo Sgouropoulou, and Ioannis Voyiatzis

Department of Informatics and Computer Engineering, University of West Attica
Egaleo 12243, Greece
{ctrouss, akrouska, mylonasf, csgouro, voyageri}@uniwa.gr

Abstract. Personalized learning in modern education helps create learning experiences based on learners' individual preferences. This paper presents an adaptive learning system integrated with ANNs and WSM that recommends particular learning activities to students based on their cognitive styles. The system classifies learners according to the Gregorc Mind Styles Model and, through the ANN, provides probabilities of predefined learning activities. Static expert-assigned weights in WSM finalize the recommendations with adherence to pedagogical best practices while harnessing AI-driven personalization. This implies that a fundamental way this system contributes to sustainable education is that it dynamically optimizes the recommendations on which learning activities are proposed to boost engagement, retention of knowledge, and instructional efficiency. In this way, it undertakes enabled, resource-efficient learning by correspondence of learning activities that require little, if any, mismatch with instructional materials and minimal manual intervention. The weights of WSM, provided by experts, are static constants which contribute to higher decision-making stability, simplicity of resource consumption on computations, with the strong achievement on recommendations. In the pursuit of assessing our approach, we embedded this system in an educational software platform for C++. A study involving 70 undergraduate students found significant improvements in engagement, retention, and performance. The results validated ANNs-implemented adaptive learning using the WSM-based ranking, laying the foundation to improve personal education both in terms of a well-organized and sustainable manner. The research captures the potential of learning systems driven by AI as becoming a transformative mode to deliver efficient, broad-scale, and bespoke education.

Keywords: Adaptive Learning Systems, Artificial Neural Networks in Education, Personalized Learning, Recommendations, Sustainable Education, Programming Education.

1. Introduction

Educational software has changed the way modern learners engage by rendering dynamic and scalable learning experiences [1]. Such technologies have incorporated methods through which learners can engage with educational content in other dynamic, interactive ways. Broadly speaking, this means the incorporation of multimedia elements, simulations, and adaptive assessment tools into the learning process, which is indeed a

better one over static textbooks and a one-size-fits-all type of instruction [2]. Through the interactions of educational software, many students can enjoy personalized learning as they progress with their own pace, revisit complex concepts when they deem fit, and access material that meets their specific needs [3]. Besides, by enabling greater access to knowledge, these systems also really help in overcoming social-economic or geographical or logistical yet relevant constraints in order to ensure that quality education is being accessed and rendered at a great population scale.

The most transformative moment for modern educational software is set by the integration of Artificial Intelligence (AI) that champions unprecedented personalization. Driven by AI, such systems exploit quite complex algorithms for analyzing students' interaction behavior so as to predict their learning preferences so that the instructional content could be dynamically adjusted in real-time [4]. Machine learning techniques can identify strengths and areas for improvement in an individual learner and deliver an adaptive and individualized experience. These intelligent systems cause levels of difficulty to be increased or decreased, they recommend further resources, and provide context-sensitive feedback, keeping the students interested and motivated during their learning experience [5]. Teachers will be able to transition from a static instructional approach to a data-driven and student-centered approach, supported by AI-driven insights ensuring the efficient and effective learning process.

The extent to which personalized learning systems are able to take into consideration learning styles is critical in determining their success, and these learning styles dictate how students perceive, process, and retain information [6]. In traditional forms of instruction, one common tendency is to take a one-size-fits-all approach underestimating the cognitive differences amongst the learners. Such underestimations equate to inefficiencies of knowledge acquisition with poorer engagement and retention rates. The integration of various learning styles in adaptive educational software allows for the differentiation of instructional delivery according to individual cognitive preferences [7]. In this sense, students engage in learning tasks that are compatible with their preferred modes of processing information, which would improve comprehension and retention. Thus, cognitive-based personalization strategies should be known and implemented in educational software for maximum student success.

Within the growth of AI techniques, in particular artificial neural networks (ANNs), many ideas thereby revolutionized adaptive learning systems [8], [9]. Using large sets of data analyzed through ANNs, complex student-type behavior patterns are easily identified, thereby giving recommendations for tailored learning. In comparison with rule-based systems that rely on existing heuristics, ANNs are robustness when learning from student interactions, updating their recommendations in a way that increasingly conforms to the changing needs of the learning. Other AI-driven approaches, such as reinforcement learning and natural language processing, also contribute to the development of intelligent, scalable, and very responsive educational solutions. These technologies add value to real-time adaptability, ensuring that learning experience remains dynamic, effective, and personalized.

In this paper, we propose an ANN-based adaptive learning model integrated with the Gregorc Mind Styles Model in order to optimize personalized education. It classifies students based on their cognitive types and predicts the most effective learning activities for them, which contributes to achieving an efficient and sustainable learning process. In con-

trast to traditional fixed-rule-based personalization methods, the proposed model dynamically adjusts the recommendations while students interact with the system, thus leading to more accurate and data-driven learning pathways. This is further improving engagement, increasing retention rates, and creating learning environments catered toward the unique needs of individual students. The experimental evaluation indicates that this model integrates the best of AI neural networks with cognitive science learning personalization for the benefit of promoting learner-centered learning.

2. Related Work

The integration of personalized educational software has become a salient aspect of contemporary learning environments, in which the instructional content is tailored to fit the individual student needs. The aim is to provide engagement and learning outcome improvements through customized learning experiences built upon data-driven insights. A few approaches to personalization exist, from rule-based adaptive learning systems, which use predefined sets of instructional rules, to machine learning-based models, which dynamically adapt to students' behaviors and learning progress ([10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]). Such intelligent systems rely on multiple data sources to steer personalized learning pathways: prior academic performance, learning preferences, and real-time interaction history. The system adjusts for each student and provides recommendations aligned with their respective strengths and weaknesses; hence, deeper comprehension and retention become viable.

The one area where adaptive learning has made significant strides is the adoption of learning style models in educational software. Models, such as VARK (Visual, Auditory, Reading/Writing, Kinesthetic) [23] and Felder–Silverman Learning Style Model [24], develop theoretical bases to assign students to categories reflecting how they prefer to take in information. Such models have been dominating in most educational settings for overcoming engagement within an instructional methodology by matching it with their cognitive preferences [24], [25]. However, many of these models use static classifications, which assume that a student has a fixed learning style, and these will not change over time. Hence, the drawback is that reduced adaptability is restrained because they do not dynamically autoregulate based on students' constantly evolving learning behaviors and interactions with materials.

The application of artificial neural networks in educational software has potentiated dangerously adaptive learning [26]. ANNs provide an advanced computational means by ransacking wooden-sized volumes of student data, pinpointing the pattern, and forecasting the most effective learning activity possible. Whereas rule-based ones need to be strictly programmed to explicitly define instructional paths, ANNs furnish an instructor with data. These studies have shown improvement from ANN-driven adaptive systems that lead progressively to any learning, involvement of students, memory retention, and general efficiency [27], [28], [29], [30], [31], [32]. Using ANNs has allowed for a very granular and personalized approach where hidden correlations connect student performance, learning pace, and instructional preferences built on the streaming values. The infusion of deep learning technologies also packs these models tighter against the intricate gymnastics of student behavior. However, rather than merging both orientations into a unified model,

studies on intelligent learning still concentrate more on learning style adaptation through fixed rules or ANN-based recommendation systems.

In contrast to earlier hybrid approaches that combined machine-learning algorithms with multi-criteria decision-making techniques, the proposed framework introduces a cognitively grounded layer through the Gregorc Mind Styles Model (GMSM) [33]. Prior ANN–MCDA (Artificial Neural Networks-Multi-Criteria Decision Analysis) models typically relied on behavioral or performance metrics alone, whereas our method explicitly incorporates a cognitive-style classification that governs how learning activities are selected and interpreted by the ANN. The Weighted Sum Model (WSM) then serves not merely as a ranking mechanism but as a pedagogical filter that embeds expert knowledge to preserve instructional soundness. This tri-layered integration – cognitive profiling $\rightarrow$ neural prediction $\rightarrow$ weighted pedagogical optimization – constitutes a distinctive advance over prior systems by aligning algorithmic adaptivity with sustainable educational principles, minimizing redundant learning effort, and promoting resource-efficient personalization. Hence, the ANN–WSM–GMSM combination bridges psychological modeling with AI-driven adaptivity in a way not addressed by previous hybrid frameworks.

This paper strives to close this gap by implementing the integration of the Gregorc Mind Styles Model within the framework of an ANN-driven adaptive learning system. In difference from earlier models assigning fixed learning styles, our method will permit the ANNs to propose and recycle learning activity recommendations in light of real-time student engagement and their achievement in learning. With its adjustment to the changes of one's cognitive styles, our model guarantees that materials for study remain vividly relevant even after a long time has passed. The incorporation of multi-criteria decision analysis, WSM specifically in our model, further amplifies the accuracy in recommending which learning activities one should pursue.

To the best of our knowledge, no prior adaptive learning framework has explicitly integrated the Gregorc Mind Styles Model within an ANN–WSM hybrid architecture. Existing hybrid systems combining artificial neural networks and multi-criteria decision-making approaches have focused primarily on behavioral or performance-based adaptation. In contrast, the proposed model introduces a cognitively grounded mechanism that links neural prediction with pedagogical weighting, constituting a novel, data-driven personalization approach oriented toward sustainable education.

3. Gregorc Mind Styles Model incorporation

Building upon this identified gap, the present study operationalizes the Gregorc Mind Styles Model within an ANN–WSM adaptive architecture to demonstrate its practical impact on personalized and sustainable learning.

The Gregorc Mind Styles Model [33] is a cognitive framework that identifies the different ways by which a learner perceives and organizes information. In that model, one comes to understand the individuality of the learners, which is quite significant in the effective design of adaptive educational systems. In contrast to conventional models of learning styles that regard sensory modalities, Gregorc's approach acknowledges the diverse ways in which the individual learners approach and process knowledge. Analysis of such cognitive preferences would help personalize the learning experience in a way that is natural to and harmonious with the students' context of assimilating information.

Gregorc, then, recognized four different cognitive styles that portray the manner through which individuals tackle learning tasks. Concrete Sequential learners work best and on well-output, orderly set formats and instructions. Those learners would seem to perform best when a task was clearly outlined or categorized in orderly steps. Because he or she depends on a predictable sequence for teaching and learning styles, an individual best performs under these stipulated methods because of a more structured version of the other Concrete Random category or kind of learner. The Concrete Random learner, however, is more tantalized with experiential modes for learning, wherein negotiations for new ideas could be bid during experiments. Those learners would like to discover new ideas for themselves through experimentation; thus, they are apt at creative and innovative settings as well as in the solution of a problem. He or she believes that rigid formality from teaching or learning content leaves little space to roam free and become explorative in learning tasks.

Abstract Sequential learners associate themselves with logical theoretical constructs and prefer analyzing concepts in somewhat orderly fashion. Their performance in acquiring information would be enhanced through reading, structured lecture presentations, or detailed research. Effective evaluation and integration of information is their strength, hence learning in analytical environments is easy for them. On the other hand, Abstract Random learners are strong in social and relational environments. Collaborative learning environments and group discussions allow those learners to share ideas with their companions. Such students connect emotionally with a particular piece of information to remember it and are taught most likely in a person-oriented way where he or she is allowed to roam free in questions and answers.

The main addition of using the model in an AI-based learning paradigm is its particular advantage for continuous improvement-perhaps, which might not be possible through most traditional learning style models with set preferences or dimensions which don't just take into account the shifting cognitive preferences for the learner. This allows for the system to use an ANN in order to analyze behavior patterns, learning outcomes, and engagement metrics based on these decisions. The result is an iterative loop of improvement, which plays into the potential for further personalized recommendations. The Gregorc Mind Styles Model in adaptive learning not only helps with sorting learners but structurally informs the way in which learning activities should be planned. The AI-driven predictions allow instructional content to be updated continually so that learners would be engaged in the material according to the cognitive style with which they can understand most. Hence the personalization includes in its perimeters life authorizing learning for optimizing resources and thus growth of general learning efficacy. The Gregorc Mind Styles Model is a cognition-based model helping to classify individuals based on preferred ways of perceiving and ordering information. It incorporates the information of how learners actually process knowledge and interact with learning materials, making it highly relevant for any adaptive learning systems.

Gregorc's model divides learners into four different cognitive styles:

- Concrete Sequential (CS) - These learners prefer structured, step-by-step instruction and are good at organized tasks, listening to instructions, and hands-on activities.
- Concrete Random (CR) - They learn best by trial-and-error and prefer exploring new ideas by working hands-on or creatively solving problems. ·

– Abstract Sequential (AS) - These learners value logical theories and structured conceptual frameworks and are good at reading, researching, and analyzing data. ·

– Abstract Random (AR) - These learners prefer discussions, collaborative learning, and brainstorming activities while putting a lot of emphasis on group interactions and emotional attachments to the learning material.

4. ANN and WSM for Recommending Learning Activities

The recommender system for learning activities developed in this research employs an ANN along with MCDA, specifically the WSM [34]. This paradigm affords the integration of structured decision-making and adaptive learning to ensure giving the most suitable recommendations in case deciding the learning activities is concerned. The model is bound by the Gregorc Mind Styles Model and provides the basis for classifying learners to feed the ANN that will be used for producing personalized recommendations. Cognitive styles influence in each student the type and pattern of learning activities that suit the person best. For their integration within the ANN, various weights are assigned for learning styles according to that learning activity's probability of engagement and effectiveness among the many instructional recommendations that could be offered. This helps in providing dynamic alignment between the learning content and student needs, from static candidacies to a model that is adaptive and continually evolving. A Concrete Sequential learner would rather have structured, step-by-step instructional tasks such as solving problems and learning activities that ask them to go through a certain procedure. But this would usually mean they are engaged with the specific type of learning activity, albeit somewhat proportioned, where some would take priority over others. Also, an Abstract Random learner would be more or less inclined to learn through brainstorming, collaborative learning, or learning through discussions but again with some variance in intensity. The ANN will use WSM to ascertain preferences given the weights relevant to each learning activity and map said computed values to the most recommended learning tasks, as shown in Fig. 1.

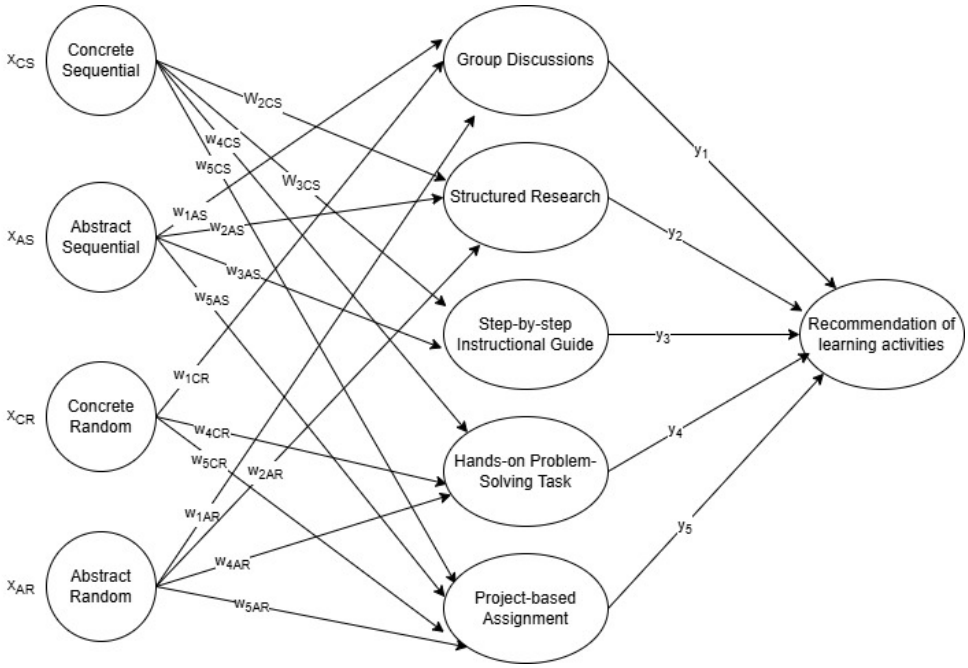


Fig. . Architecture of ANN and WSM.

The output of the presented system, based on WSM, will be as follows:

$$y_1 = \Phi(x_{AS}w_{1AS} + x_{CR}w_{1CR} + x_{AR}w_{1AR})$$

$$y_2 = \Phi(x_{CS}w_{2CS} + x_{AS}w_{2AS} + x_{AR}w_{2AR})$$

$$y_3 = \Phi(x_{CS}w_{3CS} + x_{AS}w_{3AS})$$

$$y_4 = \Phi(x_{CS}w_{4CS} + x_{CR}w_{4CR} + x_{AR}w_{4AR})$$

$$y_5 = \Phi(x_{CS}w_{5CS} + x_{AS}w_{5AS} + x_{CR}w_{5CR} + x_{AR}w_{5AR})$$

In order to ensure that the weighting system honestly reflects real-world educational needs, weight assignments of the ANN through a Delphi method were obtained from a panel of 14 experienced educators culled from university professors from diverse academic disciplines. Each of these experts had at least 12 years of teaching experience, and to them, the panel assigned the value of their insights on how the different cognitive styles influence learning preferences. Harnessing expert-guided inputs from the panel has ensured that the model's recommendations are pedagogically sound and resonate with the realities of educational practices on the ground. The ANN does output a percentage distribution of learning activities for each learner, effectively showing the inclination of varied teaching strategies to be aligned to each person's cognitive style. This means that it offers flexibility for the Instructional Designer in arranging materials into a mix

that caters to the kind of learning experiences likely to fit their learning styles. Once it's done doing all this data passage, it suggests some learning activities. As an example, Maria's case shows how this recommendation process actually works. The system has checked Maria's learning style via the recognition of her previous learning behaviors and interactions in academic life: The findings were of 70% Abstract Random and 30% Abstract Sequential. Meaning that Maria would prefer collaborative learning, brainstorming, and discussion-based activities but would also benefit from structured research assignments and analytical exercises. The expert panel assigned the following weight values: $w_1 = 0.7,\ w_2 = 0.3,\ w_3 = 0.4,\ w_4 = 0.6,\ w_5 = 0.5$.. With the use of these weights run through the WSM activation function, it becomes possible for the system to discern the most suitable learning activities for the students. In fact, according to these calculations, Maria will mostly be recommended for collaborative brainstorming and talking sessions, owing to her dominant Abstract Random learning style. The ANN constantly fine-tunes its recommendations as fresh data of interaction from students is collected. This continuous adaptive capability ensures an ever-advancing level of precision that adds more value into personalization and optimization of learning pathways as time goes on. By blending ANN and WSM, the system combines personalized instruction with a commitment to developing an efficient, engaging, and sustainable learning environment. Integration such as this limits the chances of mismatches in instructional materials, decreases cognitive overload, and guarantees that learners engage in activities well suited to their changing educational needs.

5. System Architecture and Implementation Details

The system employs a dataset comprising student interaction data, engagement with learning activities, and performance evaluations from a combination of diverse educational settings. Since the dataset contains students' answers, the time devoted to different activities, frequency concerning different instructional techniques, and self-reported preferences in agreement with Gregorc's cognitive styles, it is preprocessed to erase any inconsistencies and normalize values to reach reliability for the complete ANN training. Each student is assigned a learning Style AN in terms of associations with respect to thought processes. Such classification is used essentially as the input features for the ANN model. The ANN is a multi-layer perceptron, principally composed of an input, one hidden, and an output layer that produces recommendations for suitable learning activities to an individual learner. In terms of high performance, a justification is presented whereby a single hidden layer is taken as an assumption for the reason that if one is used, there will be efficiency in the computation such that one can model easily the relation between learning styles and the mode in which students preferred to get engaged along the process itself. The input layer receives the learning style classification; this is converted to numerical representations on which the network can act. Non-linear activation functions are used in the hidden layer to get complex relations between learning styles and engagement activities. This layer relays to the output organs a probability distribution of different activities, referring to the type of activity's efficacy and the likelihood that it will engage the particular student. . The activation function used in the hidden layer is the ReLU (Rectified Linear Unit), defined as:

$$f(x) = \max(0, x)$$

For the output layer, a softmax activation function is applied to ensure that the recommended learning activities form a probability distribution:

$$P\left(y_i\right) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Where $P(y_i)$ represents the probability of selecting learning activity i, and z_i is the corresponding output score.

An artificial neural network that receives labeled interactions from students is used to train the model; backpropagation with SGD is used to optimize the network weight values. The loss function minimizes the distance between predicted learning activity preferences and the actual engagement patterns that are observed in the historical data. A multitude of learning iterations is performed to continually refine the model's accuracy and ensure generalizability across varied student populations. Cross-validation methods are used to avoid overfitting and support the reliability of the model.

In addition to the ANN's prediction capabilities, the final recommendation process incorporates the Weighted Sum Model (WSM) to enhance decision-making. Unlike the ANN, which learns and refines student learning preferences based on historical data, the WSM operates using fixed expert-assigned weights to prioritize learning activities. As mentioned earlier, these weights were predefined by a panel of experienced educators. The assigned weights reflect pedagogical best practices and determine the importance of each learning activity type for students with different cognitive styles.

The WSM score for a learning activity S_i is calculated as:

$$S_i = \sum_{j=1}^{n} w_j x_{ij}$$

where S_i is the total score assigned to learning activity i, w_j represents the expert-defined weight assigned to criterion j, and x_{ij} is the score of learning activity with respect to criterion j. These values do not change dynamically but remain constant throughout the learning process to ensure consistency in decision-making.

From the standpoint of computation, the system is implemented with Python using TensorFlow and Keras for neural networks, with NumPy and Pandas for data preprocessing and analysis. Training of ANN occurs in a high-performance computing environment that utilizes GPUs (Graphics Processing Units) or TPUs (Tensor Processing Units) in order to accelerate matrix operations and optimize the efficiency of backpropagation. The model accommodates real-time inference for on-demand personalization of learning recommendations without introducing latency to a significant extent. This system is also scalable, intended for large-scale student datasets in online and blended educational settings. By virtue of combining ANN with WSM, the system dynamically adapts as new student records become available while constantly refining its recommendations to make learning personalized. However, WSM weights remain fixed as per predefined expert knowledge; hence, the final ranking of learning activities adheres to educationally accepted standards. By this hybrid approach, the predictive power of machine learning is combined with structured decision-making in such a way that the learner's experience remains reliable, efficient, and pedagogically sound.

6. A Sustainable Approach to Personalized Education

One of the main strengths of the proposed adaptive learning system based on artificial neural networks (ANN) is its input into ensuring that education is sustained. By sustainability in education, not just responsible resource use, but greatly adjustable long-term learning models which are favorable to various learners while maximizing educational efficiency should also be taken into account. The old one-size-fits-all instructional models are generally static, incapable of addressing individual learning needs. Such methods led to unnecessary investment in instructionally rendered time, motivating pupils to turn away from the ideal tasks and engaging in resource wastage. However, the AI-based personalized model of learning optimizes learning pathways so that students are involved with materials designed for concern effectiveness and were not wasteful of resources. The sustainability of this approach depends on less redundancy in education. With much frustration, many a learner is provided classroom material which is not commensurate with the way he/she thinks or learns best. Traditional paradigms often force students to revisit what they already know or labor through materials that do not suit their learning style. The proposed model, however, cuts these inefficiencies down via the immediate engagement of student-level data and the adaptation of recommendations in ongoing iterations. Learners are hence afforded inculcative interventions pitched with precision, which aids rapid acquisition with little intervention in revisiting past lessons. Further, education sustainability also applies to resources from technological and human intervention. The integration of ANN and WSM in the system makes possible the taking or culmination of activities in personalized learning, which would often require much of educator's attention. This means less burden on instructors and, thus, more time spent engaged in more meaningful pedagogical interactions such as activities to stimulate creative thinking as opposed to administrative ones like content assignment and tracking the performance of learners. Insights emanating from the recommendations of the system can help teachers in further adjusting their pedagogies in class, such that both AI-driven learning and human-led should act in concomitance with one another.

The system promotes sustainable e-learning from an environmental and digital resource perspective by means of minimizing unnecessary digital consumption. Many e-learning platforms unnecessarily build vast amounts of digital content, thereby making that content largely unused due to a lack of personalized targeting. This approach proposed will make sure that students get only those resources that are most relevant and beneficial, reducing excessive data storage and much of what otherwise would be applied to computation. This is also majorly contributing to reducing dropout rates since improved engagement and retention was a big problem throughout online education. A sustainable e-learning environment will keep the students motivated to go all the way through their learning journey with an applause retention and success rate at the end. Lastly, the capability of this approach to be scalable makes it highly sustainable for massive-tier educational implementation. Because the model is learning and unlearning from new student-specific interactions continuously, it gets more refined over time without requiring ongoing manual adjustments. This makes it suitable for large institutions in education, online learning services, and lifelong learning programs. With the boom of AI-centric education, sustainability has increasingly been turned into a major consideration, and such a model offers a pragmatic way to go about personalization, efficiency, and futuristic versatility.

7. Case studies

In order to address the fairness within the context of proposed ANN-WSM adaptive learning system application, we examine three cases of students who have distinct styles of learning. Each case shows in detail how the student's learning style feeds into the ANN, how WSM weights the learning activities, which ultimately results in personalized recommendations. These cases will give evidence for the system's integration into personalized learning based on cognitive preferences while continuously utilizing the already set five predefined learning activities.

7.1. Case 1: Maria – Abstract Random Learner

Maria is said to be 70% Abstract Random and 30% Abstract Sequential, so she performs best in collaborative and discussion-based learning environments while she also benefits from structured research. Her classification of learning style works as an input into the ANN, where numerical weights are given according to her distribution of preferences. Using the softmax activation function, the ANN evaluates the probability of various learning activities for Maria. The Weighted Sum Model (WSM) then fine-tunes these probabilities by assigning each of them empanelled weights by experts. The WSM function is computed as follows.

$$S_i = \sum_{j=1}^{n} w_j x_{ij}$$

where Si represents the score for each learning activity, w_j represents the assigned weight, and x_{ij} is the engagement score.

Maria's final recommendations are on a heavy emphasis on collaborative discussion and project-based assignments. This boosts her engagement, whilst also helping her develop ways of processing and synthesizing all the information and ideas through structured inquiry.

7.2. Case 2: Alex – Concrete Sequential Learner

According to Alex's cognitive classification, he is 90% Concrete Sequential and 10% Concrete Random, preferring structured, procedural tasks, benefiting from clear, well-defined learning objectives. Accordingly, the input vector in ANN responds to these preferences, enhancing the debugging methodology involving step-by-step instructional guides and structured research. The ANN processes Alex's classification and provides the WSM with an initial probability vector regarding various learning activities. With regard to the pedagogical strategies defined by the experts, the WSM model subsequently assigns the weights. Given the weight assigned to the structured learning, these traditional methods are bound to be reinforced in by way of learning.

- High weights to step-by-step instructional guides and structured research
- Low weights to collaborative and exploratory learning activities

As a result, Alex’s recommended learning activities focus on step-by-step guides and structured research, ensuring an effective and engaging experience that aligns with his learning tendencies while minimizing unstructured, discovery-based learning approaches.

7.3. Case 3: Sofia – Balanced Learner (Abstract Sequential & Concrete Random)

An abstract sequential and concrete random cognitive style assigned Sofia a 50% and 50% knowledge of mixing both logical and sequential learning and hands-on activities. This balanced profile creates a problem for adaptive learning systems because it needs to evenly distribute recommendations between theoretical and hands-on tasks. In putted with an ANN 50% toward structured research or hands-on problem-solving tasks, probability distribution appears the same. Hence, the model assigns weights to ensure diversification in terms of Sofia's strengths. Learners could pay more attention to logical learning and hands-on problems in a manner such that they would develop a good combination between analytical reasoning and creative exploration to maintain a good engagement context in the lesson given. This thus becomes an integrated strategy to guarantee that Sofia has an all-around experience-fitting outline unique to the cognitive style preferences. Table 1 illustrates the recommendations in learning activities for all the other students, supposedly showing the dynamic adaptation of the ANN and WSM concerning learning style preferences.

Table . Learning activities recommendation.

[]@ ¿p() * 0.2453 ¿p() * 0.4993 ¿p() * 0.2541@ Student Learning Style Classification Recommended Activities

Maria 70% Abstract Random, 30% Abstract Sequential Group discussions, project-based assignment

Alex 90% Concrete Sequential, 10% Concrete Random Step-by-step instructional guide, structured research

Sofia 50% Abstract Sequential, 50% Concrete Random Structured research, Hands-on problem-solving task

This case study, on the other hand, has shown how effectively the ANN-WSM model can customize itself to suit specific learning styles, while ensuring instructional content is matched to strengths in cognition. Through the ability of ANN to recognize patterns and make decisions on a framework of WSM, this system adjusts recommendations in real-time to bring about better engagement and retention in learning. This serves to illustrate the great importance of personalized adaptive learning systems to the realization of efficient, sustainable, student-centered learning.

8. Evaluation

In order to assess the effectiveness of the ANN-WSM adaptive learning system that has been proposed, the evaluation used an educational software platform specifically designed for the practical, hands-on use of C++ programming among undergraduate students. There were 70 students in the introductory C++ course and interacted with the adaptive system chosen as the testbed over the 12-week period. The objective of the evaluation was to measure the ability of the system to provide personalized learning experiences, increase student engagement, and improve learning outcomes based on their cognitive styles.

The evaluation design followed a mixed-method experimental protocol combining quantitative performance metrics and qualitative feedback. Each metric – accuracy, pre-

cision, recall, F1-score, engagement score, learning retention rate, and performance improvement – was computed directly from log-level system data. Specifically, the ANN generated probabilistic predictions of preferred learning activities, which were subsequently refined and ranked through the WSM layer using expert-defined weights. This pipeline enabled us to trace, for every recommendation, how neural predictions translated into weighted scores, ensuring transparency in how results were derived. The scope of the evaluation was limited to a 12-week controlled study with 70 undergraduate participants in a C++ programming course, designed to measure short-term learning effectiveness and user engagement within a clearly defined domain.

8.1. Experimental Setup

The evaluation was conducted in a controlled learning environment where students accessed the C++ educational platform integrated with the ANN-WSM learning recommendation system. Each student was required to complete a series of programming exercises, coding challenges, and theoretical assessments. The system continuously adjusted the learning activity recommendations based on their learning style classification and engagement patterns. At the beginning of the study, each student underwent a learning style assessment based on the Gregorc Mind Styles Model. The ANN processed these classifications and generated an initial probability distribution of suitable learning activities. The WSM model then refined these probabilities to determine the most effective personalized learning paths. Throughout the 12-week period, the system dynamically adjusted its recommendations based on student performance, engagement levels, and feedback metrics. Each student participated in three different phases:

(i) initial assessment through testing their previous knowledge of C++;
(ii) adaptive learning phase, wherein the system provided the individual with personalized learning activities; and
(iii) post-assessment evaluating profiling, engagement, and retention.

The evaluation and interpretation of the quantitative results (precision, recall, F1-score, engagement, retention, and performance improvement) were conducted by the research team based on data automatically recorded by the ANN–WSM system during the 12-week experimental period.

The system analyzed effectiveness by monitoring the time taken for exercises, exercise completion rates, quiz scores, and feedback by students.

8.2. Evaluation Metrics and Calculation of Results

The metrics reported in this section were computed based on the ANN-generated activity probabilities and their WSM-adjusted rankings, reflecting both predictive accuracy and pedagogical alignment. All performance metrics were calculated by the authors using the system's logged interaction data exported from the educational platform.

To evaluate the system's performance, we measured several key metrics:

- Accuracy of Learning Recommendations (ACC): The proportion of correctly assigned learning activities that aligned with students' cognitive preferences.

- Precision (P): The fraction of recommended activities that were actually useful and engaging for students.
- Recall (R): The proportion of relevant learning activities successfully recommended to students.
- F1-score: The harmonic mean of precision and recall, balancing accuracy and completeness.
- Engagement Score (ES): Measured as a weighted combination of time spent on activities, completion rates, and active participation in recommended tasks.
- Learning Retention Rate (LRR): The percentage of concepts retained over time, based on quiz scores after different intervals.
- Performance Improvement (PI): The increase in post-test scores compared to pre-test scores.

To enhance methodological transparency, each metric was computed directly from the system's logged interactions, linking ANN predictions and WSM-based recommendations. The ANN first generated a probability distribution over potential learning activities for each student. The WSM then refined these probabilities using expert-assigned weights to yield the final ranked recommendation list. Accuracy, precision, recall, and F1-score were calculated by comparing these final ranked recommendations to the ground-truth activities that each student selected and reported as most engaging during the experiment. Engagement Score (ES) was derived as a weighted composite of task completion rate, time on task, and number of voluntary interactions. Learning Retention Rate (LRR) measured performance stability by comparing quiz scores across three assessment intervals. Performance Improvement (PI) represented the percentage increase between pre-test and post-test scores. Together, these metrics capture both algorithmic performance and pedagogical impact, providing a comprehensive evaluation of the ANN–WSM system.

The performance of the adaptive system was calculated using the following formulas:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where:

- TP (True Positives) represents correctly recommended activities that aligned with student preferences and resulted in improved engagement.
- FP (False Positives) represents recommended activities that did not align with student learning preferences.
- FN (False Negatives) represents relevant activities that were not recommended to the student.

Table 2 presents the aggregated results from the evaluation.

Table 2. Aggregated results from the evaluation.

[]@ ¿p() * 0.2449 ¿p() * 0.4986@ Metric

Example

Accuracy of Learning Recommendations 87.5%
Precision 85.2%
Recall 81.7%
F1-score 83.4%
Engagement Score 82.3%
Learning Retention Rate 79.6%
Performance Improvement 18.2% increase

Note: Accuracy, precision, recall, and F1-score were derived from the ANN–WSM combined predictions; Engagement Score (ES), Learning Retention Rate (LRR), and Performance Improvement (PI) quantify behavioral and cognitive outcomes.

Results reveal that the ANN-WSM adaptive system achieved a very high sense of accuracy (87.5%) in facilitating the customization of learning recommendations to the needs of the individual student. While precision at 85.2% attests to the system's capacity for assigning meaningful and pleasurable learning activities, recall, at 81.7%, illustrates that significant intent activities were generally recommended. The F1 score, at 83.4%, combines the two and indicates an equal strength of the model in proposing anxiety concurrent with the varied needs of the learners. Engagement score: 82.3%, which stands for the uninterrupted ventures of the pupils in the recommended activities confirms the system's capability to elevate engagement and interactivity with the learning material. The learning retention score of about 79.6% indicates the styled learning paths or processes allow one to better retain knowledge over time. In addition, the performance improvement (18.2%) from pre-test to post-test scores attests that the adaptive learning proposition enhances the students' C++ programming prowess.

8.3. Discussion

The results, however, suggest that an important finding from the study was the relationship between adaptive learning pathways and knowledge retention, as longitudinal test tools such as surveys and formative and summative assessments show long-term retention rates to be better for students who engaged with content congruent with their style of learning. From these few findings, cognitive-based personalization may be the serviceable ingredient toward sustainability of learning outcomes over time.

These results are consistent with findings from related adaptive learning studies employing artificial neural networks or hybrid multi-criteria models. For example, Okewu et al.[8] and Borhani & Wong [9] also reported precision and recall values above 80%, indicating that ANN-based personalization can accurately model learner preferences. Similarly, Troussas et al. [29] achieved comparable gains in engagement and retention through hybrid inference mechanisms. Our observed 18.2% performance improvement aligns with the trend that ANN-driven adaptation enhances programming proficiency while maintaining pedagogical soundness. Collectively, these consistencies reinforce the validity of our ANN–WSM framework and confirm that its outcomes correspond closely with those of established research in adaptive educational systems.

Such a positive story does not come without the challenges and limitations appearing in the study. Some students had difficulty with the recommendations provided by the sys-

tem from the very beginning and had to be manually worked on by instructors to adjust their learning paths. Moreover, the model's functionality very much relied on the quality of the initial learning style assessment, which only underscores the need for further refinement in accuracy of classification. Future developments may consider continuously recalibrating learning style classifications to adapt to changes in student behavior and preferences over time. Another important aspect, in this case, is the further scalability of the system in the field, which in comparison to our research with the ANN-WSM methodology conducted on the controlled group of 70 students requires more optimization before it can be applied to larger platforms for education without increasing computational costs during real-time adaptivity. Future analyses should explore developing methodologies to simulate processing efficiency and improve the system's preparedness toward different populations of students.

This study further demonstrates the effectiveness of AI-led adaptive learning systems in higher education contexts. The ANN-WSM model provides a data-oriented, sustainable approach to personalized teaching, which will help boost student engagement, retention, and overall success in programming education. These findings point to a manageably extended research and development program for AI-based learning frameworks so that future learning environments remain flexible, inclusive, and successful.

9. User Experience and Feedback in Adaptive Learning

The efficiency of an adaptive learning system, thus, is not merely a function of how well its algorithms perform, but also of how well students and instructors engage the system and how satisfied they feel about it. How do students use the system? What other aspects about it are perceivable by them? What do they do from time to time that creates any sort of problems for them? These different aspects must be considered core in the model of learning, hence informing improvements with its design progress. This study used qualitative and quantitative feedback from students and instructors gathered throughout the 12-week evaluation period to assess the influence of ANN-based learning recommendations on engagement, motivation, and learning outcomes.

9.1. Student Feedback and Learning Experience

During the evaluation, students provided feedback on how well the system attunes itself to their individual learning styles, perceived usefulness of recommended activities, and engagement with C++ programming due to its adaptive nature. Surveys and structured interviews were carried out at the beginning, midpoint, and the end of the evaluation interval, giving students opportunities to reflect upon their experience throughout the evaluation period.

Most students reflected positively on the personalized recommendations, stating that, when compared to traditional static materials, the system offered them a more engaging alternative. Particularly there were many students classified as Abstract Random or Abstract Sequential learners who comprehended the process of learning theoretical concepts through collaborative recommendations and discussions due to interactive experiences imposed on the learning process. Even Concrete Sequential learners acknowledged the

utility of the structured exercises where they could go through coding tasks step-by-step aligned with their cognitive style.

On the other hand, for some students, open recommendation auto-processes were difficult in the beginning, mostly for those used to linear learning backed by a teacher. They stated the absence of a human teacher's explicit guidance and reported first periods of trial and error for deciding whether or not to place confidence in the recommendations made by the machine. However, by the middle of the study, other students had adapted to the system and were beginning to actively engage with the personalized learning activities, which meant that confidence in the recommendations given by AI was improving with prolonged using time.

On average, students offered the system an overall satisfaction level from 1 (very dissatisfied) to 5 (very satisfied) for a score of 4.3 out of 5, indicating that most of the learners found the system helpful in building their understanding of C++ concepts.

9.2. Instructor Feedback and Observations

Instructors participating in the study described very useful opinions about the real impact of ANN-WSM system in students progress and interactions in the classroom. Most of them mentioned the advantages of the adaptive system in that they were a great deal less involved in the manual assignment of activities, which allowed teachers to focus on helping students having problems with specific programming concepts.

Other major points noticed by the instructors were students' better autonomy and functioning on problem-solving skills. Since the system tailored activities to each student's learning style, the students carried out their learning increasingly by themselves because of less need for them to seek an instructor for repetitive clarifications. Participation of the students in the classrooms increased, according to the instructors, particularly among those hesitant to speak in discussions about coding.

However, some of the instructors noticed how a small group of the students would possibly need further support to adapt to personalized learning. A few students were not eager to diverge from conventional teaching; such students asked for assurance that AI-driven recommendations were valid. Some suggested that the installation of explanatory contents into the system would even help by providing brief reasons as to why specific activities were recommended, thereby increasing students' trust and confidence in the model.

9.3. Challenges Faced by Students and Instructors

Despite substantial gains in engagement and personalized learning by the ANN-WSM system, a few evaluation challenges did emerge. One of the most common struggles voiced by students is some rigidity of the recommended learning activities. This was particularly noted for those instances when the learner would prefer additional exploratory avenues beyond those mapped onto that individual's classified learning style. This became particularly visible among the blended cognitive style students, where, on occasions, the weights-fixed WSM approach fell short in capturing the subtle crests and troughs sometimes apparent in their learning preference over time. Another issue consisted of the reliance on the suggested activity types, in that some learners became dependent on activities prompted by their preferred style classifications and shied away from different

learning methods. According to the teachers, though there was high student commitment, certain students tended to avoid challenging themselves by going for something outside their main classification; this, they felt, ran the risk of limiting their exposure to diverse avenues for solving problems. That said, from a technical viewpoint, the real-time recommendation updates sometimes produced some activity recommendations that were redundant. In other words, rapid pacing through material enabled students to continue getting the same suggestions for activities. This shows the need for building in dynamic recalibration mechanisms for further versions of such systems-ones that would permit the model to adjust based on real-time learning behavior and not just for initial classification.

Incorporating brief explanatory content that clarifies why specific learning activities are recommended could further enhance user trust and engagement, especially for students accustomed to traditional instructor-led methods. Such transparency would strengthen acceptance of AI-driven personalization by making the recommendation process more interpretable and pedagogically reassuring.

9.4. Correlation Between User Satisfaction and ANN-Based Recommendations

To assess quantitatively how user satisfaction is related to the effectiveness of an ANN-based recommendation, statistical analysis was performed comparing engagement levels, learning performance, and satisfaction scores as reported. With a Pearson's correlation coefficient of 0.82, the relationship between engagement levels and satisfaction scores is deemed very strong evidence of a positive relation; that is, students who engaged more in the recommended activities were more likely to find it useful afterward. In terms of performance data, those students who did engage with their assigned learning activities showed an average score improvement of 18.2% from pre-test to post-test assessments, compared with only 10.4% improvement by students who generally failed to maximize their use of the personalized recommendations. These results confirm the ANN-WSM system not only helps students become more involved but also enhances learning performance, reaffirming its efficacy as an adaptive learning tool.

10. Conclusions

In this study, an adaptation learning system using artificial neural networks (ANNs) and the Weighted Sum Model (WSM) for a personalized and sustainable educational approach has been developed based on the Gregorc Mind Styles Model. The system will dynamically change learning activity recommendations based on cognitive styles, engagement patterns, and becoming new learning behaviors. The evaluation shows the advantages of the ANN-WSM that enhance learning personalization, encouraging student engagement, knowledge retention, and overall academic performance. The high accuracy of recommendations on learning material and the great uptick in performance for students prove that AI-powered adaptive learning systems can bring much to educational life, particularly in the arena of technical disciplines such as programming education. The results show that cognitive adaptation increased engagement and motivation, which allowed students to interact with content corresponding to their learning styles. The personalization of instructional material fits the way students advance through learning, taking into account their cognitive needs, allowing learning at a speed that maximizes the synergy between

knowledge about theory and hands-on experience. In addition, the dynamic nature of the system ensures that learning recommendations keep pace with learners by adapting itself according to learning progress, feedback, and changing cognitive preferences with time. A robust and scalable means of influencing learning intersubjective dynamics is enabled by the integration of ANNs for recognizing patterns and WSM for multi-criteria decision-making.

These notwithstanding, various limitations should be acknowledged: the overall efficacy of the system is strictly bound to the initial learning style classification system, which may not always adequately address the whole range of a student's cognitive preferences. Implementing continuous recalibration of learning style assessments could help the system adapt to evolving learner preferences over time, ensuring that recommendations remain aligned with actual behavioral and cognitive changes. While the Gregorc Mind Styles Model provides a good framework for classification, human cognitive processing is nuanced, and this makes individual learning styles dynamic and susceptible to change over time. Future works could add a continuous learning style flexibility mechanism whereby the system improves the historical classification system over real-time use and on the basis of actual student engagement patterns. Another area of concern is that of scalability and computational efficiency. Whereas the framework has been shown to have good classification accuracy within a controlled testbed of 70 students, the deployment of any such system onto a larger scale like Massive Open Online Course (MOOCs) requires more optimization. The computational scheme of real-time ANN inference and WSM decision-making may inject latency into running the system with thousands of learners at once. Future works should explore distributed computing techniques, cloud-based AI deployment, and model compression strategies to smoothen and scale performance in more extensive academic settings. Although the initial rates of student engagement and retention have considerably improved, the study on long-term efficiencies of personalized recommendation still needs to be done. Conducting longitudinal studies to evaluate the long-term retention of knowledge and the sustained effectiveness of personalized recommendations would further strengthen the evidence base of the proposed model. Such studies could reveal how adaptive learning pathways influence learning persistence, motivation, and performance over extended academic cycles. The 12-week evaluation period gave proper insight into very short-term learning improvements; however, future research should address the effects of the system through extended academic cycles, looking directly into long-term retention of knowledge, responsiveness of different subjects, and overall stream of success into the following classes. Expanding the study to include diverse subjects, educational contexts, and learning domains would provide a more comprehensive understanding of the model's effectiveness and ensure its applicability beyond programming education. Finally, it could expand the dataset to technologically related disciplines apart from C++ programming.

Future work should explore integrating reinforcement learning with real-time feedback loops to enhance decision-making in the ANN-WSM framework. Such mechanisms would enable the system to learn continuously from student feedback and improve its recommendations dynamically. In this direction, future iterations could also adjust the WSM weights based on ongoing learner interactions, allowing the model to balance expert-defined pedagogical priorities with data-driven evidence. This adaptive weighting would refine personalization beyond static expert inputs. Moreover, incorporating multi-modal

learning data – such as eye-tracking, keystroke dynamics, or speech analysis – could further deepen insights into engagement and cognitive load. Overall, these extensions would make the ANN-WSM model more responsive, efficient, and scalable, advancing intelligent personalized education across diverse learning contexts.

References

1. David Mhlanga. Digital transformation of education, the limitations and prospects of introducing the fourth industrial revolution asynchronous online learning in emerging markets. *Discov Educ*, 3:32, 2024.
2. A. L. Kempe and Å. Grönlund. Collaborative digital textbooks – a comparison of five different designs shaping teaching and learning. *Educ Inf Technol*, 24:2909–2941, 2019.
3. S. Mishra, S. Dubey, M. Sharma, and A. Shukla. Revolutionizing education: The impact of ai-enhanced teaching strategies. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 12(9):9–32, 2024.
4. W. Strielkowski, V. Grebennikova, A. Lisovskiy, G. Rakhimova, and T. Vasileva. Ai-driven adaptive learning for sustainable educational transformation. *Sustainable Development*, 1:1–27, 2024.
5. S. Zharmukhanbetov and C. P. Singh. Enhancing flipped classroom engagement and personalized learning through ai-powered adaptive content delivery. In *3rd International Conference on Technological Advancements in Computational Sciences (ICTACS 2023)*, pages 1411–1416, Tashkent, Uzbekistan, 2023. IEEE.
6. P. Chuaphun and T. Samanchuen. Exploring success factors and relationships in virtual learning using ism and fuzzy micmac analysis. *Heliyon*, 10(7):e28100, 2024.
7. Y. Yousaf, M. Shoaib, M. A. Hassan, and U. Habiba. An intelligent content provider based on students' learning style to increase their engagement level and performance. *Interactive Learning Environments*, 31(5):2737–2750, 2021.
8. E. Okewu, P. Adewole, S. Misra, R. Maskeliunas, and R. Damasevicius. Artificial neural networks for educational data mining in higher education: A systematic literature review. *Applied Artificial Intelligence*, 35(13):983–1021, 2021.
9. K. Borhani and R. T. K. Wong. An artificial neural network for exploring the relationship between learning activities and students' performance. *Decision Analytics Journal*, 9:100332, 2023.
10. M. Mejeh and M. Rehm. Taking adaptive learning in educational settings to the next level: leveraging natural language processing for improved personalization. *Education Tech Research Dev*, 72:1597–1621, 2024.
11. R. Pelánek, T. Effenberger, and P. Jarušek. Personalized recommendations for learning activities in online environments: a modular rule-based approach. *User Modeling and User-Adapted Interaction*, 34:1399–1430, 2024.
12. C. Halkiopoulos and E. Gkintoni. Leveraging ai in e-learning: Personalized learning and adaptive assessment through cognitive neuropsychology—a systematic analysis. *Electronics*, 13(18):3762, 2024.
13. C. Papakostas, C. Troussas, A. Krouska, and C. Sgouropoulou. A rule-based chatbot offering personalized guidance in computer programming education. In A. Sifaleras and F. Lin, editors, *Generative Intelligence and Intelligent Tutoring Systems*, volume 14799 of *Lecture Notes in Computer Science*. Springer, Cham, 2024.
14. C. Yang, F.-K. Chiang, Q. Cheng, and J. Ji. Machine learning-based student modeling methodology for intelligent tutoring systems. *Journal of Educational Computing Research*, 59(6):1015–1035, 2021.

15. S. Oeda and S. Hasegawa. A student modeling method combining deep learning and forgetting models with knowledge tracing. In *Procedia Computer Science*, volume 225, pages 1191–1200, 2023.
16. J. Essington. Review of learning styles, classroom instruction, and student achievement. *Educational Review*, 30, 2023.
17. M. A. Molina-Cabello, K. Thurnhofer-Hemsi, D. Molina-Cabello, and E. J. Palomo. Are learning styles useful? a new software to analyze correlations with grades and a case study in engineering. *Computer Applications in Engineering Education*, 31:537–551, 2023.
18. M. A. Molina-Cabello, J. Serrano-Angulo, J. Benito-Picazo, and K. Thurnhofer-Hemsi. Grades and learning styles: New software for analyzing correlations in a longitudinal study across different branches of knowledge and years. *Applied Sciences*, 14(24):11941, 2024.
19. C. Troussas, K. Chrysafiadi, and M. Virvou. Personalized tutoring through a stereotype student model incorporating a hybrid learning style instrument. *Education and Information Technologies*, 26:2295–2307, 2021.
20. Y.-H. Hung and R.-I. Chang. Investigating the influence of students' learning-style preferences on user intentions regarding illustration software. In *10th International Conference on E-Education, E-Business, E-Management and E-Learning (IC4E 2019)*, pages 179–183, New York, 2019. ACM.
21. M. Hidalgo, K. Rodriguez, and H. Astudillo. Mapping kolb's learning style to roles in software development team. In *L Latin American Computer Conference (CLEI 2024)*, pages 1–9, Buenos Aires, Argentina, 2024. IEEE.
22. N. Waibel, Y. Sedelmaier, and D. Landes. Using learning styles to accommodate for heterogeneous groups of learners in software engineering. In *IEEE Global Engineering Education Conference (EDUCON 2020)*, pages 819–826, Porto, Portugal, 2020. IEEE.
23. N. D. Fleming. *Teaching and Learning Styles: VARK Strategies*. Neil Fleming, Christchurch, 2001.
24. R. M. Felder and L. K. Silverman. Learning and teaching styles in engineering education. *Engineering Education*, 78(7):674–681, 1988.
25. D. A. Kolb. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall, Englewood Cliffs, NJ, 1984.
26. S. J. Lee and K. Kwon. A systematic review of ai education in k-12 classrooms from 2018 to 2023: Topics, strategies, and learning outcomes. *Computers and Education: Artificial Intelligence*, 6:100211, 2024.
27. Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, Y. A. Alsariera, A. Q. Ali, W. Hashim, and S. K. Tiong. Toward predicting student's academic performance using artificial neural networks (anns). *Applied Sciences*, 12(3):1289, 2022.
28. S. Aghaei, Y. Shahbazi, M. Pirbabaei, and H. Beyti. A hybrid sem-neural network method for modeling the academic satisfaction factors of architecture students. *Computers and Education: Artificial Intelligence*, 4:100122, 2023.
29. C. Troussas, A. Krouska, and M. Virvou. A multilayer inference engine for individualized tutoring model: adapting learning material and its granularity. *Neural Computing and Applications*, 35:61–75, 2023.
30. A. Zakaryan. Application of artificial intelligence (neural networks) in education. *Main Issues of Pedagogy and Psychology*, 8(1):78–87, 2021.
31. H. Chavez, B. Chavez-Arias, S. Contreras-Rosas, J. M. Alvarez-Rodríguez, and C. Raymundo. Corrigendum: Artificial neural network model to predict student performance using nonpersonal information. *Frontiers in Education*, 8:1171995, 2023.
32. P. L. Bhari. An intelligent educational system driven using ann. In *Recent Advances in Sciences, Engineering, Information Technology & Management*, volume 2782 of *AIP Conference Proceedings*, page 020054, Jaipur, India, 2023. AIP.
33. A. F. Gregorc. *The Mind Styles Model: Theory, Principles and Practice*. Gregorc Associates, Inc., 1998.

34. E. Triantaphyllou. *Multi-Criteria Decision-Making Methods: A Comparative Study*. Kluwer, 2000.

Christos Troussas is Assistant Professor in the Department of Informatics and Computer Engineering at the University of West Attica. He holds a PhD in Informatics and has extensive experience in personalization systems, human–computer interaction and AI-driven technologies. According to the Stanford University ranking, he has been repeatedly listed among the world's top 2% of scientists.

Akrivi Krouska is Assistant Professor in the Department of Informatics and Computer Engineering at the University of West Attica. She received her PhD in Informatics from the University of Piraeus and has published widely on AI, multimedia applications, adaptive software and AR/VR systems.

Phivos Mylonas is Associate Professor in the Department of Informatics and Computer Engineering at the University of West Attica and earlier served senior research roles at NTUA. His research spans multimedia analysis, user modelling and intelligent systems.

Cleo Sgouropoulou holds a Diploma and a PhD in Electrical and Computer Engineering from NTUA and is Professor at the Department of Informatics and Computer Engineering, University of West Attica. Her research interests include software engineering, interoperability frameworks and learning technology systems.

Ioannis Voyiatzis is Head of the Department of Informatics and Computer Engineering at the University of West Attica. His research focuses on computer systems, data analytics, and software technologies.

Received: August 31, 2025; Accepted: November 20, 2025.

Computer Science and Information Systems 23(1):207–230 https://doi.org/10.2298/CSIS250908006P

AI-Powered Virtual Assistants in Education: A Systematic Review of Applications and Implications (2014–2025)

Eleni Papachristou, Christos Troussas, Akrivi Krouska, and Cleo Sgouropoulou

Department of Informatics and Computer Engineering, University of West Attica
Greece
{epapachristou, ctrouss, akrouska, csgouro}@uniwa.gr

Abstract. The integration of Artificial Intelligence (AI) in education has revolutionized both learning and teaching processes, with Virtual Assistants (VAs) emerging as one of its most impactful applications. This systematic review synthesizes empirical studies published between 2014 and 2025 that examine the role of AI-powered VAs in primary, secondary, and tertiary education. Four research questions are addressed: (1) the educational uses and implications of VAs, (2) the most frequently applied models and algorithms, (3) their contribution to learner motivation and self-determination, and (4) the benefits and challenges of instructional design supported by VAs. A total of sixty-two studies were analyzed following PRISMA guidelines. The findings indicate that VAs are most often implemented in activity-based, emotion-aware, and classroom management contexts, primarily through algorithms such as natural language processing models, convolutional neural networks, and decision trees. While VAs enhance personalization, interaction, and inclusiveness, persistent concerns remain regarding cultural diversity in emotion recognition, data privacy, and pedagogical alignment. One major conclusion of this study is that a persistent gap remains between technological innovation and educational theory, highlighting the need for Virtual Assistants to be pedagogically embedded within authentic classroom environments.

Keywords: Artificial Intelligence in Education, Virtual Assistants, Intelligent Tutoring Systems, Systematic Literature Review, Personalized Learning.

1. Introduction

Interest in Artificial Intelligence (AI)–enhanced Virtual Assistants (VAs) has grown steadily in recent years, particularly in education [1], where they aim to improve both teaching effectiveness and administrative efficiency through innovative, pedagogically sound practices [2–4]. The integration of AI into VAs has the potential to transform the educational experience, significantly contributing to improved learning performance, supporting self-regulated learning, and enhancing interaction between teachers and students [5]. These intelligent systems serve as supportive tools in the educational process, facilitating repetitive daily tasks such as classroom management, planning, and coordination, particularly in technology-enriched classrooms such as STEM (Science, Technology, Engineering and Mathematics)-focused virtual classrooms [6–8]. VAs promote a student-centered learning approach by providing individualized learning experiences and encouraging student motivation, creativity, and active engagement [9, 10].

When incorporated into conventional teaching, they can enhance classroom organization and contribute to the more efficient delivery of educational goals [11–13]. Technologically, VAs rely on advanced systems such as AI, Machine Learning (ML), and Deep Learning (DL) [14, 15]. Many applications utilize Natural Language Processing (NLP), ML algorithms, and DL techniques to enable natural communication with users, personalize content, and offer predictive capabilities [6, 16]. Others operate through predefined rule-based logic, responding to specific commands without adaptive behavior.

The use of neural networks has further expanded the capabilities of VAs. Convolutional Neural Networks (CNNs) are used for visual input analysis (e.g., emotion recognition), while Recurrent Neural Networks (RNNs) are effective for sequential data and natural language processing [17]. These technologies are already being applied across all educational levels-primary, secondary, and tertiary-particularly in subjects that involve experimentation, require safety protocols, or benefit from simulation-based instruction [13, 18–20]. Contemporary research indicates that VA integration in classrooms can lead to better study organization, increased motivation and participation, and accessible personalized learning experiences for all learners [21].

This systematic review provides an overview of empirical research on the use and importance of virtual assistants in education over the past decade. In particular, it examines:

1. **RQ1:** What are the functions, results, and implications of integrating artificial intelligence and intelligent techniques into virtual assistant applications in education?
2. **RQ2:** Which models and algorithms are most commonly used?
3. **RQ3:** What is the role of virtual assistants in student motivation and autonomy?
4. **RQ4:** What are the benefits and challenges of using VAs in classroom-supported instructional design?

One contribution of the present review is its particular targeting of Virtual Assistants as a particular subtype of AI-based education technologies, whereas most existing systematic reviews cover larger categories such as Intelligent Tutoring Systems, generic chatbots, or education-at-large with respect to AI. By specializing the focus on VAs, the current work offers an enriched mapping of the pedagogic roles, algorithmic basis, and implication for student motivation and self-regulation. In addition, the present review synthesizes not only the technology features but also the education implications, including the compatibility of VA functionality with established learning theories such as ICAP, the Constructivism, and the Zone of Proximal Development. This dual focus—technological as well as pedagogical—distinguishes the current work from previous surveys. Finally, by including literature between 2014 and 2025, the current review includes the most contemporary development covering the most recent trends such as the development for emotion-aware systems and metaverse-inspired learning environments, thereby providing an updated and comprehensive repository for researchers as well as practicing educators.

The systematic review analyzes 62 selected studies from an initial pool of 316 academic records retrieved across databases before duplicate removal, covering publications between 2014 and 2025. It evaluates the key functions, outcomes, and challenges of AI-based VAs, identifies widely used algorithms, and assesses their role in motivation and autonomy. It also discusses the critical success factors influencing their adoption in educational settings.

Findings show that core functions include performance prediction, learning monitoring, automated assessment, and adaptive support. Traditional AI techniques like NLP,

decision trees, and CNNs are prevalent, while more advanced methods such as deep reinforcement learning are less common. VAs deliver high-quality, personalized recommendations that enhance academic outcomes and accommodate individual learner needs.

The review proposes the following implications:

1. Deeper integration of educational theories (e.g., Bruner, Gagné, Vygotsky) into VA design.
2. Adoption of advanced AI to enrich human–AI interaction.
3. Increased empirical studies in real-world classrooms to assess pedagogical efficacy.

2. Methodology

This methodology section outlines the systematic process followed to identify, select, and analyse academic studies related to the use of Virtual Assistants (VAs) in primary, secondary, and higher education. The goal is to understand their role in enhancing school organization, promoting student autonomy, and improving learning outcomes.

2.1. Research methods

This literature review follows the systematic protocol proposed by Kitchenham et al. (2007), one of the most widely used standards for conducting systematic reviews in software and educational research. A SoTA (State-of-the-Art) review tool was employed for data extraction and management. The initial search was conducted in peer-reviewed journals, primarily through the Scopus database and supplemented by Web of Science, to ensure comprehensive coverage and minimize publication bias.

The review process began with the identification of 62 relevant papers. The studies were first characterized according to their frameworks and definitions of virtual assistants in education, followed by classification based on country of origin, research focus and applied technologies. The final dataset includes studies from 34 different countries, with China contributing the highest number (12), followed by the USA (10) and India (9). Pakistan, UK, and the UAE contributed three studies each, while other countries contributed one or two (see Fig. 1).

This global distribution reflects the international interest in the educational role of virtual assistants and provides a diverse basis for the analysis.

The search strategy used Boolean operators (e.g., OR, AND) to combine key terms related to education and intelligent technologies, targeting titles, abstracts, and keywords. Only peer-reviewed journal articles written in English and published between 2014 and 2025 were considered.

The selected time frame of 2014–2025 was chosen to capture the decade in which Artificial Intelligence–powered Virtual Assistants (VAs) evolved from conceptual prototypes into mature educational technologies. The year 2014 marks the appearance of the first AI-driven tutoring systems and conversational learning agents, while extending the review to 2025 ensures the inclusion of the most recent studies reflecting generative AI developments and the integration of large language models in education.

The search terms used are shown in Table 1.

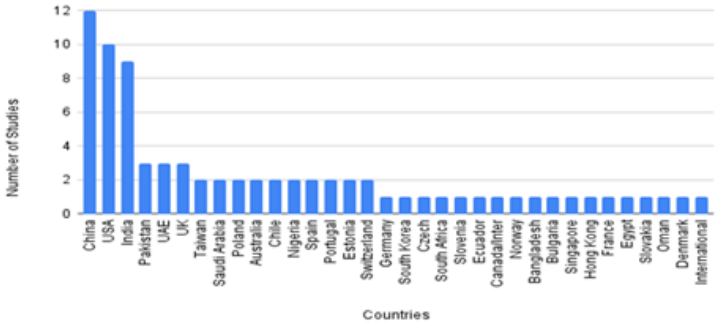


Fig. 1. Numbers of papers per country

Table 1. Key search terms used in the systematic review

Search terms
(Virtual Assistant * OR Chatbot * OR Teaching Assistants* OR Computer Simulation* OR Simulation Training * OR Deep Learning * OR Artificial Intelligence* OR Learning Experiences)
AND (Primary education* OR Secondary education* OR Higher education*)

Although some of the keywords in Table 1 (e.g., "Computer Simulation" and "Simulation Training") are broader than the term Virtual Assistant, their inclusion was intentional. The aim was to capture studies describing intelligent or interactive learning environments that integrate AI-driven agents, conversational interfaces, or simulation-based assistants often referred to under different terminologies. This broader scope minimized the risk of omitting relevant works where Virtual Assistants were embedded within simulation or training contexts but not explicitly labeled as such. All retrieved records were subsequently filtered during the screening phase to ensure that only studies meeting the Virtual-Assistant-related inclusion criteria were retained.

2.2. Inclusion and exclusion criteria

Based on the four core research questions of this systematic review, the inclusion and exclusion criteria were organized into three levels: general criteria, specific inclusion criteria, and exclusion conditions (see Fig. 2).

The inclusion and exclusion criteria were designed to ensure both relevance and methodological quality. Studies were included if they presented empirical results, conceptual models, or system implementations directly involving AI-powered Virtual Assistants for educational purposes. Excluded were non–peer-reviewed materials, opinion papers, and studies focusing solely on non-AI virtual environments or traditional chatbots without educational intent. This process ensured a coherent and high-quality dataset aligned with the objectives of the systematic review.

At the general level, studies were evaluated according to their relevance to the topic, methodological quality, completeness, independence, transparency, and reproducibility.

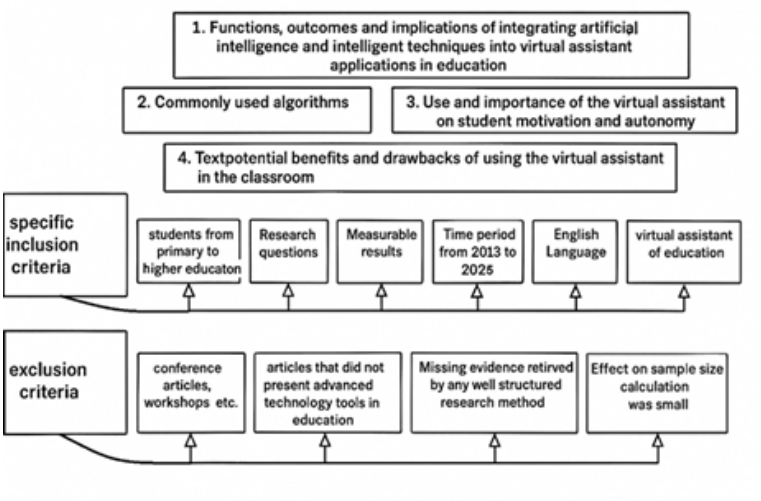


Fig. 2. Framework of Inclusion and Exclusion Criteria

Only those that demonstrated academic validity and alignment with the objectives of the review were retained.

The specific inclusion criteria required that studies:

- Focused on students in primary, secondary, or higher education
- Formulated clear research questions
- Reported measurable results
- Were published between 2014 and 2025
- Were written in English
- Focused on the use of virtual assistants in educational settings.

Examined the functions, outcomes, and implications of integrating AI into virtual assistant applications, commonly used algorithms, the role of virtual assistants in supporting student motivation and autonomy, and pedagogical benefits and drawbacks of using virtual assistants in the classroom.

The exclusion criteria included:

- Conference papers, workshop summaries, or non-peer-reviewed material,
- Studies that did not present technological tools or advanced technology applications in education,
- Articles lacking clear methodological documentation or reliable research evidence,
- Studies with minimal or negligible impact due to insufficient sample size justification.

2.3. The screening process

This screening process was designed to ensure methodological transparency and rigor in accordance with the PRISMA guidelines [22] and the systematic review framework by Kitchenham et al. (2007). The process involved three main phases: Planning, Conducting,

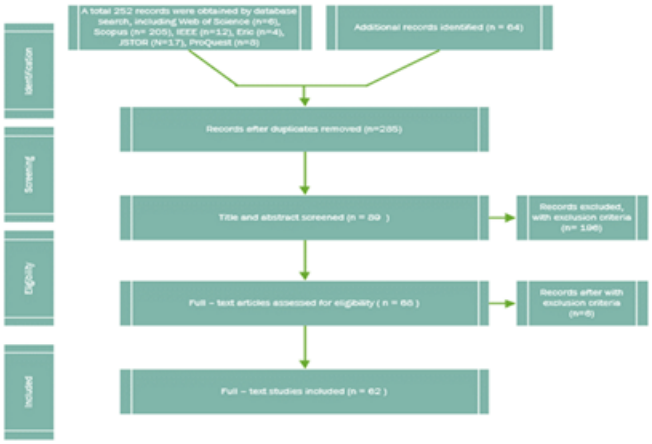


Fig. 3. PRISMA Flowchart: Study Identification and Selection Process

and Assessing. A total of 316 records were retrieved using predefined search terms across multiple databases. After removing 31 duplicates, 285 records remained. Titles and abstracts were screened, narrowing the pool to 89 articles. A full-text review was conducted on 68 articles, resulting in a final selection of 62 eligible studies. The process followed five structured steps: (1) removing duplicates, (2) screening titles and abstracts, (3) full-text eligibility review, (4) snowball sampling via Google Scholar, and (5) data extraction from selected articles (see Fig. 3).

2.4. Analysis

A series of steps were followed to achieve an unbiased search strategy [23]. The search started from the most popular databases by collecting information that was mainly related to the way people interact with computers and computer systems, and specifically to HCI issues, such as computer science/engineering and educational technology. The articles that met the inclusion criteria were analyzed using the bibliometric analysis approach [24]. The qualitative content analysis method was carried out to categorize the articles [25]. Classified information was collected from the articles and that was relevant to the research questions.

The systematic review of AIVA (Artificial Intelligence Virtual Assistant), based on a collection of 62 articles published between 2014 and 2025, provides important insights into the role played by the virtual assistant in the field of education. The early years (2014–2015) laid the foundations with studies on basic models of intelligent tutoring and the emergence of the first chatbot systems. During the period 2016–2023, research gradually intensified, with particular emphasis on affective computing and the development of learning environments incorporating emotion recognition.

The COVID-19 pandemic appears to have acted as a catalyst, as the years 2020–2022 saw a significant increase in publications. Research during this period mainly focused on adaptive learning, self-regulated support, and the integration of AI/VA into online and hybrid education. In 2023 and 2024, research interests became even more diversified, with

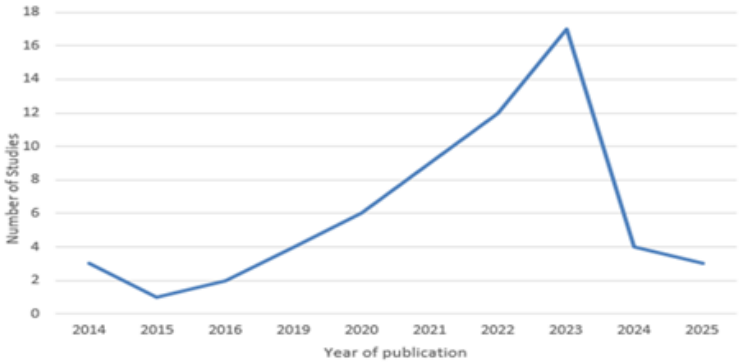


Fig. 4. Existing papers on AI/VA research over time

the introduction of virtual and augmented reality, metaverse-inspired learning environments, and the application of advanced deep learning models. Finally, the studies published in 2025 reveal a clear shift towards ethical, emotional, and inclusive applications. (see Fig. 4).

3. Research results

3.1. RQ1. What are the functions, results, and implications of integrating artificial intelligence and intelligent techniques into virtual assistant applications in education?

Recent studies examining the integration of artificial intelligence (AI) and intelligent techniques in educational virtual assistants (VAs) highlight a wide range of functions and pedagogical implications. These functions include instructional personalization, emotion-sensitive support, classroom space management, and assistance for students with special needs.

A considerable portion of the literature (n = 24) focuses on activity-based educational support, where VAs contribute to simulations, practice exercises, real-time guidance, and the enhancement of student engagement through gamified or interactive learning experiences [17, 26].

Another important group of studies (n = 19) concerns classroom space management, where VAs help improve classroom organization, scheduling, and coordination, aiming to facilitate more effective teaching and learning processes [7].

A further significant cluster of studies (n = 16) centers on emotion-based applications, in which VAs are designed to recognize and respond to students' affective states by leveraging advanced AI models (e.g., CNNs, facial recognition). These systems foster adaptive learning environments by integrating affective feedback mechanisms based on learners' emotional states [16, 27].

In parallel, a smaller set of studies (n = 3) addresses support for students with special needs, emphasizing adaptive functionalities that respond to individual learning difficulties [28–30].

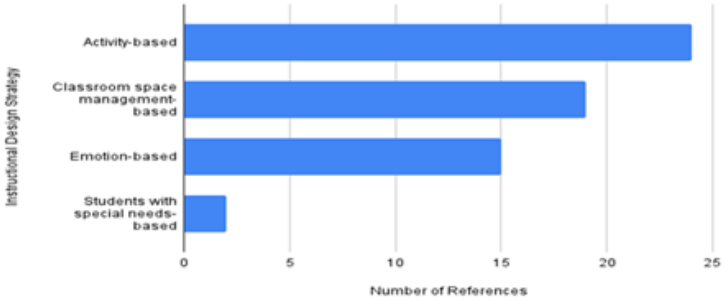


Fig. 5. Intelligent Systems design strategies in education

Overall, the selected studies were analyzed and classified into four main categories: educational subjects, emotion recognition, special needs, and classroom space management. These documents were analyzed and grouped into four main categories: educational subject, emotion recognition, special needs, and classroom space management (see Fig. 5).

The Intelligent Systems used are shown in Table 2.

Table 2. Intelligent Systems design strategies in education

Instructional design strategies	**Primary Studies**
Activity-based	Mathew et al. (2021), Pande et al. (2021), Malik et al. (2022), Halvoník & Psenak (2021), Chi & Wylie (2014), Sabharwal et al. (2023), Neuendorf & Kumar (2016), Adhikari et al. (2023), Zapata et al. (2024), Ng et al. (2024), Sunday et al. (2023), Moral-Sánchez et al. (2022), Memik & Nikolic (2021), Prinsloo et al. (2024), Ogbonnaya (2020), Hobert (2023), Cobos-Guzman et al. (2021), Alawadhi & Thabet (2023), Tarng et al. (2023), Zupic & Čater (2015), Gutierrez, Villegas-Ch & Govea (2025), Sajja et al. (2023), Senapati, S. (2022), Ng et al. (2024)
Emotion-based	Zhang et al. (2021), El-metwally et al. (2023), Zhai & Wibowo (2023), Wang et al. (2020), Li et al. (2020), Wang et al. (2014), Owusu et al. (2014), Biswas & Sil (2020), Hegde et al. (2016), Prasad & Chandana (2023), Shahzad et al. (2023), Khattak et al. (2022), Fakhar et al. (2022), Liu et al. (2019), Fang et al. (2022), Niu et al. (2021), Sehar (2021)
Classroom space management-based	Martin et al. (2024), Gonzalez (2021), Chandel et al. (2019), Farah et al. (2022), Ribeiro et al. (2019), González et al. (2022), Kuppusamy & Joseph (2021), Hu (2023), David et al. (2023), Audras et al. (2022), Stoyanov et al. (2022), Bahaddad (2025), Van Busum & Fang (2025), Lee & Kim (2023), Kadepurkar et al. (2020), Pereira et al. (2022), Premalatha et al. (2020), Quan et al. (2022), Alcala, D'Achille & Bruckman (2023)
Students with special needs-based	Balsam et al. (2019), Ricci et al. (2023), Islam et al. (2023)

The integration of Artificial Intelligence (AI) into educational Virtual Assistant (VA) applications has brought significant pedagogical and organizational changes. VAs support the personalization of learning by adapting to students' learning styles, pace, and cognitive preferences, leading to improved and learner engagement [31]. They automate assessment, track learner progress, and offer personalized feedback, thereby reducing the burden on teachers and enhancing instructional efficiency [32].

Digital tutors, such as chatbot-based assistants, have been widely adopted in programming education and STEM-related fields. These systems help students overcome conceptual difficulties, offering real-time feedback and customized explanations through natural language interaction [28]. Research shows that learners report high levels of satisfaction with chatbot tutors, particularly due to their ability to provide immediate, contextualized support during complex tasks [33].

Additionally, emotion-aware VAs can recognize learners' affective states and adjust feedback accordingly, thereby creating a psychologically supportive and inclusive learning environment. This is particularly important for students with special educational needs, where VAs can offer consistent adaptive support and help bridge instructional gaps [32].

The theoretical foundation of many successful implementations is rooted in the ICAP framework, which categorizes cognitive engagement into four levels: Interactive, Constructive, Active, and Passive. VAs that foster interactive and constructive learning (e.g., through dialogue and reflection) tend to enhance cognitive outcomes more effectively than passive content delivery [34].

Rather than replacing educators, virtual assistants are intended to complement their instructional role by automating repetitive tasks and allowing teachers to focus on mentorship, creativity, and emotional support [35]. However, successful integration depends not only on their technical capabilities but also on considerations such as user trust, data privacy, and ethical transparency.

3.2. RQ2. Which models and algorithms are most commonly used?

In recent years, Virtual Assistants (VAs) have rapidly evolved within educational contexts, with the integration of Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP), Augmented Reality (AR), and Virtual Reality (VR) technologies. These technologies have been embedded into intelligent educational systems such as chatbot tutors, intelligent tutoring systems, and recommendation engines chatbot tutors, intelligent tutoring systems, and recommendation engines to personalize instruction, assess student performance, and offer real-time feedback.

The COVID-19 pandemic further intensified the adoption of such technologies due to the increased demand for remote learning. During this period, the rise of metaverse-inspired platforms, blending VR, AR, and AI, provided more immersive and realistic educational experiences, making these tools indispensable in modern educational environments [36–38].

Virtual Teaching Assistants (VTAs) are increasingly used to reduce teacher workload while enhancing student outcomes. VTAs play a critical role in areas like exam preparation, after-school tutoring, automated grading, and performance analytics. Their impact depends on thoughtful integration that balances academic outcomes with student well-being [32].



Fig. 6. Online Platforms used in the field of Education

A variety of educational platforms now incorporate AI, VR, and AR to enhance engagement and interaction in learning environments. These platforms – such as Labster, Nearpod, ENGAGE, Oculus Education, AltspaceVR, Google Expeditions, Unimersiv, UroVers, and PedUroVerse – demonstrate how AI-driven analytics and immersive simulations are being embedded into teaching practice. They collectively illustrate the convergence of adaptive learning, experiential instruction, and data-supported feedback in modern education [33] (see Fig. 6).

Labster offers interactive virtual science laboratories for biology, chemistry, and physics, while Nearpod enhances lesson interactivity and supports real-time student assessment. ENGAGE and Oculus Education focus on the use of VR/AR for immersive instruction, and Unimersiv provides experiences in history, art, and science. Platforms like AltspaceVR and Google Expeditions emphasize social and exploratory dimensions of learning, while UroVers and PedUroVerse provide high-fidelity surgical simulations [33, 37–43].

In parallel, various development tools facilitate deep learning and ML implementation in educational contexts. Libraries like TensorFlow, Keras, scikit-learn, OpenCV, Pandas, and Numpy support the construction of scalable systems capable of personalizing instruction.

The following Table 3 gives the educational VR/AR Platforms.

Studies Emphasize the integration of STEM centers into virtual education. Platforms like ViToS enable game-based learning supported by VAs and educational robots. These environments enhance access to STEM content, encourage collaboration, and deliver real-time, personalized feedback [31, 37, 39–41]. These platforms improve the attractiveness and accessibility of educational content, particularly in STEM fields, and support collaborative resource sharing between universities and secondary schools. Embedded virtual assistants facilitate real-time feedback and enable a degree of personalization that supports differentiated instruction and student autonomy [13, 31–33, 39].

A core area of research focuses on emotion recognition using automatic facial expression recognition systems. Convolutional Neural Networks (CNNs) and variants such as

Table 3. Educational VR/AR Platforms

Platform	**Description**
Labster	Offers virtual labs in physical and scientific sciences, including biology, chemistry, and physics.
Nearpod	Enhances interactive teaching and helps teachers monitor and evaluate students (ICT-focused).
Unimersiv	Uses VR and AR to provide educational experiences in geometry, history, art, and science.
ENGAGE	Allows participation in multi-use virtual environments using VR and AR technology.
Oculus Education	Provides pre-designed or user-created VR learning environments and supports remote educational activities.
AltspaceVR	Facilitates communication and collaboration in virtual social environments using VR.
Google Expeditions	Offers VR-based educational experiences such as exploring landscapes, monuments, and ecosystems.
UroVers	Provides virtual surgical simulations to train doctors in various scenarios.
PedUroVerse	Provides virtual surgical simulations to train doctors in various scenarios.

DenseNet are commonly employed in this domain. These models process visual input to detect learners' affective states in real time, enabling emotionally responsive systems [36, 44–54].

Table 4 presents the Integrated Technologies for Machine Learning and Educational Simulations commonly used in modern research. This ecosystem includes core deep learning frameworks such as TensorFlow and Keras, alongside machine learning libraries like scikit-learn. These are integrated with the Unity game engine to create immersive virtual learning environments. To support these systems, specialized Python libraries – including Pandas for data manipulation, NumPy for numerical computation, and OpenCV for computer vision – are utilized. Together, these technologies enable seamless data processing, real-time simulation, and intelligent content personalization in educational contexts.

The growing use of virtual assistants and intelligent systems in education is reflected in four main categories of instructional design strategies (see Table 5.

These systems collectively aim to deliver personalized, adaptive, and emotionally intelligent learning support. They also relieve teachers from repetitive tasks, enhance student autonomy, and promote more inclusive, engaging, and effective learning environments [5, 13, 55, 32, 35].

The deep learning techniques used in the last decade to build automatic facial expression recognition systems with the convolutional neural network (CNN) be one of the most effective and widely used models for developing an automatic emotion recognition system [36, 44, 45, 47, 51, 54]. Also, the Dense Net (Densely Connected Convolutional Network) technique is widely used in image recognition applications and has demonstrated excel-

Table 4. Integrated Technologies for Machine Learning and Educational Simulations

Name	Description
Keras	Open source library developed by François Chollet. It is used for building and training neural network models both by beginners because of its ease of use and by advanced users because it is a powerful and versatile tool.
Unity	A game engine widely used to create educational and interactive applications. Unity supports integration with AI frameworks and APIs for immersive learning and simulation environments.
matplotlib	Open source library used to create charts, graphs and data visualizations in the Python programming language.
numpy	High performance arrays and functions library in Python. It also provides the ability to perform fast calculations on large data sets.
opencv-python	Open source library used in Python programming languages to process, analyze, recognize and manipulate images and videos.
Pillow	The Pillow library (known as PIL - Python Imaging Library) is easy to use and provides many features for creating and manipulating images in Python applications. Its functions are: opening images from various file formats, editing features (size, color, text, rotation, filters, etc.) and saving the processed image in various file formats, such as JPEG, PNG, GIF, BMP and more.
tensorflow	TensorFlow is an open source library widely used in artificial intelligence, machine learning and image processing developed by Google. It has become one of the main tools for implementing machine learning algorithms.
imutils	imutils is a utility library used by developers using Python and OpenCV. It provides utility functions to simplify and speed up image processing, performing various tasks and especially in applications such as computer vision and object recognition.
imageio	imageio is a Python library used for managing and editing images. It is used to load, save, edit images from various file formats. One of its most basic functions is managing image metadata, such as tags, user information, and more.
scikit-learn	scikit-learn (also known as sklearn) is an open source library in Python, widely used in the field of data science and machine learning for developing models, analyzing data, and solving prediction and classification problems.
Pandas	pandas is a flexible and extensible Python programming library used for data analysis and manipulation. It provides functions for data processing (remove, add, rename columns, replace, etc.), analyze graphical representations and time data. Pandas also supports importing data from various sources, such as CSV files, Excel, SQL databases, and data environments in interaction with other libraries. Finally, it allows the merging and joining of data from various sources based on common characteristics.
joblib	joblib is a Python library used to easily store and load Python objects.
datetime	datetime is a programming library used to manage dates and times.
pickle	pickle is a Python library used to store and load Python objects.
Time APIs	Time APIs (Application Programming Interfaces) are programming interfaces that provide access to time-related data and services. These interfaces are typically used to retrieve information such as the current time, dates, times, time zones, and other related information.

Table 5. Virtual Assistants and Intelligent Systems

System Category	Description
Activity-based systems	Enhance student engagement through simulations, gamification, and interactive environments (e.g., Labster, Nearpod).
Emotion-based systems	Use AI to detect and respond to student emotions, fostering supportive and emotionally responsive learning environments.
Classroom space management systems	Improve logistical organization, scheduling, and coordination within the classroom setting.
Special needs-focused systems	Provide adaptive and individualized support for learners with disabilities or learning difficulties.

lent performance in face analysis. When performing the analysis, it uses synchronized face images from a database that includes face images of different rotations, illuminations, and expressions [36, 40, 50, 54].

Table 6 presents an overview of Facial Expression Recognition (FER) techniques used in educational contexts.

Table 6. FER Techniques

Author	Technique	Dataset
El-metwally et al. [14]	DNN, MFCC, STFT	DCASE2016, CHiME-home
Zhai & Wibowo [24]	WGAN, NLP, CNN	generators/critics
Fakhar et al. [38]	CNN, HOG	FER-2013, JAFFE, CK+
Quan et al. [39]	Cross-modal attention, BERT, Bi-GRU	CMU-MOSI, CMU-MOSEI
Wang et al. [41]	CNN	JAFFE, CK+
Wang et al. [42]	SVM	JAFFE
Owusu et al. [43]	GF, MFFNN	JAFFE, Yale
Li et al. [44]	CNN	JAFFE, CK+
Biswas et al. [45]	DCT, SVM	JAFFE, CK+
Hegde et al. [46]	KLSWFDA, Gabor, SVM	JAFFE, YALE, FD
Prasad & Chandana [47]	DoG +, Median filter, EfficientNet+, YOLOv4+, DenseNet	RGB-D-T
Liu et al. [48]	CNN, Image Preprocessing	JAFFE, CK+, FER2013
Niu et al. [49]	LBP and ORB Features	CK+, JAFFE, MMI
Shahzad et al. [50]	CNN, AlexNet, VGG-16	MLF-W-FER
Fang et al. [51]	Ghost-CNN	RAF-DB, FER2013, FERPlus
Khattak et al. [52]	CNN	JAFFE, CK+, UTKFace

‘

3.3. RQ3. What is the role of virtual assistants in student motivation and autonomy?

Recent research consistently shows that Virtual Assistants (VAs) significantly enhance learner motivation, autonomy, and personalized engagement in virtual learning environments. These systems tailor educational content to student interests, offer emotionally intelligent feedback, and support self-regulated learning [18, 32, 35].

VAs foster autonomy by allowing students to control the pace and direction of their learning. For instance, they can independently navigate digital content, access on-demand assistance, and engage in exploratory problem-solving activities. This independence promotes deeper understanding and confidence [32].

Furthermore, VAs provide real-time feedback and constructive guidance that reinforces positive learning behaviors. Studies indicate that such interactions encourage students to persist in challenging tasks, improve their self-efficacy, and take ownership of their learning outcomes [5, 11, 30].

Beyond the immediate feedback provided by virtual assistants, the effectiveness of AI-generated feedback depends greatly on its design. Generic feedback is often perceived as limited and impersonal, whereas context-specific feedback is regarded as targeted, useful, and supportive. In some cases, students even view it as a "partner" in the learning process. The most significant benefits emerge when AI- and human-generated feedback are combined, as each complements the other [56].

Recent empirical evidence further supports these observations. A ChatGPT-enhanced virtual assistant (SRLbot) significantly improved students' self-regulated learning strategies, motivation, and science knowledge compared to a rule-based chatbot. Students valued the personalized recommendations and flexibility of the system, which reduced anxiety, strengthened their autonomy, and fostered more sustainable learning habits [57].

According to the ICAP framework [32], VAs support higher-order cognitive engagement (Interactive and Constructive levels), as they facilitate active dialogue, collaboration, and reflective thinking. This form of engagement correlates with stronger motivation and better academic performance.

Research in STEM and programming education environments [11, 28, 31, 40] highlights the VA's role in gamified, scenario-based, and emotionally adaptive learning activities. For example, a usability evaluation of the Imikode Virtual Reality Game showed that students reported higher engagement, improved understanding of object-oriented programming, and overall positive experiences when interacting with a VR-supported learning environment [58].

Additionally, studies show how VAs contribute to student persistence by offering consistent support even in complex learning tasks or VR-based simulations, fostering both affective and cognitive engagement [11, 26].

In summary, Virtual Assistants contribute significantly to student development by enhancing motivation through personalized and gamified learning experiences, fostering learner autonomy through self-paced and independent exploration, and increasing emotional engagement via responsive, adaptive feedback. Additionally, they support cognitive growth by aligning with ICAP-based learning behaviors and promote greater inclusion and participation, particularly among underrepresented or disengaged learners . The Fig. 7 illustrated, which maps the learning outcomes enabled by intelligent virtual assistants.

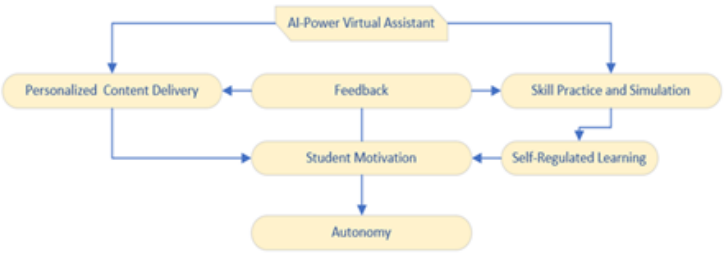


Fig. 7. Learning Outcomes Enabled by Intelligent Virtual Assistants

3.4. RQ4. What are the benefits and challenges of using VAs in classroom-supported instructional design?

In recent years, the integration of Virtual Assistants (VAs) powered by Artificial Intelligence (AI) into educational environments has brought significant shifts in instructional design and classroom dynamics. These tools aim to personalize the learning experience and adapt to students' emotional and cognitive needs [21, 35].

In particular, platforms informed by EduTech and Metaverse technologies-such as VR, AR, AI, and big data-create more engaging, immersive, and flexible learning spaces. These systems accommodate learners' individual needs, adjust content pacing, and promote inclusive participation by offering multimodal instructional formats [38]. Additionally, smart classrooms, often integrated with VAs, assist in classroom organization, energy efficiency, and attendance monitoring, alleviating teachers' operational workload [17, 21, 59].

Emotion-aware Vas, especially those that incorporate facial expression recognition (FER) and real-time sentiment analysis, enhance the learning experience by adapting their responses based on students' affective states. These systems recognize emotions such as joy, fear, sadness, or confusion, contributing to tailored feedback and improved emotional engagement [27, 44, 60]. Despite their promise, challenges remain regarding accuracy, ethical concerns, and cultural variation in emotional expression [61].

Moreover, the incorporation of dialog systems enriched with empathy, humor, and cultural sensitivity enhances student-system interaction. Such features help reduce anxiety and increase satisfaction, particularly during difficult learning tasks [62]. At the same time, the reliance on technology in education introduces valid concerns. These include digital inequity, data privacy, potential overdependence on automated systems, and the risk of dehumanizing the learning experience [31, 35].

Beyond pedagogical considerations, recent studies emphasize that user acceptance of chatbots in education depends on factors such as trust, usability, and emotional intelligence. A recent study found that integrating machine learning and semantic web technologies into chatbot design enhanced their reliability and responsiveness in educational technical support systems. Students and instructors particularly valued fast responses, emotional sensitivity, and seamless integration with widely used platforms such as WhatsApp, which collectively increased satisfaction and adoption rates [63].

While these findings highlight the conditions under which chatbots can be positively received, other research warns that the increasing reliance on AI in education also carries risks for students' well-being.

Recent research highlights that AI-powered educational decision support systems (AI-EDSS) have a profound impact on students' digital well-being. For vulnerable learners in particular, these systems may amplify existing inequalities or create new forms of disadvantage, thereby raising pressing concerns about fairness, equity, trust, and ethical responsibility in higher education [64].

In addition to these considerations, ensuring fairness in AI systems is increasingly viewed as a prerequisite for their sustainable use in education. An interactive human-in-the-loop approach has been introduced to mitigate algorithmic bias in machine learning models for undergraduate admissions. The findings show that carefully designed bias adjustment processes can reduce inequities for sensitive groups (e.g., gender, race, first-generation students) while also strengthening trust in AI-supported decision-making. This perspective underscores that fairness and user trust are interdependent, and that technical solutions must be complemented by human oversight to achieve ethical and equitable integration of AI in education [65].

The following Table 7 provides an overview of the key benefits and limitations associated with the classroom use of VAs.

Table 7. Benefits and Limitations of Virtual Assistants in Classroom-Based Instruction

Aspect	**Benefits**	**Limitations**
Personalized Learning	Adapts content to student needs and pace	
Real-Time Feedback	Encourages persistence and improves outcomes	
Emotion Recognition and Support	Supports emotional engagement and well-being	Cultural and contextual variability in emotion interpretation
Gamified and Scenario-Based Learning	Enhances motivation and immersion	Requires advanced user control skills, possible frustration
Smart Classroom Integration	Automates classroom management and saves energy	Technical complexity and infrastructure needs
Digital Equity		Limited access for underserved students
Data Privacy and Ethics		Concerns over surveillance, bias, loss of teacher role
Emotional Misinterpretation Risk		Challenges in accurately reading and reacting to emotions

Even though the investigated studies aggregate the virtual assistant contribution to personalization, motivation, and classroom management positively, critical consideration of the consistency and robustness across findings is required. One crucial observation is the rate at which most empirical evidence has been drawn on the basis of small scales or short-term investigations conducted in controlled conditions, which limits the generaliz-

ability across conditions. For instance, although emotion-aware systems using CNNs and FER approaches report promising findings for the promotion of engagement, most studies indicate that cultural diversity in the expression of emotional states undermines accuracy and inclusivity. In a similar context, activity-based VAs achieve motivation increase, but the most part of these interventions rely on the use of gamification, which possesses high concern for the long-term maintenance motivation post the novelty effect has diminished. Another major issue is the lack of comparative research. Few studies contrast the operational performance of AI-powered assistants with established teaching strategies or with other computer- or online-based tools, making it difficult to identify the unique contribution of VAs. Furthermore, where performance on the technical level is appropriately documented (e.g., recognition models efficacy, swiftness of response), fewer still examine the pedagogical richness of these interventions, for instance, the degree to which they abide by common theories of learning or have long-term impact on self-regulation abilities. This skewness suggests a disconnection between technological innovation and learning theory, and highlights the future need for VAs to be evaluated not only methodologically but also as pedagogical tools that occur organically in natural classroom ecologies.

4. Discussion

Technological advancements over the past decade have brought significant changes to the way we approach the educational process, as highlighted in this review. The findings confirm the development of modern digital assistants that support educators and provide students with access to digital learning and simulation environments, offering an interactive educational experience [14]. Realistic simulations, using intelligent learning environments supported by Virtual Assistants (VAs), enhance deeper understanding and promote interactive student engagement, having a positive impact on academic performance [13, 26, 31, 37]. This review identified recurring patterns and emerging trends in the areas of instructional practice, student motivation, affective learning, and classroom management [55]. It also recognizes challenges and limitations, and proposes directions for future development and research.

Beyond a descriptive synthesis of the results, the reviewed studies reveal notable differences in how AI-powered Virtual Assistants (VAs) are conceptualized, implemented, and evaluated. These variations often stem from divergent theoretical orientations and instructional goals. For instance, systems emphasizing adaptive feedback and personalized scaffolding reflect principles consistent with Vygotsky's Zone of Proximal Development and Gagné's Conditions of Learning, whereas game-based or exploratory assistants align with Bruner's constructivist approaches. Conversely, some studies emphasize automation or administrative efficiency at the expense of pedagogical grounding, revealing a gap between technological advancement and instructional design.

Furthermore, contradictions across studies – particularly regarding learner engagement and performance outcomes – can be explained by contextual differences such as subject area, target population, or level of human–AI interaction. Some studies report improved motivation when VAs provide socio-emotional support, while others observe diminishing returns when excessive automation reduces learner agency. This contrast underscores the importance of balancing intelligent automation with human-centered design.

Taken together, these findings advance the state of the art by revealing that the pedagogical value of AI-powered VAs depends not only on their algorithmic sophistication but also on their theoretical integration into learning processes. Bridging technical and educational perspectives thus represents the next critical step for research in this field.

The literature highlights VAs as promising tools for personalized learning and improved educational outcomes. According to theoretical frameworks such as ICAP, they support collaborative learning processes and interactive activities [34]. Beyond providing high-quality resources, virtual assistants also emotionally engage learners. The ability to interact with a learning environment that dynamically responds to students' needs enhances their interest and engagement, fostering active participation [5, 35, 58, 55].

Moreover, current applications are found to handle time-consuming and repetitive tasks, allowing educators to focus more on their pedagogically valuable work. Simultaneously, studies emphasize the support VAs offer for differentiated and personalized learning [10, 17]. Recent systems can adapt learning content to each student by providing real-time personalized feedback, thereby enhancing learning progress based on individual needs [3, 5, 11]. Additionally, research shows that when VAs offer immediate feedback, emotionally sensitive communication, and guidance at a learner-defined pace, they promote autonomy and learning interest [5, 26].

Equally notable, emotion recognition systems based on Convolutional Neural Networks (CNNs) and Facial Expression Recognition (FER) algorithms [54] extend this socio-emotional dimension by detecting learners' affective states and adjusting responses to sustain engagement [27, 36, 51]. Furthermore, game-based learning activities and the integration of augmented and virtual reality environments (VR/AR) enhance interest and active participation, particularly in STEM subjects [40, 43].

From a pedagogical perspective, studies present VAs not only as teaching aids but also as emotionally supportive tools. By taking over repetitive administrative or instructional tasks, they reduce teacher workload and increase time for student interaction, facilitating individualized support, guidance, and the cultivation of critical thinking skills [10, 11, 55].

At the same time, the adaptability of VAs paves new paths for inclusive education by supporting students with special educational needs and promoting equity and accessibility [26, 29].

Despite their many benefits, several challenges have been identified. One of the main issues is the inaccuracy in emotion recognition due to cultural differences and diversity in facial expressions [45, 36]. Additionally, there is a risk of diminished human empathy and reduced educational interaction between teacher and student [13, 35].

It is also worth noting that adopting an environment with advanced learning systems requires a high level of user control skills. This level affects students' ability to effectively engage with the educational content [18, 26]. Consequently, there is a risk of inequality in access due to a lack of technological literacy [35]. Inadequate digital skills may lead to student frustration, as they may be unable to fully benefit from the learning environment. The integration of technology in education requires not only technical competence but also a deeper understanding of issues such as personal data protection and how such systems influence learning relationships within the classroom [25, 60].

From a pedagogical perspective, these insights bridge the gap between technology and learning theory. The most favorable results found in this synthesis come from interventions offering constructive or interactive learning activities, as framed by the ICAP

model, while the principles of Vygotsky, Gagné, and Bruner remain central for adaptive feedback, sequencing, and discovery learning.

From a cognitive perspective, principles from Gagné's Conditions of Learning continue to inform feedback sequencing and instructional design in VA-supported environments. Furthermore, the inclusion of the ICAP framework illustrates how VAs can move learners away from passive reception towards constructive, active, and interactive engagement.

Very little theoretical basis in most studies, however, is an intimidating gap. Lacking an explicit pedagogical basis, VAs risk being built largely on grounds of technical feasibility rather than pedagogical imperative. Future research should strengthen this alignment between technical innovation and pedagogical theory to ensure that AI-powered Virtual Assistants enhance, rather than replace, human-centered teaching.

5. Conclusions

The findings of this review enhance our understanding of the use of advanced, modern systems that integrate technologies such as VR, AR, AI, and other related innovations. These systems simulate realistic and interactive learning experiences, which were strengthened during the pandemic and have continued to attract increasing interest and application in recent years.

This review is valuable both for developers aiming to design or implement educational virtual assistants (VAs), and for educators seeking to enhance their teaching practices through digital systems supported by VAs. Emphasis is placed on student motivation, engagement, and academic performance.

Moreover, the findings include evidence from previous experimental studies, empirical data, and valuable insights into how the integration of advanced technologies with virtual assistants can enhance learning processes. Collectively, the reviewed evidence demonstrates that when designed with pedagogical intent, AI-powered Virtual Assistants substantially enrich learning environments and foster deeper understanding, while providing educators with effective support for instruction and assessment.

This study highlights the importance of advanced technologies in education and offers clear directions for the future development of educational environments. Finally, it emphasizes the necessity of aligning technological innovation with pedagogical design and ethical responsibility to maximize educational impact. Overall, this review underscores the transformative potential of AI-powered Virtual Assistants in education while emphasizing that their sustainable integration requires ongoing collaboration among educators, psychologists, and developers to balance innovation with ethical and pedagogical integrity.

Even though the present review offers an overarching description of AI-driven Virtual Assistants for education, there are certain limitations that need to be addressed. To begin with, the analysis was limited to English articles indexed primarily in Scopus and Web of Science. In doing so, potentially relevant studies written in other languages or indexed in smaller databases might have gone unnoticed, possibly limiting the cultural and regional scope of the findings. Second, the overwhelming majority of the studies included used small sample sizes, pilot studies, or short-term interventions, which impairs the generalizability of the observed outcomes. This is especially true for studies on the recognition

of affect or gamified settings, where the novelty effect tends to transiently exaggerate motivation and engagement. Third, there was an overrepresentation in the methodology used by the reviewed studies, with many placing strong emphasis on technical performance indicators (e.g., the accuracy or the swiftness of the model) over long-term pedagogical effects. Lastly, the heterogeneity of the studies included – covering different schooling levels, settings, and disciplines – impeded the possibility of conducting direct comparisons or meta-analytic synthesis.

These limitations also open meaningful directions for future research. Future studies should prioritize longitudinal and comparative designs to evaluate the long-term pedagogical impact of AI-powered Virtual Assistants across diverse educational levels and contexts. Researchers are encouraged to enhance cross-cultural validation, particularly for emotion-aware systems, ensuring inclusivity and fairness. In practice, stronger collaboration among educators, developers, and policymakers will be essential to integrate Virtual Assistants into curricula, teacher training, and assessment frameworks while maintaining pedagogical coherence and ethical integrity.

References

1. F. Martin, M. Zhuang, and D. Schaefer. Systematic review of research on artificial intelligence in k-12 education (2017–2022). *Computers and Education: Artificial Intelligence*, 6:100195, 2024.
2. L. A. Gonzalez. Investigating the benefits of applying artificial intelligence techniques to enhance learning experiences in capstone courses. In *Proceedings of ICER 2021*, pages 398–400. ACM, 2021.
3. A. N. Mathew, V. Rohini, and J. Paulose. Nlp-based personal learning assistant for school education. *International Journal of Electrical and Computer Engineering*, 11(5):4522–4530, 2021.
4. S. Chandel, Y. Yuying, G. Yujie, A. Razaque, and G. Yang. Chatbot: Efficient and utility-based platform. In *Advances in Intelligent Systems and Computing*, volume 858, pages 109–122. Springer, 2019.
5. S. Zhang, Z. Meng, B. Chen, X. Yang, and X. Zhao. Motivation, social emotion, and the acceptance of artificial intelligence virtual assistants—trust-based mediating effects. *Frontiers in Psychology*, 12:728495, 2021.
6. J. C. Farah, B. Spaenlehauer, V. Sharma, M. J. Rodríguez-Triana, S. Ingram, and D. Gillet. Impersonating chatbots in a code review exercise to teach software engineering best practices. In *IEEE International Conference on Advanced Learning Technologies (ICALT)*, pages 76–80. IEEE, 2022.
7. J. Ribeiro, V. Alves, H. Vicente, and J. Neves. Planning, managing and monitoring technological security infrastructures. In *HELIX 2018*, volume 505 of *Lecture Notes in Electrical Engineering*, pages 10–16. Springer, 2019.
8. P. Pande, A. Thit, A. E. Sorensen, B. Mojsoska, M. E. Moeller, and P. M. Jepsen. Long-term effectiveness of immersive vr simulations in undergraduate science learning: lessons from a media comparison study. *Research in Learning Technology*, 29, 2021.
9. S. I. Malik, M. W. Ashfaque, R. Mathew, R. S. Al-Nuaimi, and A. Alsideiri. Fostering the learning process in a programming course with a chatbot. *International Journal of Online Pedagogy and Course Design*, 12(1), 2022.
10. L. A. González, A. Neyem, I. Contreras-McKay, and D. Molina. Improving learning experiences in software engineering capstone courses using artificial intelligence virtual assistants. *Computer Applications in Engineering Education*, 30(5):1370–1389, 2022.

11. R. Sajja, Y. Sermet, M. Cikmaz, D. Cwiertny, and I. Demir. Artificial intelligence-enabled intelligent assistant for personalized and adaptive learning in higher education. *Computers and Education: Artificial Intelligence*, 2023.
12. D. Halvoník and P. Psenak. Design of an educational virtual assistant software. *International Journal of Emerging Technologies in Learning*, 16(11):309–321, 2021.
13. U. Sehar, S. Kanwal, K. Dashtipur, U. Abbasi, and F. Khan. Urdu sentiment analysis via multimodal data mining based on deep learning algorithms. *IEEE Access*, 9:153072–153082, 2021.
14. F. Sh. El-Metwally, A. I. Eldesouky, N. B. Abdel-Hamid, and S. M. Elghamrawy. Optimized deep neural networks audio tagging framework for virtual business assistant. *Journal of Advances in Information Technology*, 14(3):550–558, 2023.
15. J. Lee and Y. Kim. Sustainable educational metaverse content and system based on deep learning for enhancing learner immersion. *Sustainability*, 15(16):12663, 2023.
16. R. Pereira, A. Reis, J. Barroso, J. Sousa, and T. Pinto. Virtual assistants applications in education. In *Communications in Computer and Information Science*, volume 1720, pages 468–480. Springer, 2022.
17. P. Kuppusamy and S. K. Joseph. A deep learning model to smart education system. In *Proceedings of the International Conference on Emerging Trends in Smart Computing and Informatics*, pages 78–83. Pondicherry University, 2021.
18. E. A. Ogbonnaya. The impact of technology acceptance model on the use of information and communication technology in training. *International Journal of Research and Innovation in Social Science (IJRISS)*, 2020.
19. X. Hu. The role of deep learning in the innovation of smart classroom teaching mode under the background of internet of things and fuzzy control. *Heliyon*, 9(8):e18594, 2023.
20. P. Kadepurkar, P. Dsouza, and N. Jomichan. Iot based smart classroom. *International Journal of Applied Sciences and Smart Technologies*, 3(1):35–54, 2020.
21. B. Premalatha and J. Hari Krishnan. Iot based smart classroom. *International Journal of Scientific and Technology Research*, 9(2):1644–1650, 2020.
22. S. Hobert. Fostering skills with chatbot-based digital tutors – training programming skills in a field study. *i-com*, 22(2):143–159, 2023.
23. R. Gutierrez, W. Villegas-Ch, and J. Govea. Development of adaptive and emotionally intelligent educational assistants based on conversational ai. *Frontiers in Computer Science*, 7:1628104, 2025.
24. C. Zhai and S. Wibowo. A wgan-based dialogue system for embedding humor, empathy, and cultural aspects in education. *IEEE Access*, 11:71940–71952, 2023.
25. S. Cobos-Guzman, S. Nuere, L. De Miguel, and C. König. Design of a virtual assistant to improve interaction between the audience and the presenter. *International Journal of Interactive Multimedia and Artificial Intelligence*, 7(2):232–240, 2021.
26. D. Audras, A. Zhao, C. Isgar, and Y. Tang. Virtual teaching assistants: A survey of a novel teaching technology. *International Journal of Chinese Education*, 11(2), 2022.
27. B. David, R. Chalon, and X. Zhang. Virtual assistants (chatbots) as help to teachers in collaborative learning environment. In *Lecture Notes in Networks and Systems*, volume 633, pages 135–146. Springer, 2023.
28. M. T. H. Chi and R. Wylie. The icap framework: linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49:219–243, 2014.
29. F. S. Ricci, A. Boldini, M. Beheshti, J.-R. Rizzo, and M. Porfiri. A virtual reality platform to simulate orientation and mobility training for the visually impaired. *Virtual Reality*, 27(2):797–814, 2023.
30. R. B. Islam, S. Akhter, F. Iqbal, M. Saif Ur Rahman, and R. Khan. Deep learning based object detection and surrounding environment description for visually impaired people. *Heliyon*, 9(6):e16924, 2023.

31. P. Balsam, S. Borodzicz, K. Malesa, K. J. Filipiak, and G. Opolski. Oculus study: Virtual reality-based education in daily clinical practice. *Cardiology Journal*, 26(3):260–264, 2019.
32. D. Sabharwal, R. Kabha, and K. Srivastava. Artificial intelligence (ai)-powered virtual assistants and their effect on human productivity and laziness: Study on students of delhi-ncr (india) & fujairah (uae). *Journal of Content, Community and Communication*, 17(9):162–174, 2023.
33. B. Kitchenham et al. Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report Version 2.3, Keele University and University of Durham, 2007. Accessed 15 Oct 2019.
34. D. Moher et al. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *PLoS Medicine*, 6(7):e1000097, 2009.
35. K. A. Neuendorf and A. Kumar. Content analysis. *The International Encyclopedia of Political Communication*, pages 1–10, 2015.
36. I. Zupic and T. Čater. Bibliometric methods in management and organization. *Organizational Research Methods*, 18(3):429–472, 2015.
37. A. Liberati et al. The prisma statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: explanation and elaboration. *BMJ*, 339:b2700, 2009.
38. S. Fakhar, J. Baber, S. U. Bazai, Z. Leonowicz, and S. Hussain. Smart classroom monitoring using novel real-time facial analysis. *Applied Sciences*, 12(23):12134, 2022.
39. Z. Quan, T. Sun, M. Su, and J. Wei. Multimodal sentiment analysis based on cross-modal attention and gated cyclic hierarchical fusion network. *Computational Intelligence and Neuroscience*, 2022:4767437, 2022.
40. B. Adhikari, E. J. Rapos, and M. Stephan. Simima: a virtual simulink intelligent modeling assistant. *Software and Systems Modeling*, 2023.
41. Y. Wang, Y. Li, Y. Song, and X. Rong. The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, 10(5):18, 2020.
42. H. Wang, H. Huang, and F. Makedon. Emotion detection via discriminant laplacian embedding. *Universal Access in the Information Society*, 13(1):23–31, 2014.
43. E. Owusu, Y. Zhan, and Q. R. Mao. A neural-adaboost based facial expression recognition system. *Expert Systems with Applications*, 41(7):3383–3390, 2014.
44. K. Li, Y. Jin, M. W. Akram, R. Han, and J. Chen. Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy. *The Visual Computer*, 36(2):391–404, 2020.
45. S. Biswas and J. Sil. An efficient face recognition method using contourlet and curvelet transform. *Journal of King Saud University – Computer and Information Sciences*, 32(6):718–729, 2020.
46. G. P. Hegde, M. Seetha, and N. Hegde. Kernel locality preserving symmetrical weighted fisher discriminant analysis based subspace approach for expression recognition. *Engineering Science and Technology, an International Journal*, 19(3):1321–1333, 2016.
47. S. B. R. Prasad and B. S. Chandana. Human face emotions recognition from thermal images using densenet. *International Journal of Electrical and Computer Engineering Systems*, 14(2):155–167, 2023.
48. K.-C. Liu, C.-C. Hsu, W.-Y. Wang, and H. H. Chiang. Real-time facial expression recognition based on cnn. In *International Conference on System Science and Engineering (ICSSE 2019)*, pages 120–123, 2019. Article ID 8823409.
49. B. Niu, Z. Gao, and B. Guo. Facial expression recognition with lbp and orb features. *Computational Intelligence and Neuroscience*, 2021:8828245, 2021.
50. H. M. Shahzad, S. M. Bhatti, A. Jaffar, M. Alhajlah, and A. Mahmood. Hybrid facial emotion recognition using cnn-based features. *Applied Sciences*, 13(9):5572, 2023.
51. B. Fang, G. Chen, and J. He. Ghost-based convolutional neural network for effective facial expression recognition. In *Proceedings of the 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE 2022)*, pages 121–124, 2022.

52. A. Khattak, M. Z. Asghar, M. Ali, and U. Batool. An efficient deep learning technique for facial emotion recognition. *Multimedia Tools and Applications*, 81(2):1649–1683, 2022.
53. S. Senapati. Peeking into the sophisticated world of interactive science simulations. *Resonance*, 27(11):1971–1983, 2022.
54. A. Alawadhi and R. A. Thabet. Using nearpod to promote engagement in online esl classes: A mixed-methods study in the context of higher education. In *Lecture Notes in Civil Engineering*, volume 320, pages 117–129. Springer, 2023.
55. W. Tarng, Y.-C. Su, and K.-L. Ou. Development of a virtual reality memory maze learning system for application in social science education. *Systems*, 11(11):545, 2023.
56. G. C. Zapata, A. K. Saini, A. O. Tzirides, B. Cope, and M. Kalantzis. The role of ai feedback in university students' learning experiences: An exploration grounded in activity theory. *Ubiquitous Learning*, 18(2):1–30, 2025.
57. D. T. K. Ng, C. W. Tan, and J. K. L. Leung. Empowering student self-regulated learning and science education through chatgpt: A pioneering pilot study. *British Journal of Educational Technology*, 55(4):1328–1353, 2024.
58. K. Sunday, S. S. Oyelere, F. J. Agbo, O. S. Balogun, and N. Bouali. Usability evaluation of imikode virtual reality game. *Technology, Knowledge and Learning*, 28(4):1871–1902, 2022.
59. S. N. Moral-Sánchez, M. T. Sánchez-Compaña, and I. Romero. Geometry with a stem and gamification approach: A didactic experience in secondary education. *Mathematics*, 10(18):3252, 2022.
60. S. Stoyanov, T. Glushkova, V. Tabakova-Komsalova, V. Ivanova, and L. Doukovska. Integration of stem centers in a virtual education space. *Mathematics*, 10(5):744, 2022.
61. E. Memik and S. Nikolic. The virtual reality electrical substation field trip: Exploring student perceptions and cognitive learning. *STEM Education*, 1(1):47–59, 2021.
62. K. Alcala, A. D'Achille, and A. Bruckman. The stage and the theatre: Altspacevr and its relationship to discord. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW1), 2023.
63. A. A. Bahaddad. Increasing the degree of acceptability for chatbots in technical support systems for educational sector by using ml and semantic web. *Alexandria Engineering Journal*, 116:296–305, 2025.
64. P. Prinsloo, M. Khalil, and S. Slade. Vulnerable student digital well-being in ai-powered educational decision support systems (ai-edss) in higher education. *British Journal of Educational Technology*, 55(5):2075–2092, 2024.
65. K. Van Busum and S. Fang. Interactive mitigation of biases in machine learning models for undergraduate student admissions. *AI Switzerland*, 6(7):152, 2025.

Eleni Papachristou is a PhD Student in the Department of Informatics and Computer Engineering at the University of West Attica researching the use of AI as an educational reflection tool. Her research interests lie in the integration of AI to enhance teacher-student interaction.

Christos Troussas is Assistant Professor in the Department of Informatics and Computer Engineering at the University of West Attica. He holds a PhD in Informatics and has extensive experience in personalization systems, human–computer interaction and AI-driven technologies. According to the Stanford University ranking, he has been repeatedly listed among the world's top 2% of scientists.

Akrivi Krouska is Assistant Professor in the Department of Informatics and Computer Engineering at the University of West Attica. She received her PhD in Informatics from

the University of Piraeus and has published widely on AI, multimedia applications, adaptive software and AR/VR systems.

Cleo Sgouropoulou holds a Diploma and a PhD in Electrical and Computer Engineering from NTUA and is Professor at the Department of Informatics and Computer Engineering, University of West Attica. Her research interests include software engineering, interoperability frameworks and learning technology systems.

Received: September 08, 2025; Accepted: December 14, 2025.

Computer Science and Information Systems 23(1):231–253 https://doi.org/10.2298/CSIS250915076B

Meeting Consumer Demand: Supporting Efficient Services Deployment for Cloud-Connected Devices

Iván Bernabé-Sánchez[1], Daniel Díaz-Sánchez[2], Alberto Fernández[1], Holger Billhardt[1], and Sascha Ossowski[1]

[1] CETINIA, Universidad Rey Juan Carlos
Tulipán s/n, 28933 Móstoles (Madrid)
{ivan.bernabe,alberto.fernandez,holger.billhardt,sascha.ossowski}@urjc.es
[2] Universidad Carlos III de Madrid
Avda. de la Universidad 30, 28911 Leganés (Madrid)
dds@it.uc3m.es

Abstract. The demand for digital services driven by the growth of internet-connected devices and consumer needs is constantly increasing. Therefore, the provision of services to achieve a good user experience is becoming crucial. Network Functions Virtualization (NFV) has been established as a key technology for service providers. NFV decouples network functions from hardware devices by allowing network services, called virtualized network functions (VNF), to be hosted on commodity hardware, simplifying and making service deployment more flexible. The placement of VNFs within infrastructure presents a critical technical challenge due to its direct impact on network performance, reliability, and operational costs. This article presents a service architecture that can be integrated with network service managers to identify priorities that should be taken into account when deploying VNFs or services in infrastructures. The proposed solution collects information about the network service to be deployed and analyzes and evaluates its requirements by comparing them with operating patterns. As a result, recommendations for the deployment of each service are obtained. The solution also proposes a provisioning service which, based on the generated recommendations, adapts the configuration files for deployment and connects with the configuration management tools to automate the deployment.

Keywords: VNF placement, Resource Allocation, Virtualization, Cloud Computing, Edge Computing, Network function virtualization, Service function chain.

1. Introduction

The proliferation of end-user devices and the emergence of new mobile technologies, such as smartphones, wearables, and IoT sensors—have sharply increased mobile data consumption [8]. This trend is driven by enhanced device capabilities (e.g., high-resolution displays and HD cameras) and the growing demand for immediate access to digital content. As a result, telecommunication service providers (TSPs) are under increasing pressure to adapt their infrastructures, facing challenges related to resource optimization, hardware rigidity, and inflexible control mechanisms. These factors increase capital and operating expenditures, negatively impacting profitability [35].

Despite advances in cloud computing and Network Functions Virtualization (NFV), several challenges remain. Efficient resource allocation, optimal configuration selection,

and effective distribution of virtualized network functions (VNFs) remain complex problems, particularly in heterogeneous and distributed environments that include edge computing [19, 9]. Existing solutions often optimize individual metrics, such as latency or resource use, overlooking holistic strategies that address the full complexity of these infrastructures.

To address these issues, this work introduces an architecture that automates the deployment and configuration of VNFs. The system leverages optimization techniques that consider geographic location, infrastructure compatibility, and service-specific requirements to recommend suitable deployment configurations.

The proposed approach addresses the following key challenges:

- *Optimal Configuration Selection for VNFs*: As Service Function Chains (SFCs) grow in complexity, deployment strategies must optimize resources, minimize latency, and ensure component compatibility across infrastructures.
- *Management of Heterogeneous Infrastructures*: Cloud-edge integration introduces disparities in capacity, location, and technology, complicating orchestration.
- *Automation of Resource Allocation*: Real-time resource provisioning is essential to meet fluctuating demands while preserving Quality of Service (QoS).
- *Validation in Real-World Scenarios*: Evaluation under realistic conditions such as multimedia streaming, ensures broader applicability and reliability.

This article makes the following key contributions:

- *Architecture for Efficient VNF Deployment*: A systematic approach is proposed for the automated selection and configuration of VNFs in distributed environments.
- *Optimization of Distributed Service Management*: The solution integrates edge and cloud resources to reduce latency and enhance user experience in latency-sensitive applications.
- *Automated Configuration Selection System*: A decision model evaluates infrastructure compatibility, service demands, and optimal placement to guide deployment.
- *Simulation-Based Validation*: A multimedia streaming scenario demonstrates the effectiveness of the system, using latency and performance metrics as benchmarks.

The remainder of this article is structured as follows. Section 2 discusses related work; Section 3 presents the proposed architecture; Section 4 explains configuration selection mechanisms; Section 5 details the implementation and results and Section 6 provides conclusions and outlines future research directions.

2. Related work

A multitude of papers can be found in the literature proposing different strategies to decide the location of VNFs. This decision process takes into account relevant factors such as the characteristics and resources of the available nodes [20], the capacities of the VNF nodes to be deployed [31], the traffic and capacity of the network in which they are to be deployed, the energy consumption and management of the systems [27, 25] or even the use of data centers located in the cloud or at the edge of the network [24]. All these factors condition the techniques and strategies for the optimization of VNF deployment systems, which has generated different solutions to address the problem. For example, works such

as [28] and [4] focus on optimizing latency when VNFs are deployed. Other works such as [34], [1] and [37] take into account network traffic to decide the placement of VNFs. In [16] customer satisfaction is taken into account as the main objective to decide where to deploy VNFs. In [4] and [11] the authors also consider the costs associated with the deployment of VNFs.

Other works are oriented towards managing the deployment of services on top of compute nodes located at the fog layer and taking into account requirements related to deployment cost, required computation, or energy consumption. These works consider services that can perform all kinds of tasks, from virtualised network functions (VNF) to other types of tasks related to various domains.

In [29], a solution for resource management is presented in the fog layer that integrates fog computing, machine learning, and context-aware computing to provide efficient resource management. [29] takes into account compute, buffer and disk space requirements. In [13], a method is proposed for selecting edge nodes in the smart manufacturing domain to minimize the total cost of edge node deployment, network delay, packet loss, and energy consumption. To this end, they proposed an algorithm based on game theory to select the most appropriate nodes to deploy services.

In [23] a recommender framework is proposed to decide whether to run services on nodes located on the edge or in a cloud datacenter.

Finding suitable positions to place VNFs or services in a way that satisfies the requirements is a critical problem. It has been shown that this problem is NP-hard [3] so, due to its difficulty, finding the optimal solution for VNF Placement (VNF-P), especially in large-scale network scenarios, is impossible. To address the problem of VNF placement in small- to moderate-scale environments, VNF-P can be formulated as Integer Linear Programming (ILP) and Mixed ILP (MILP) problems. [2] and [15] use ILP and MILP to calculate the optimal placement. In [2], ILP is used with the objective of finding the optimal placement of an Evolved Packet Core (EPC) service considering latency limits in a mobile network scenario. In [15] the authors also study the VNF placement problem for big data processing by providing a MILP model where they use a simple relaxation-based heuristic algorithm to minimize the overall communication cost. However, ILP models are unsuitable for large systems and only apply to scenarios where all variables are integers.

In order to help reduce the energy consumption of consumer electronics, [36] presents a solution to address Mixed Integer NonLinear Programming (MINLP) problems of bandwidth and computational resource allocation by carefully considering the trade-off between computational complexity and implementation efficiency.

Most of the works take the optimization of costs, energy, network traffic, available resources in the nodes, and latency as the main factors. However, we consider that it is also important to decide on other relevant aspects, even before launching the planning processes for the deployment of VNFs in the works discussed. For example, some solutions will provide us with the ideal location for a VNF based on network traffic or energy expenditure. However, it is important to assess the service requirements that will directly influence its operation. Once the service requirements have been identified, we propose to launch the optimization process for the deployment of VNFs. However, unlike these approaches, our proposal does not focus on optimizing the runtime performance of services or applications. Instead, we emphasize a preliminary but essential phase: identifying the service requirements that will directly influence their operation before initiat-

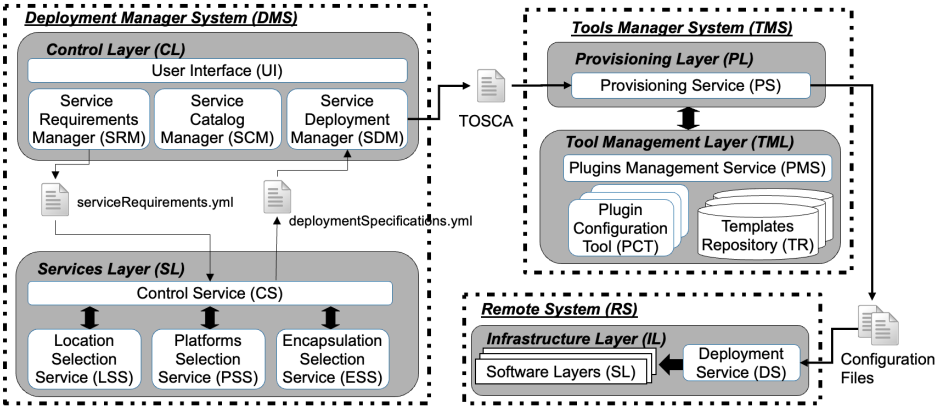


Fig. 1. Main blocks of the architecture presented in this article and their relationships. The proposed solution allows to manage the deployment of services on a service architecture

ing any deployment planning or optimization process. By assessing these requirements in advance, our approach focuses on generating optimal deployment configurations that enhance compatibility with heterogeneous infrastructures, simplify management, and improve the overall deployment process of applications and services. This perspective introduces a complementary dimension to the state of the art, addressing aspects generally assumed or overlooked in existing VNF and service placement solutions.

3. Solution architecture

To achieve the objectives outlined in Section 1, we have designed a novel architecture capable of providing the necessary functionalities. Although existing architectures facilitate service deployments in distributed cloud and edge environments, they often lack comprehensive support for the selection and optimization of deployment configurations, which is a key focus of our proposed solution.

Solutions such as T-NOVA[21], SONATA NFV[7], OSM [30], ONAP (Open Network Automation Platform)[6], Cloudify[10], and the MEC (Multi-access Edge Computing) Framework [12] offer valuable functionalities for infrastructure resource management, cloud and edge service deployment, and task automation. Each also provides specialized capabilities tailored to specific domains or needs. However, the architecture designed in this article differs mainly from these solutions in that it can consider specific factors such as infrastructure compatibility, location selection, and configuration of the service according to its needs and characteristics, taking them into account when deploying the service.

Fig. 1 illustrates the main blocks of our proposed architecture and their interrelationships. This solution enables the automated deployment and configuration of service-based systems. The architecture is divided into three main blocks: i) Deployment Manager System (DMS); ii) Tools Manager System (TMS); iii) Remote System (RS). Each block is described below.

3.1. Deployment Manager System (DMS)

The DMS offers tools to recommend the configuration of services and infrastructures before those services are deployed. Service providers (SPs) use the DMS to obtain configurations adapted to the operational requirements of the service to be deployed. When SPs want to deploy a new service, they could use the solution proposed in this work to obtain the configuration of the software layers where the services are to be deployed and then deploy them.

SPs interact with the DMS through the control layer (CL) which has a user interface (UI) that allows them to enter the description and requirements of the services. The Service Requirements Manager (SRM) is responsible for obtaining information on the operating requirements of the service to be deployed. As a result, the SRM writes service specifications and provides information in an understandable format for the modules contained in the SL.

The CL also integrates the Services Catalog Manager and Service Deployment Manager. The Service Catalogue Manager (SCM) maintains the information of services that have been previously registered. Facilitates the deployment of services that have been previously used, as it has a registry of previous deployment configurations. The Service Deployment Manager (SDM) receives the deployed specifications generated by the SL. The user can see the recommended options to deploy the desired service. In addition, the SDM provides the deployment specifications for the Tools Manager System (TMS). The TMS transforms the deployment specifications into configuration files that can be understood by configuration management tools.

The service layer (SL) is responsible for processing the information received from the CL. Upon receiving service requirements from the SRM, the SL initiates a process to determine the most suitable deployment configurations by evaluating three key aspects: i) the location of the service to be deployed; ii) the software architecture on which it will be deployed; and iii) the encapsulation technology. Each of these aspects is processed separately.

The SL uses the Location Selection Service (LSS) to decide whether the service should be deployed in an infrastructure located at the edge of the network or in a data center hosted in the cloud. Then, the SL uses the Platform Selection Service (PSS) to decide whether the service should be installed on a server or cloud architecture. Finally, the Encapsulation Selection Service (ESS) is responsible for evaluating the available options in terms of encapsulation technology. The ESS will decide whether it is more convenient to encapsulate a service inside a heavy or light virtualization container or even decide that it is better not to use virtualization.

3.2. Tools Manager System (TMS)

The TMS is in charge of deploying services over remote infrastructures by making use of different configuration management tools [17]. The TMS receives the service specifications and transforms them into an understandable format for configuration management tools (CMT). CMTs are responsible for facilitating and automating the configuration management of computer systems, applications and other software components [18]. In addition, CMTs favor maintenance update management [14], therefore they play an important role in software maintenance.

The TMS is composed of two layers: the Provisioning Layer (PL) and the Tool Management Layer (TML). The PL receives from the CL the service to be deployed and recommendations on how the service should be deployed (e.g., where the service should be deployed and whether or not virtualization technology should be used). For instance, if the software requires high processing capacity with low latencies, it might be better to run it on a system that does not have a virtualization layer between the host operating system and the service to be deployed, so that latency can be avoided. In such a case, the TMS receives from the DMS the specifications where the latency requirement is contemplated and proposes a latency-friendly deployment configuration. The Provisioning Service (PS) is able to process the recommendations proposed by the DMS with the objective of translating them into a format that is understandable by the configuration management tool installed in the service provider infrastructure. To perform this adaptation, the PL connects to the Plugins Management Service (PMS), which provides templates and configuration parameters specific to the selected configuration tool. Here, we propose a plugin-based system where each plugin contains relevant information about a configuration tool in order to enable the system to work with different such tools. This plugin-based design allows us to have a catalogue of configuration tools available in the system. In addition, plugins can be easily added, removed and even updated without affecting the operation of the rest of the system. The PMS consults the Plugin Configuration Tool and obtains information depending on the selected tool. In addition, it provides parameters and configuration templates to the PS which are necessary to make configuration files understandable to the configuration tool selected by the SP. Configuration management tools use local agents to facilitate management tasks on remote systems. The agents are deployed on the remote nodes and are responsible for implementing the instructions and commands received from the central node. Finally, the PS generates a package of configuration files according to the specifications received from the PMS.

3.3. Remote System (RS)

The RS represents the infrastructure on which the desired service will be installed. The RS is composed of an infrastructure layer where software services are deployed. This infrastructure layer can contain multiple software layers, each providing specific capabilities or technologies. Examples of software layers are the Java Virtual Machine or Microsoft's .NET framework, both providing environments for running applications.

3.4. DMS, TMS and RS Working Together

Our proposed solution integrates a set of distinct services, functionally grouped into the DMS, TMS, and RS, serving as a central point for various participants within a digital service ecosystem. Each participant in this ecosystem assumes a specific role based on their potential actions (see Fig. 2). The process unfolds as follows:

1. Service request by the consumer. A consumer requests a service from a Service Provider (SP).
2. Checking service availability. The SP queries the service catalogue to determine whether the requested service is already deployed in the available infrastructures.

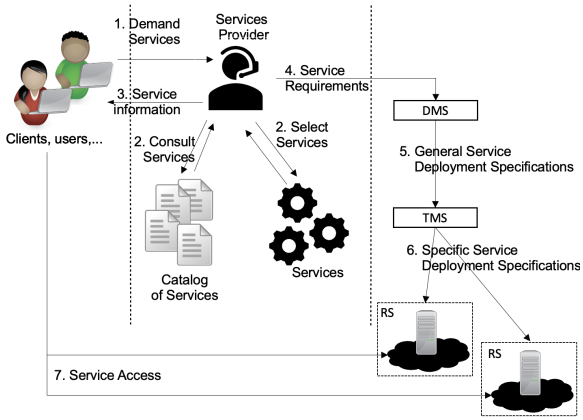


Fig. 2. Types of users, actions and flow of information exchanged between the main blocks of the proposed system and the users

3. Providing access when the service is already deployed. If the service exists and meets the operational requirements, the SP provides the consumer with the necessary information to access it.
4. Forwarding requirements when the service is not deployed. If the service is not deployed, the SP forwards the service requirements to the DMS.
5. Generating deployment specifications. The DMS processes the received requirements and generates a deployment specification in a TOSCA format file [22].
6. Adapting the deployment to the target infrastructure. Using the TOSCA file, the SP interacts with the TMS to generate the final deployment specifications tailored to the characteristics of the selected Remote System (RS).
7. Service deployment and access. Once the services have been deployed on the RS, consumers can access the requested service

4. Configuration Selection Process

As seen in the previous section, the DMS is able to analyze service requirements and decide which technological execution environment is the most suitable for the services to be deployed. This section shows the mechanisms used to select the most appropriate deployment configuration. For this purpose, it is necessary to define the requirements of the services considered in the selection process. Each requirement is defined by one or more qualitative values (Low, Medium, High). For example, a service may require a low boot time ({Low}), while another may accept low or medium ({Low, Medium}). This simple and well-defined way of describing the requirements, rather than using numerical values, facilitates their interpretation and specification by service providers. The list of identified service requirements $SR = \{r_1, r_2, ..., r_{16}\}$ is presented below:

– Time to be transferred (r_1). Time taken to transfer a given service to deploy it in another location or system.

- Time To Automatically Deploy (r_2). Time taken to deploy a given service.
- Boot time (r_3). Time required for a service to start operation, i.e. to transition from the stopped state to the running state.
- Compatibility (r_4). Ability of the service to be seamlessly deployed across a variety of infrastructures.
- Easy of Portability (r_5). Complexity of a service when it is deployed on different infrastructures.
- Isolation Degree (r_6). Degree of preference for a service to be encapsulated. Encapsulating a service facilitates its mobility (because it contains the configuration of the service) and provides security to it (deploying a service in a virtualised environment provides an additional software layer).
- Consumption of Resources (r_7). Indicates whether a service normally requires few, many, or an intermediate level of computational resources.
- Easy to manage (r_8). Degree of complexity of a service to be deployed.
- Scalability (r_9). The ability to dynamically allocate computational resources to a service that is running.
- Hardware Fault Tolerant (r_{10}). Indicates whether a service requires the infrastructure to be fault-tolerant at the hardware level.
- Cost (r_{11}). The cost of the use of the infrastructure where the service is deployed.
- Latency (r_{12}). Is the latency required on network connections.
- Jitter (r_{13}). The jitter of the network connection that the service is used.
- Distance to User (r_{14}). The physical distance between the deployed service and the users consuming it.
- Location Awareness (r_{15}). Used to indicate whether the service is strongly bound to a specific location or can be used in other available locations.
- Centralization Degree (r_{16}). Indicates whether the service should be deployed in a centralized location or not.

The deployment configuration selection process takes into account the service requirements and considers three dimensions of the execution environment, namely (i) the encapsulation technology, (ii) the deployment infrastructure, and (iii) the location. They are detailed in the following subsections.

4.1. Encapsulation technology selection

During the analysis carried out in this work, the following three types of encapsulation have been identified, each of them with different characteristics:

- *Heavy virtualization (HV)*: It is characterized by the installation of services in an operating system encapsulated in a virtual machine. The HV emulates the operating system and the hardware resources.
- *Light virtualization (LV)*: Similar to heavy virtualization based on virtual machines, but in this case, no hardware resources are virtualised and there is no need to install a new operating system.
- *No virtualization (NV)*: It is characterized by the installation of the service directly on the operating system and the service is not encapsulated.

Table 1. Comparison of deployment characteristics depending on the virtualization technology used. No virtualization (NV), light virtualization (LV) and heavy virtualization (HV)

Characteristic	NV	HV	LV
T. to be transferred (c1)	High	Medium	Low
T. to automatic deployment (c2)	High	Medium	Low
Boot time (c3)	High	Medium	Low
Compatibility (c4)	High	Medium	Low
Easy portability (c5)	Low	Medium	High
Isolation degree (c6)	High	Medium	Low
Consumption of resources (c7)	Low	Medium	Low
Easy to manage (c8)	High	Medium	Low

Normally, network functions are deployed in virtualized environments based on virtual machines [2, 26]. However, the use of VMs is not always the best option due to performance requirements which sometimes cannot be met by this technology (e.g. low start-up or high transfer times). For this reason, telecommunication service providers (TSPs) have explored other alternatives such as container network functions (CNFs) [5]. CNFs consist of running network services in containers instead of VMs, providing network functions (NFs) with performance similar to that they would have if they were running in a non-virtualized environment.

Containers are based on virtualization [32] and run as an application on top of a general-purpose operating system. This approach allows to boost efficiency through the elimination of the overhead produced by the virtualization of resources and drivers for the management of hardware resources in VMs. In addition, containers reduce system overhead compared to VMs. This model can run a multitude of isolated containers that share the operating system kernel. The CS collects the requirements for the services to be evaluated and then applies a selection process to obtain a numerical result for each of the types of encapsulation. The type of encapsulation in which a service is deployed has an impact on its performance, providing unique and different capabilities compared to other types of encapsulation.

Table 1 presents a comparison of how different encapsulation technologies impact key service characteristics (c_i) relevant to their selection.

For the encapsulation type selection process, our proposed mechanism compares a specific subset of service requirements ($SR_E = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$) with the characteristics of each encapsulation type presented in Table 1.

Let $E = \{NV, HV, LV\}$ be the set of encapsulation types. The Encapsulation Selection Service (ESS) uses expression (1) to obtain a numerical value (OP_e) that indicates the number of requirements that are fulfilled for the service being evaluated.

$$OP_e = \sum_{r \in SR_E} ET_{re}, \forall e \in E \tag{1}$$

$$ET_{ie} = \begin{cases} 1, & \text{if } v_{ie} \in r_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Table 2. Example of obtaining the value of OP_E as the number of matches between the service requirements (SR_E) and the characteristics of each encapsulation technology

	SR_E	NV	HV	LV
c1	Low	High	Medium	Low
c2	Low	High	Medium	Low
c3	Low	High	Medium	Low
c4	{Low, Medium}	High	Medium	Low
c5	{Medium, High}	Low	Medium	High
c6	{Low, Medium, High}	High	Medium	Low
c7	{High}	Low	Medium	Low
c8	{High}	High	Medium	Low
OP_E		2	3	6

where v_{ie} is the value of characteristic c_i for the type of encapsulation e. Finally, the decision on the recommended encapsulation type (V_E) is made according to the expression (3).

$$V_E = \arg\max_{e \in E} OP_e \tag{3}$$

Table 2 shows an example. The last row (OP_E) shows the results of applying equation (1) to each column NV, HV and LV of Table 2. In this example, the environment LV is the most appropriate to deploy that service.

4.2. Deployment infrastructure selection

This section shows the Platform Selection Service (PSS) mechanisms to obtain the optimal infrastructure when a given service is to be deployed. In this work, we have identified two possible infrastructures to consider:

- *Infrastructure based on virtual private servers (VPS)*: This infrastructure is characterized by having all the physical resources available for the services.
- *Cloud-based infrastructure (CC)*: This infrastructure is characterized by having a layer that manages the infrastructure resources transparently to the upper layers. The execution instances are managed by a hypervisor that manages the infrastructure resources globally and allocates them to the instances depending on their needs. The services and applications are deployed and executed in those instances.

VPS is a fixed block of physical server resources, while cloud-based infrastructures are based on multiple dedicated servers. VPS provides better customization options, resource allocation and speed and is more affordable than cloud-based infrastructures. In addition, VPS allows for near-instant scalability and also avoids noisy neighbor syndrome. This syndrome occurs in situations where a virtual machine (VM) monopolizes resources (such as bandwidth, disk I/O, and CPU) in a shared infrastructure. This causes performance issues for other virtual machines that share the same physical resources, which can lead to uneven network performance, increased CPU timeouts, disk I/O accesses, and more. In the case of cloud-based infrastructures, they offer higher uptime than VPS hosting due to their redundant system. Cloud hosting ensures high availability due to multiple servers in

Table 3. Comparison of the characteristics depending on the type of infrastructures used

	VPS	Cloud computing
Scalability (c9)	Low	High
Hardware fault tolerant (c10)	Low	High
Cost (c11)	Low	High

a redundant system. If one server fails, files and applications are immediately migrated to another server without stopping applications and services from running. Memory capacity and CPU power are also expanded on demand to meet the needs of another customer or compensate for those without affecting other users. The CS collects the requirements registered in the Platform Selection Service (PSS) related to the service to be evaluated. Choosing the right infrastructure for a service is important because it influences the behavior of the service, and therefore the user's experience when consuming that service. Table 3 presents a comparison of key service characteristics relevant to the selection of the deployment infrastructure. The selection process works similarly to the one in Section 4.1. Table 3 collects the characteristics related to different types of deployment infrastructures. The PSS, similar to the ESS, also models the service requirements through $SR_I = \{r_9, r_{10}, r_{11}\}$. Then, the selection process assigns a numerical value to each deployment type presented in the columns of Table 3. Let $I = \{VPS, CC\}$ be the set of deployment types. The Platform Selection Service (PSS) uses the expression (4) to obtain a numerical value (OP_i) that indicates the number of requirements that are fulfilled for the service being evaluated.

$$OP_i = \sum_{r \in SR_I} IT_{ri}, \forall i \in I \tag{4}$$

$$IT_{ji} = \begin{cases} 1, & \text{if } v_{ji} \in r_j \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where v_{ji} is the characteristic value c_j for infrastructure type i. Finally, the decision on the recommended infrastructure type (V_I) is made according to the expression (6).

$$V_I = \arg\max_{i \in I} OP_i \tag{6}$$

4.3. Location selection

This section discusses the decision process for deploying a service in a centralized or a decentralized data center. This is an important aspect when a service is to be deployed because it can influence the quality of the services. Edge computing and cloud computing [33] are two different technologies that serve different goals. Edge computing and cloud computing are different technologies that cannot replace each other. However, they can work together to provide an end-to-end solution. Cloud computing is more suitable for applications where large amounts of data need to be intensively processed at once (e.g., image recognition), can handle large-scale workloads, and can also provide centralized storage in large-scale data centers allowing access from anywhere with an Internet connection. On the other hand, edge computing is best suited for its use in applications that

Table 4. Comparison of the characteristics depending on each option of localization: Centralized Computing (CC), Edge Computing (EC)

	CC	EC
Latency (c12)	High	Low
Jitter (c13)	High	Low
Distance to the user (c14)	High	Low
Location awareness (c15)	Low	High
Centralization degree (c16)	High	Low

require real-time data processing and inter-device communication, processing real-time sensitive data, using in remote locations with limited or no network connectivity, providing local storage, etc.

This work takes into account the requirements and needs that are part of the selection process when the location of the infrastructure to deploy a service has to be decided. In this work, two locations have been identified and are listed in Table 4.

The Location Selection Service (LSS) applies a similar selection process, taking into account a specific subset of the service requirements relevant to the location of deployment $SR_L = \{r_{12}, r_{13}, r_{14}, r_{15}, r_{16}\}$ and compares them with the characteristics shown in Table 4.

We define $L = \{CC, EC\}$ as the set of localization types, where CC represents centralized computing in a specific location and EC represents edge computing distributed in different localizations.

The LSS uses expression (7) to obtain a numerical value that indicates the number of requirements that are fulfilled for the service being evaluated.

$$OP_l = \sum_{r \in SR_L} LT_{rl}, \forall l \in L \tag{7}$$

$$LT_{il} = \begin{cases} 1, & \text{if } v_{il} \in r_i \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where v_{il} is the value of characteristic c_i for the type of location l. Finally, the decision on the recommended location (V_L) is made according to the expression (9).

$$V_L = \arg\max_{l \in L} OP_l \tag{9}$$

5. Implementation and results

This section shows a prototype and the results of the tests performed on the proposed solution. To evaluate the work presented in this article, the deployment of a video streaming service has been considered. This service provides multimedia content to clients who connect to consume such content. Listing 1.1 presents the content of the *serviceRequirements.yml* file, which defines the requirements for the video streaming service used in our evaluation.

Listing 1.1. Code contained in the file serviceRequirements.yml

```
1  Service:
2    Name: VideoService
3    Description: It is a video server
4    timeToBeTransfered: low
5    timeToAutomaticallyDeploy: low
6    bootTime: low
7    compatibility: [low, medium]
8    easyOfPortability: [medium, high]
9    isolationDegree: [low, medium, high]
10   consumptionOfResources: high
11   easyToManage: high
12   scalability: high
13   hardwareFaultTolerant: low
14   cost: medium
15   latency: low
16   jitter: low
17   distanceToUser: medium
18   locationAwareness: low
19   centralizationDegree: medium
```

Listing 1.2. Code contained in the file *deploymentSpecification.yml*, which contains the results obtained after applying the selection process by LSS, PSS and ESS

```
1  V_E: LV
2  V_I: VPS
3  V_L: EC
```

Lines 2-3 are used to assign a name and description to the service, and the other lines are used to define the requirements of the service, which have been described in Section 4.

Specifically, the requirements related to the encapsulation technology (see Section 4.1) are those included in lines 4-11. They provide information to select the type of encapsulation. Lines 12-14 are discussed in Section 4.2 and contain aspects of configuration to decide the infrastructure where the service will be deployed. Finally, lines 15-19 are discussed in Section 4.3 and provide information about the requirements that depend on the service's location.

Listing 1.2 presents the output generated by our proposed solution after processing the service requirements. The resulting *deploymentSpecification.yml* file includes the recommended values for the encapsulation type (V_E), infrastructure type (V_I), and location (V_L). In this case, these values indicate a deployment in a lightweight virtualization environment hosted on a VPS server located at the network edge. The options associated with V_E, V_L, and V_I are fully independent from one another; no constraints exist between them. This independence enables a large number of valid combinations, allowing the generation of highly tailored deployment solutions.

Variable $V_E \in \{NV, HV, LV\}$ is related to the type of encapsulation of the service. When V_E is *NV*, it means that it is not recommended to encapsulate the service. In the case that V_E is *HV* or *LV*, the service should be encapsulated with heavy virtualization (HV)

or light virtualization (LV), respectively. $V_I \in \{VPS, CC\}$ indicates the type of infrastructure in which the service is deployed. If V_I is *VPS* it means that it is recommended to deploy the service in a virtual private service (VPS) infrastructure, while $V_I = CC$ means that it is recommended to deploy the service in a cloud computing (CC) infrastructure. Finally, $V_L \in \{CC, EC\}$ indicates the location most recommended to deploy the service, CC meaning a centralized infrastructure (cloud computing) and EC a non-centralized infrastructure (edge computing).

To verify the results obtained by the recommendation mechanisms, a prototype has been developed and deployed on two different infrastructures to compare the performance. The prototype and its operation are described in Section 5.1.

5.1. Target Scenario Emulation

This section presents the test environment employed and the main components used by the developed prototype. Fig. 3 shows a diagram of the location and connection of the components. The test environment attempts to simulate a multimedia content distribution scenario. The scenario models two multimedia content distribution options, the first option (option (a)) simulates a multimedia content distribution system deployed on a cloud computing provider where users connect to this server to consume multimedia content. The second option (option (b)) also emulates a content distribution system, but in this case, the streaming server is hosted in a data center with less computing power, in a network segment close to the users who consume the content. There are obvious differences between the systems in options. The streaming server in option (a) has a higher processing capacity than the server deployed in option (b), so option (a) will be able to process a greater number of requests and could serve more users. However, option (a), located somewhere on the Internet, could present worse network characteristics compared to the connections used in option (b). For example, the latency and packet loss parameters are expected to be worse in option (a) compared to option (b).

5.2. Performance Analysis

This section presents a comparison of the operation of a distribution system in different configurations. To simulate the operation as realistically as possible, we have used a video streaming server that provides videos on demand. Several tests have been carried out in which a set of nodes plays content from the server at a given moment. Several sets of nodes have been defined, including 25, 50, 100, 150, and 200 nodes.

The purpose of the tests is to verify how the end-user experience may be affected when trying to play multimedia content on a video distribution system. In theory, the video distribution service can vary the quality of service depending on whether the allocated resources are capable of meeting its needs. The processing capacity of the video streaming server is crucial to ensure its proper functioning. As the number of requests the server receives increases, there is a risk of overload. As a consequence, the time to content delivery is probably going to increase. Another relevant resource is related to the network when the server connects to the clients, since network saturation and bandwidth limitation are key factors for proper operation. The video distribution service can vary the quality of the service depending on whether the allocated resources can meet its needs. Among the

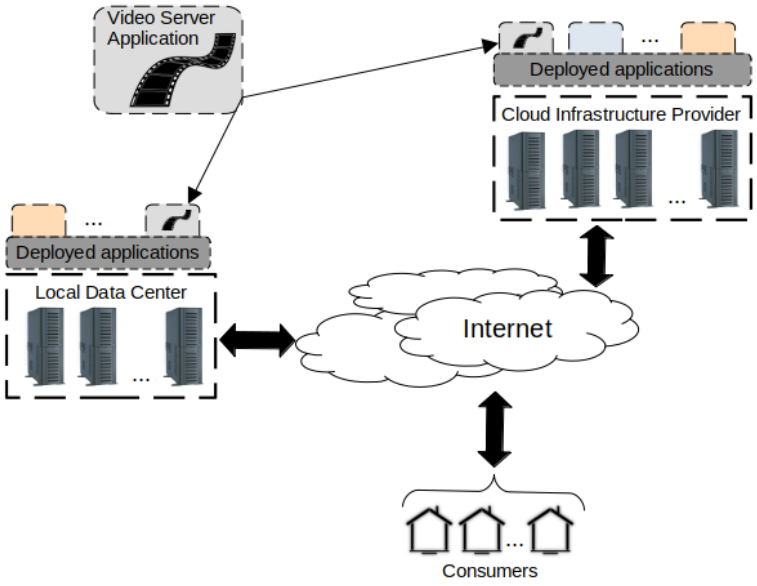


Fig. 3. A scenario consisting of a media server deployed in the cloud or on the edge network that delivers videos to consumers

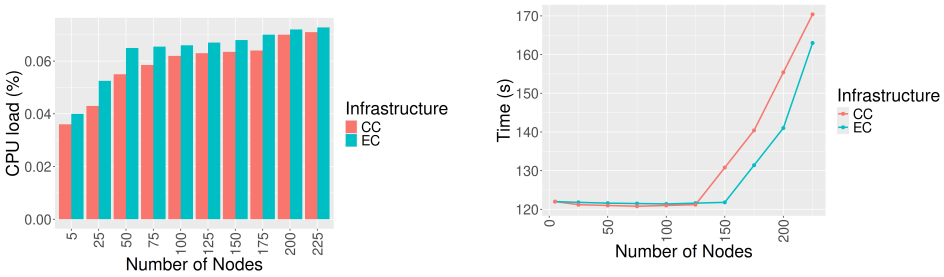
resources affected, one of the most important is the processing capacity available to the video streaming server, if it receives many requests it will need to process them all and will produce delays and latencies in the delivery of content. Another relevant resource is related to the network that connects the server to the clients, since network saturation and bandwidth limitation are key factors for proper operation.

Fig. 3 shows two possible options in which consumers view a video served on a media server. In option (a) the media server is deployed in a Cloud Computing infrastructure (CC). Option (b) presents the same case, but the media server is hosted in a data centre deployed at the edge of the network, in an Edge Computing infrastructure (EC). To model these options, we have developed a simulator based on Docker containers because they allow us to easily deploy the services involved and configure functional parameters that make it easy to simulate both options (a) and (b). The simulator consists of a set of containers, one container acting as a video server and the other containers simulating clients that consume the videos provided by the video server. Two networks, *Network1* and *Network2*, have also been created to simulate the path of the packets until they reach the video server. *Network1* and *Network2* are configured with different parameters that affect the performance of their connections, namely latency, bandwidth, and packet loss rate. In particular, *Network1* configuration is similar to that that connects consumers to a content server deployed in the Cloud Computing Infrastructure (CC), shown in option (a). While *Network2* configuration emulate to a Edge Computing infrastructure (EC), represented in option (b), where the media server is deployed at the edge of the network. The network parameters are set as follows: *Network1*: latency=15ms, bandwidth=50Mbps and packets loss rate=5%; *Network2*: latency=7ms, bandwidth=300Mbps and packets loss rate=2.5%.

To differentiate one option from the other, we have also considered reducing the computational resources available in the containers on which the media server runs. In option (b), the CPU resource of the container in which the media server is deployed has been reduced to 50% with respect to option (a) because we assume that the computational capacity in a datacenter at the edge of the network is more limited than in the cloud.

For a more accurate simulation, it is also important to adjust the computational resources to analyze the behavior of the media server when it is deployed in the cloud and at the edge of the network. Adequately satisfying the computational resources can significantly influence the response time of requests made to the video server by consuming nodes. If the server has abundant computational resources, the server could comfortably meet the demand. Otherwise, the server would have problems meeting requests and could even stop responding.

The results presented in this section are the average of 5 experiment repetitions.



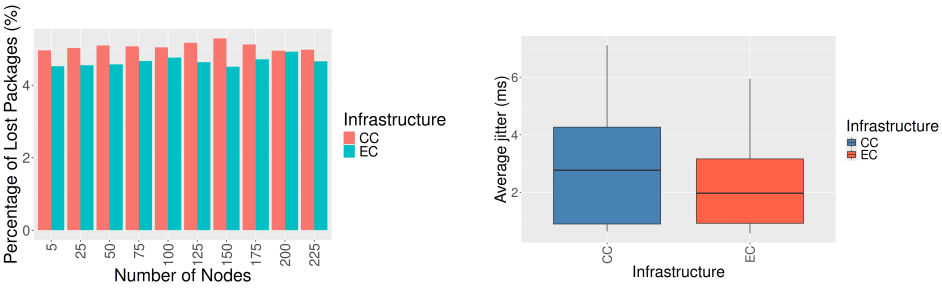
(a) Server CPU load

(b) Average time spent viewing multimedia content

Fig. 4. (a) Server CPU load (b) Average time spent viewing multimedia content

Fig. 4 (a) shows the processor load depending on the number of nodes. In scenario (a) (CC, Cloud Computing), the CPU load gradually increases as the number of clients increases. In scenario (b) (EC, Edge Computing), the CPU load is higher and grows faster than in scenario (a). The use of Edge Computing brings the processing closer to the user, however, the Edge Computing infrastructure has fewer resources compared to cloud-based infrastructures. This can lead to higher CPU stress on the Edge Computing infrastructure while the number of connected clients increases. In the cloud infrastructure, this does not occur because of its scalability, which allows it to handle more users with less impact on the CPU.

Fig. 4 (b) shows the average display time of the content as a dependency of the number of connected clients. In both options, the display time is initially stable, adjusting to the duration of the video (120 s), but when the load increases, the system collapses at a critical point, shooting up the display time. The collapse occurs earlier in scenario (a), possibly because the path that the packets have to travel to reach the destinations may be a bottleneck at some point in the network, consequently degrading the quality of service. In the case of the edge computing option, since the server is closer to consumers, network traffic must travel a shorter path, reducing the factors that can affect traffic transit.



(a) Comparison of the percentage of lost packets between cloud computing and edge computing infrastructures

(b) Comparison of average jitter measured between cloud computing and edge computing infrastructures

Fig. 5. (a) Comparison of the percentage of lost packets between cloud computing and edge computing infrastructures (b) Comparison of average jitter measured between cloud computing and edge computing infrastructures

In terms of network transmission quality, there are significant differences. The percentage of packet loss (Fig. 5 (a)) is lower in EC than in CC under high concurrency conditions, which directly results in more stable content viewing. Packet loss is a critical metric for measuring service quality and occurs when packets sent by the server do not reach the client, due to network congestion, node saturation, or transmission errors. In multimedia applications, even low packet loss can be perceived as dropouts, image degradation, or interruptions in media content display. A particularly relevant parameter in multimedia transmission applications is jitter, defined as the variation in packet arrival times on the network. Low jitter indicates regular and predictable delivery, which translates into a satisfactory user experience. However, high jitter causes irregularities that degrade the user experience. Fig. 5 (b) shows that EC maintains low jitter even under high load conditions, while CC shows a significant increase as concurrency increases. This confirms that EC provides greater temporal stability, a key factor in services sensitive to variability, such as real-time streaming.

The latency analysis also reinforces this conclusion. Latency is the total time it takes for a packet to travel from the source (server) to the destination (client). This is a fundamental parameter as it directly affects the speed with which users receive information. In multimedia services, low latency values improve user experience, while high values can cause noticeable delays. Fig. 6 shows the minimum latency values, where the EC consistently shows shorter times due to the proximity of the nodes to the end user. In contrast, the maximum latency (also shown in Figure 6) reflects that CC experiences much higher peaks when concurrency increases, while EC, although it also increases its latency in extreme conditions, does so in a more contained manner. This means that the edge infrastructure not only reduces average response times, but also limits extreme peaks that can compromise service stability.

The lower curves in Fig. 6 also show the response times (RTT) standard deviation. The standard deviation acts as a complementary indicator to jitter. Although jitter reflects variations in arrival intervals between consecutive packets, the standard deviation measures the overall dispersion of response times from the average. A low value indicates a more

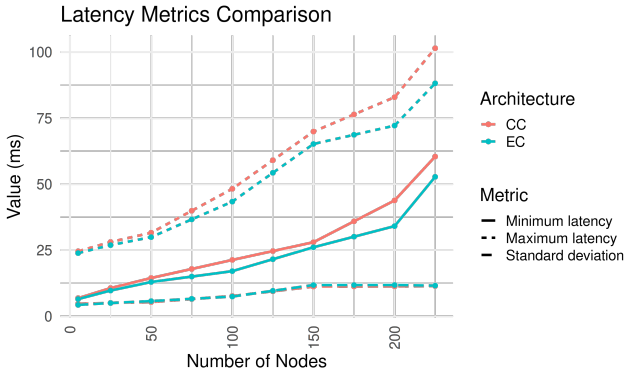


Fig. 6. Comparison of maximum and minimum latency measured between cloud computing infrastructures and edge computing versus standard network latency

stable and predictable network, whereas a high value reveals instability, with significant fluctuations affecting service consistency. Values between 1–5 ms are adequate for most interactive and multimedia applications. Values between 5–10 ms may be sufficient for video-on-demand services, but variations are already noticeable in delay-sensitive applications. Values greater than 10 ms present a risk of micro-cuts, buffering, or loss of fluidity in real-time multimedia services. As shown in Fig. 6 EC achieves lower standard deviations compared to CC, resulting in greater consistency in communication and therefore a more satisfactory user experience. In combination with jitter values, this metric confirms that systems deployed on the edge offer more robust performance under intensive loads.

In summary, the experimental comparison shows that Edge Computing outperforms Cloud Computing in terms of scalability, reliability, and network stability. Lower packet loss, lower jitter, reduced latency, and lower standard deviations position EC as an ideal infrastructure for streaming applications and interactive services where quality of experience is a critical factor.

5.3. Discussion

The objective of developing the prototype presented in Section 5 was to evaluate the performance of a video service. The features are listed in Listing 1.1 . Listing 1.2 in Section 5 shows the result of the selection process proposed in this work, which provides the recommended values for encapsulation type (V_E), infrastructure type (V_I), and location (V_L).

In the case of the location selection process, the satisfaction of characteristics 15 to 19 coincides with the decision to use EC. As can be seen in the analysis in the previous section, the characteristics that indicate latency (line 15) and jitter (line 16) are critical parameters in multimedia transmission, and the results confirm the advantage of EC. While CC shows high peaks of maximum latency (Fig. 6) and a notable increase in jitter under high concurrency (Fig. 5 (b)), in EC both parameters remain low and stable, ensuring higher quality of experience for the user.

The characteristics *distanceToUser* (line 17), *locationAwareness* (line 18), and *centralizationDegree* (line 19) are factors closely related to location. Centralization in remote data centers increases dependence on the transport network, negatively affecting latency and stability. In contrast, the geographical proximity of the nodes in EC reduces the distance perceived by the user and disperses the load, mitigating centralized saturation.

For the choice of encapsulation technology, the algorithm recommends using lightweight virtualization, for example Docker, which is what has been used in the prototype. Docker meets the requirements of fast deployment and startup (lines 5 and 6) because containers share the host operating system kernel and start up in seconds. However, VMs require a complete guest operating system startup, increasing boot time; while the option no virtualization, deploying the same software contained in the container but installed directly on the host machine, is even slower and more difficult to automate. Another key aspect is portability and compatibility. The definition of the portability attribute (line 8) shows that the service must be relatively easy to transfer. Docker encapsulates dependencies and execution environments, which reduces problems derived from limited compatibility (line 7) and facilitates its execution in heterogeneous infrastructures such as those of Edge Computing.

The high resource consumption attribute (line 10) makes the choice of virtualization particularly relevant. Docker introduces minimal overhead compared to VMs, which require additional resources to maintain complete operating systems. Thus, the density of instances per node is higher in Docker, allowing it to support more concurrent clients without degrading performance, which aligns with the high scalability (line 12) defined for the service. In addition, the ease of management (line 11) is another advantage of Docker because it offers standardized orchestration, monitoring, and update mechanisms, which simplifies the operation of distributed services.

6. Conclusion

Typically, automated service deployment mechanisms are modeled using Integer Non-Linear Programming (INLP) problems. However, it is not a viable solution due to the NP-hard nature of INLP problems, particularly when dealing with large-scale systems or those with a multitude of factors to consider. As a result, most of the work in this field relies on heuristic-based algorithms to efficiently solve problems within acceptable time frames. Our proposed solution seeks to address this issue by providing the capability to define features and requirements that help reduce the search space compared to existing approaches. It is important to note that, unlike other approaches, our solution does not focus on optimizing the performance of services or applications. Instead, it focuses on creating optimal configurations for software deployment, thereby enhancing compatibility, management, and the deployment process of applications and services.

This work proposes an architecture to improve the deployment of services on different infrastructures. The solution also proposes three selection mechanisms that help select the most appropriate configuration for the deployment of a service. For this purpose, the following configuration selection processes have been presented: the selection process to decide the most appropriate encapsulation technology with which to deploy a service; the selection process to detect the appropriate infrastructure in which to deploy the service; and the selection process to choose the most appropriate location in which to deploy the

service. To carry out these processes, a set of relevant service requirements have been identified and, in addition, a set of characteristics related to the configuration option on which the services are deployed have also been identified. The work detects which requirements are satisfied by the characteristics associated with the different configuration options available and proposes a deployment configuration.

In summary, our proposal presents a valuable and efficient alternative to INLP-based approaches in service deployment, effectively tackling the complexities of deploying applications and services in large-scale systems by focusing on optimal software configuration for enhanced compatibility, management, and deployment. For future research, we plan to analyze additional service deployment scenarios to identify a wider range of relevant requirements. Furthermore, we intend to explore the incorporation of more features to enable a more granular definition of infrastructures and address other complex configuration aspects.

Acknowledgments. Iván Bernabé Sánchez has been funded by the Spanish Ministry of Universities through a grant related to the Requalification of the Spanish University System 2021–23 Margarita Salas by the Carlos III University of Madrid. This work was supported by the Spanish Government under the research projects "Enhancing Communication Protocols with Machine Learning while Protecting Sensitive Data (COMPROMISE)" PID2020-113795RBC32, COSASS PID2021-123673OB-C32, funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", and VAE: TED2021-131295B-C33 funded by MCIN/AEI/ 10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR". This research also results from the I-Shaper Strategic Project (C114/23), due to the collaboration agreement signed between the Instituto Nacional de Ciberseguridad (INCIBE) and the Universidad Carlos III de Madrid. It is worth highlighting that this initiative is being carried out within the framework of the Recovery, Transformation and Resilience Plan funds, funded by the European Union (Next Generation).

References

1. Agarwal, S., Malandrino, F., Chiasserini, C.F., De, S.: Vnf placement and resource allocation for the support of vertical services in 5g networks. IEEE/ACM Transactions on Networking 27(1), 433–446 (2019)
2. Alnaim, A.K., Alwakeel, A.M., Fernandez, E.B.: A pattern for an nfv virtual machine environment. In: 2019 IEEE International Systems Conference (SysCon). pp. 1–6 (2019)
3. Bari, F., Chowdhury, S.R., Ahmed, R., Boutaba, R., Duarte, O.C.M.B.: Orchestrating virtualized network functions. IEEE Transactions on Network and Service Management 13(4), 725–739 (2016)
4. Battisti, A.L.É., Macedo, E.L.C., Josué, M.I.P., Barbalho, H., Delicato, F.C., Muchaluat-Saade, D.C., Pires, P.F., Mattos, D.P.d., Oliveira, A.C.B.d.: A novel strategy for vnf placement in edge computing environments. Future Internet 14(12), 361 (2022)
5. Cziva, R., Pezaros, D.P.: Container network functions: Bringing nfv to the network edge. IEEE Communications Magazine 55(6), 24–31 (2017)
6. Debeau, E., Quintuna-Rodriguez, V.: Onap: An open source toolkit for zero touch automation. In: Design Innovation and Network Architecture for the Future Internet, pp. 212–249. IGI Global (2021)
7. Dräxler, S., Karl, H., Peuster, M., Kouchaksaraei, H.R., Bredel, M., Lessmann, J., Soenen, T., Tavernier, W., Mendel-Brin, S., Xilouris, G.: Sonata: Service programming and orchestration for virtualized software networks. In: 2017 IEEE international conference on communications workshops (ICC workshops). pp. 973–978. IEEE (2017)

8. Ericsson: Ericsson mobility report. Tech. rep., Ericsson (Nov 2024), https://www.ericsson.com/4adb7e/assets/local/reports-papers/mobility-report/documents/2024/ericsson-mobility-report-november-2024.pdf, accedido: 15 de diciembre de 2024
9. Fei, X., Liu, F., Xu, H., Jin, H.: Towards load-balanced vnf assignment in geo-distributed nfv infrastructure. In: 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS). pp. 1–10 (2017)
10. Gangadharan, G.: Open source solutions for cloud computing. Computer 50(01), 66–70 (2017)
11. Gao, X., Liu, R., Kaushik, A., Zhang, H.: Dynamic resource allocation for virtual network function placement in satellite edge clouds. IEEE Transactions on Network Science and Engineering 9(4), 2252–2265 (2022)
12. Giust, F., Costa-Perez, X., Reznik, A.: Multi-access edge computing: An overview of etsi mec isg. IEEE 5G Tech Focus 1(4), 4 (2017)
13. Goudarzi, S., Soleymani, S.A., Anisi, M.H., Jindal, A., Dinmohammadi, F., Xiao, P.: Sustainable edge node computing deployments in distributed manufacturing systems. IEEE Transactions on Consumer Electronics pp. 1–1 (2023)
14. Grinter, R.E.: Using a configuration management tool to coordinate software development. In: Proceedings of conference on Organizational computing systems. pp. 168–177 (1995)
15. Gu, L., Tao, S., Zeng, D., Jin, H.: Communication cost efficient virtualized network function placement for big data processing. In: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 604–609. IEEE (2016)
16. Harris, D., Raz, D.: Dynamic vnf placement in 5g edge nodes. In: 2022 IEEE 8th International Conference on Network Softwarization (NetSoft). pp. 216–224. IEEE (2022)
17. Hintsch, J., Görling, C., Turowski, K.: A review of the literature on configuration management tools (2016)
18. Jha, D.N., Li, Y., Jayaraman, P.P., Garg, S., Ushaw, G., Morgan, G., Ranjan, R.: Challenges in deployment and configuration management in cyber physical system. Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things pp. 215–235 (2020)
19. Jin, P., Fei, X., Zhang, Q., Liu, F., Li, B.: Latency-aware vnf chain deployment with efficient resource reuse at network edge. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. pp. 267–276 (2020)
20. Kim, S.I., Kim, H.S.: A vnf placement method based on vnf characteristics. In: 2021 International Conference on Information Networking (ICOIN). pp. 864–869. IEEE (2021)
21. Kourtis, M.A., McGrath, M.J., Gardikis, G., Xilouris, G., Riccobene, V., Papadimitriou, P., Trouva, E., Liberati, F., Trubian, M., Batallé, J., et al.: T-nova: An open-source mano stack for nfv infrastructures. IEEE Transactions on Network and Service Management 14(3), 586–602 (2017)
22. Koziolek, H., Hark, R., Eskandani, N., Nguyen, P.S., Rodriguez, P.: Tosca for microservice deployment in distributed control systems: Experiences and lessons learned. In: 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C). pp. 11–21. IEEE (2023)
23. Kumar, M., Walia, G.K., Shingare, H., Singh, S., Gill, S.S.: Ai-based sustainable and intelligent offloading framework for iiot in collaborative cloud-fog environments. IEEE Transactions on Consumer Electronics pp. 1–1 (2023)
24. Leivadeas, A., Kesidis, G., Ibnkahla, M., Lambadaris, I.: Vnf placement optimization at the edge and cloud. Future Internet 11(3), 69 (2019)
25. Li, L., Qiu, Q., Xiao, Z., Lin, Q., Gu, J., Ming, Z.: A two-stage hybrid multi-objective optimization evolutionary algorithm for computing offloading in sustainable edge computing. IEEE Transactions on Consumer Electronics pp. 1–1 (2024)
26. Mandal, P.: Comparison of placement variants of virtual network functions from availability and reliability perspective. IEEE Transactions on Network and Service Management 19(2), 860–874 (2022)

27. Pham, C., Tran, N.H., Ren, S., Saad, W., Hong, C.S.: Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach. IEEE Transactions on Services Computing 13(1), 172–185 (2017)
28. Promwongsa, N., Ebrahimzadeh, A., Glitho, R.H., Crespi, N.: Joint vnf placement and scheduling for latency-sensitive services. IEEE Transactions on Network Science and Engineering 9(4), 2432–2449 (2022)
29. Reddy, K.H.K., Goswami, R.S., Luhach, A.K., Chatterjee, P., AlNumay, M., Roy, D.S.: Eflsm:-an intelligent resource manager for fog layer service management in smart cities. IEEE Transactions on Consumer Electronics pp. 1–1 (2024)
30. Reid, A., González, A., Armengol, A.E., de Blas, G.G., Xie, M., Grønsund, P., Willis, P., Eardley, P., Salguero, F.J.R.: Osm scope, functionality, operation and integration guidelines. ETSI, White Paper (2019)
31. Sallam, G., Ji, B.: Joint placement and allocation of vnf nodes with budget and capacity constraints. IEEE/ACM Transactions on Networking 29(3), 1238–1251 (2021)
32. Soltesz, S., Pötzl, H., Fiuczynski, M.E., Bavier, A., Peterson, L.: Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In: Proceedings of the 2Nd ACM SIGOPS/EuroSys european conference on computer systems 2007. pp. 275–287 (2007)
33. Sriram, G.: Edge computing vs. cloud computing: an overview of big data challenges and opportunities for large enterprises. International Research Journal of Modernization in Engineering Technology and Science 4(1), 1331–1337 (2022)
34. Sun, J., Liu, F., Wang, H., Wu, D.O.: Joint vnf placement, cpu allocation, and flow routing for traffic changes. IEEE Internet of Things Journal 10(2), 1208–1222 (2022)
35. Virtualisation, N.F.: An introduction, benefits, enablers, challenges & call for action. In: White Paper, SDN and OpenFlow World Congress. p. 73 (2012)
36. Wu, Y., Tang, S., Zhang, L., Fan, L., Lei, X., Chen, X.: Resilient machine learning based semantic-aware mec networks for sustainable next-g consumer electronics. IEEE Transactions on Consumer Electronics pp. 1–1 (2023)
37. Zeng, Z., Xia, Z., Zhang, X., He, Y.: Sfc design and vnf placement based on traffic volume scaling and vnf dependency in 5g networks. CMES-Computer Modeling in Engineering & Sciences 134(3) (2023)

Iván Bernabé-Sánchez received a degree in information technology engineering from the Carlos III University of Madrid, Leganés, Spain, in 2007 and a PhD degree in 2021. His research interests include the virtualization of devices and infrastructures defined through code, cloud and edge computing architectures, and self-configuration systems mechanisms based on knowledge representation and semantic technologies. He has participated in several nationally or internationally funded research projects.

Daniel Diaz-Sánchez is a full professor at University Carlos III of Madrid. His research interests include distributed authentication/authorization, content protection, distributed computing, fog computing, IoT and Smart Cities. Díaz-Sánchez received a PhD in telematics engineering from UC3M.

Alberto Fernández is a full professor at the University Rey Juan Carlos (URJC) in Madrid, where he is a member of the Artificial Intelligence Group of the CETINIA research centre. He obtained a PhD in Computer Science from the URJC. His main research lines are multi-agent systems, knowledge representation, semantic technologies,

open systems, etc. He is especially interested in the application of previous technologies in domains such as intelligent transportation systems, fleet management, etc. He has participated in many national and international projects on the above topics and has published more than 80 articles in international journals, books and conferences.

Holger Billhardt received his M.Sc. in computer science from the TH Leipzig, Germany, and his PhD in computer science at the Universidad Politécnica in Madrid. He is currently a full professor of computer science at Universidad Rey Juan Carlos in Madrid, where he is a member of the Artificial Intelligence Group at the Centre for Intelligent Information Technologies (CETINIA). His research is concerned with multi-agent systems, especially with the coordination of agents in distributed, open and dynamic environments. He is the author or co-author of more than 100 research papers and has participated in several nationally or internationally funded research projects.

Sascha Ossowski is a full professor of computer science and director of the CETINIA research centre at the University Rey Juan Carlos in Madrid. He received a MSc degree in informatics from U Oldenburg (Germany) and a PhD in artificial intelligence from TU Madrid (Spain). The main themes of his research refer to models and mechanisms for coordination in all sorts of agent systems and environments. He was co-founder of the European Association for Multiagent Systems (EURAMAS), chaired the European COST Action on Agreement Technologies, and is an emeritus board member of the International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

Received: September 15, 2025; Accepted: November 30, 2025.

Computer Science and Information Systems 23(1):255–276 https://doi.org/10.2298/CSIS250922007Z

Weld proximity defect detection model for steel thin plates based on EP-YOLOv7

Runmei Zhang[1], Jingwei Fan[2], Zihua Chen[1,3,⋆], Zhong Chen[2], and Bin Yuan[2]

[1] School of Electronic and Information Engineering, Anhui Jianzhu University, 230000 Hefei, China
zhangrong@ahjzu.edu.cn
czh1619@ahjzu.edu.cn

[2] School of Mechanical and Electrical Engineering, Anhui Jianzhu University, 230000 Hefei, China
fjw3516@stu.ahjzu.edu.cn
cz1982@ahjzu.edu.cn
yuanbinwork@163.com

[3] Provincial and Ministerial Key Laboratory, Chang'an University, 710000 Xi'an, China
czh1619@ahjzu.edu.cn

Abstract. Quality inspection of steel plate welding is critical in industrial manufacturing. However, weld proximity defects often present diverse morphologies, overlapping regions, and dense distributions, posing challenges to accurate industrial defect inspection. Therefore, we propose an industrial detector based on the EP-YOLOv7. First, an Efficient Multiscale Channel Attention (EMCA) is introduced to strengthen multi-scale feature perception and improve the model's focus on weld proximity defects. Second, the EMCA module is integrated into the Efficient Layer Aggregation Network to enhance feature fusion and defect representation. Finally, a Partial-Bottleneck Decoupling Predictor Head (P-BD Head) is designed to significantly improve localization accuracy and reduce missed detections of small targets. Experimental evaluations on both a self-built weld proximity defect dataset and a public generalization dataset show that EP-YOLOv7 achieves mAP of 85.2%/56.2% and F1 scores of 80.3%/43.3%. Meanwhile, the model size increases by only 0.6 MB (total 37.9 MB), demonstrating that the proposed approach delivers substantial accuracy gains while maintaining lightweight computational complexity, suitable for practical industrial applications.

Keywords: Weld proximity defects, YOLOv7, EMCA, P-BD Head, machine learning

1. Introduction

Welding is a fundamental joining technology in modern industrial manufacturing, and its quality directly affects the service life and structural reliability of components. Among commonly used welded materials, steel thin plates are widely applied in industrial construction, aerospace, and intelligent manufacturing due to their lightweight properties,

⋆ Corresponding author

high strength, and recyclability [1]. However, their small thickness and narrow heat-affected zones make them highly vulnerable to thermal deformation during welding, often leading to weld proximity defects such as weld tumors, burning-through, shrinkage cavity, and black-gray oxidation[2], as shown in Figure 1. With the advancement of manufacturing technologies toward higher precision and automation, the requirements for weld quality inspection have become increasingly stringent. Therefore, achieving efficient and accurate detection and localization of weld proximity defects in steel thin plate welds is essential to ensuring welding safety and enhancing overall production efficiency.

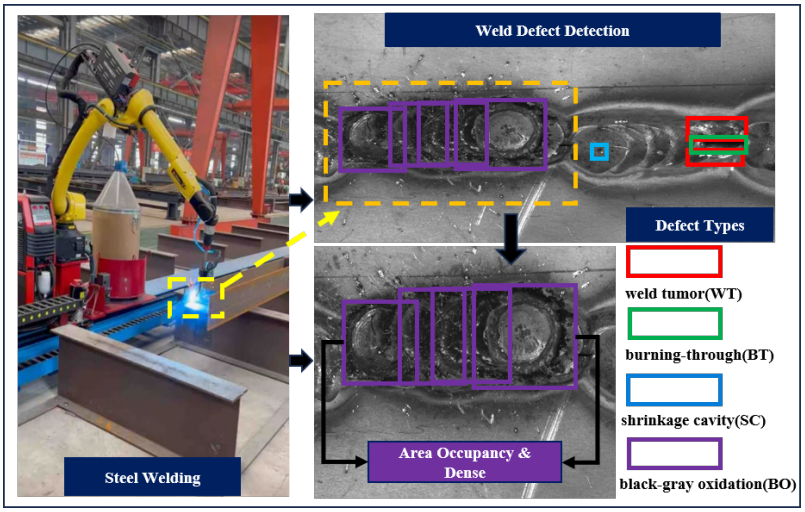


Fig. 1. Weld proximity defects

In recent years, several advanced defect detection techniques have emerged, mainly categorized into traditional classification-based methods [3] and deep learning-based object detection methods [4]. Traditional methods often combine handcrafted features with shallow classifiers, such as the Support Vector Machine (SVM) [5] and Histogram of Oriented Gradients (HOG) [6]. However, these approaches heavily rely on manually designed prior knowledge and shallow classifiers, limiting their ability to automatically learn multi-level and contextual semantic representations. Consequently, their performance deteriorates in complex visual scenes with dense and small defects, such as weld proximity defects.

Deep learning-based object detection has achieved remarkable progress in industrial defect detection, primarily categorized into two-stage algorithms(*e.g.*, Faster R-CNN [7,8,9] and Mask R-CNN [10]), and single-stage algorithms (*e.g.*, SSD [11] and YOLO [12,13,14] series). The two-stage detectors achieve target identification and localization by first generating candidate regions and then classifying and regressing the edges of each region. Chen et al [15] propose an improved Faster R-CNN model with a Res2Net backbone and a weighted fusion module to improve the accuracy of welding defect detection effectively. Although the two-stage detectors can enhance the defect recognition accu-

racy, their detection efficiency remains relatively low. It struggles to achieve a reasonable balance between model accuracy and speed, rendering it unsuitable for weld proximity defect detection scenarios in industrial manufacturing.

In contrast, single-stage detectors perform detection in a single network pass without generating candidate regions, enabling superior efficiency. Researchers have continuously optimized YOLO-based models to establish an effective trade-off between accuracy and speed in industrial defect detection.

For example, Wang et al [16] incorporate the ConvNeXt module into the backbone and introduce an attention mechanism in the pooling stage, effectively enhancing small-target feature extraction. Lu et al [17] design a mixed spatial pyramid pooling module (MIX-PCSPC) to preserve fine details that are often lost in traditional spatial pyramid pooling. Zhang et al [18] further improve small-target localization accuracy by integrating the attention mechanism and optimizing the ELAN module. Zhang et al [19] use Partial Convolution and Switchable Atrous Convolution to replace the 3×3 convolution in efficient layer aggregation networks, thereby improving adaptability to varying defect scales [20] and reducing the misdetection rate. Li et al [21] introduce a Coordinate Attention–based detection head and feedback connections in feature fusion, improving feature recognition and localization precision. Zhang et al [22] introduce an omnidimensional dynamic convolution that is highly sensitive to small targets, achieving significant improvements in model performance.

Although existing defect detection models have achieved notable progress, most approaches still focus on relatively simple weld defect scenarios and lack robustness in complex industrial environments where defects exhibit diverse morphologies, mutual occlusion, and high-density distributions. Therefore, this paper proposes a weld proximity defect detection model for steel thin plates based on EP-YOLOv7. The main contributions are summarized as follows:

1)We propose an Efficient Multiscale Channel Attention (EMCA) that combines fixed multiscale and dynamic adaptive convolutional branches in parallel for feature extraction. It adaptively captures multiscale information while maintaining computational efficiency, thereby enhancing the model's ability to locate weld proximity defects accurately.

2)To strengthen feature fusion and improve the discriminability of dense and overlapping defects, the EMCA module is integrated into the original efficient layer aggregation network module to form EM-ELAN and EM-ELAN-H modules. It optimizes intra-channel and inter-channel weight allocation, reinforces defect feature representation, and enhances the model's ability to extract and distinguish complex weld features.

3)To address localization inaccuracies caused by the coupled prediction head in baseline, we propose a Partial-Bottleneck Decoupling Predictor Head (P-BD Head) based on partial convolutions (PConv) and a dual-branch bottleneck layer. It effectively alleviates small-target miss detections and improves localization accuracy with minimal computational overhead.

The rest of this paper is organized as follows: Section 2 reviews some related works of the attention mechanism for target feature extraction and the YOLO detector for weld defect detection. Section 3 introduces the structure of the EP-YOLOv7 model. Section 4 gives the experimental results and some theoretical analysis. Finally, the conclusion and future work are given in Section 5.

2. Related Works

2.1. Attention Mechanism For Target Feature Extraction

In recent years, attention mechanisms[23,24,25] have been widely applied in object detection tasks. Their core idea is to dynamically assign weights so that the model focuses on the most relevant features, thereby improving feature extraction and object recognition accuracy. Existing attention mechanisms mainly include channel attention, spatial attention, and self-attention.

Channel attention (*e.g.*, Squeeze and Excitation (SE) Blocks[26]) reweights channel features by calculating dependencies between channels, thereby highlighting information-rich channels. For example, Xu et al [27]introduce SE Blocks into the final stage of the YOLOv5 backbone, assigning weights to image locations based on the channel domain perspective to obtain more feature information. Similarly, Li et al [28]based on the YOLOv8, introduced the SEV2 (Squeeze-and-Excitation Version 2) attention mechanism to effectively filter out general interference factors.

Spatial attention generates spatial attention maps, assigning different weights to each position in the feature map.(*e.g.*, Convolutional Block Attention Module (CBAM)[29]). Zoubir. et al[30] incorporate CBAM into the decoder to enhance feature fusion during upsampling by weighing encoder and Gabor filter features.

Self-attention mechanisms establish global dependencies between features, enabling long-range feature interactions. Recent Transformer-based detectors(*e.g.*, Detection Transformer (DETR)[31]) replace traditional components with global self-attention, achieving outstanding performance on COCO.

In summary, although existing attention mechanisms enhance feature extraction from different perspectives, they still struggle to balance multi-scale modeling, computational efficiency, and robustness in dense weld proximity defect scenarios.

2.2. YOLO Detectors For The Welding Defects

Weld defect detection methods are generally divided into two-stage and single-stage approaches. Single-stage detectors directly perform classification and regression on the full image, providing higher inference speed and better meeting industrial weld detection tasks. For example, Liang et al [32] propose an improved YOLOv4 algorithm for wheel weld detection by optimizing anchor sizes with k-means and employing the Distance-IoU loss function, which achieves significant accuracy improvement in both its homemade dataset and public dataset. Liu et al [33] propose an LF-YOLO model based on a reinforced multiscale feature (RMF) module and an efficient feature extraction (EFE) module, effectively enhancing multiscale feature extraction and addressing dimensional disparities in weld defect detection with extremely low consumption. To improve performance on more complex scenarios, Chen et al [34] developed DDT-YOLOv5 with DT-CSPNet and YOLO Di-head, significantly boosting detection accuracy. Xu et al [35] proposed enhancements to YOLOv7 by designing the Le-HorBlock module, coordinate attention, and the SIOU loss function, achieving a 15.9% increase in mAP and effectively reducing missed detections. A summary of the relevant literature is provided in the Table 1. However, although these improved YOLO-based methods perform well on general weld

defect datasets, they are still limited in detecting weld proximity defects. Therefore, to address these shortcomings and improve the detection of complex weld proximity defects, we propose the EP-YOLOv7 model.

Table 1. Summary of YOLO-based Weld Defect Detection Methods

Reference No.	Year	Model	Improvements	Shortcoming
[32]	2022	Improved YOLOv4	K-means and Distance-IoU loss	Limited multi-scale feature
[33]	2023	LF-YOLO	RMF and EFE modules	Limited dense-defect feature
[34]	2024	DDT-YOLOv5	DT-CSPNet and YOLO Di-head	Heavy computation
[35]	2024	Improved YOLOv7	Le-HorBlock module, CA, and SIOU loss	Insufficient generalization

3. Proposed EP-YOLOv7 Model

YOLOv7 is one of the single-stage target detection algorithms, and its network architecture is mainly composed of four parts: the input layer (Input), the backbone network (Backbone), the neck module (Neck), and the detection head (Head), as shown in Figure 2.

The Input module performs image preprocessing and enhancement [36], the Backbone extracts multi-level semantic features, the Neck fuses multi-scale information, and the Head completes final localization and classification.

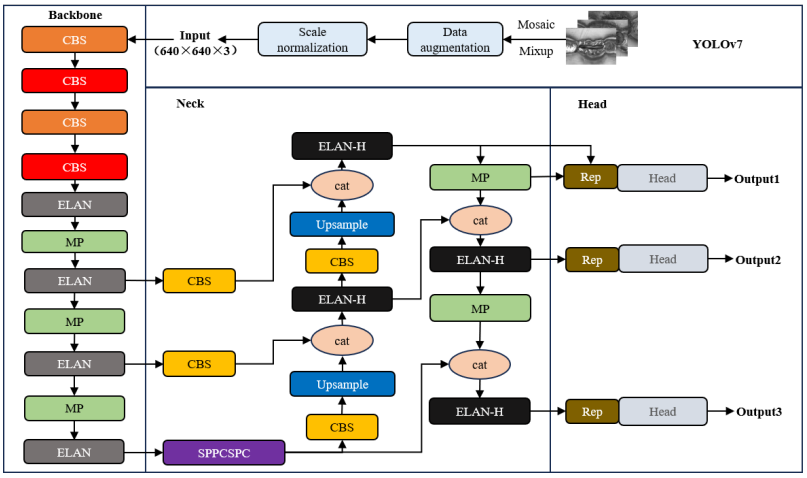


Fig. 2. Structure of YOLOv7

Although YOLOv7 performs well in many detection tasks, its performance remains limited when dealing with weld proximity defects in steel thin plates. Therefore, this paper proposes the EP-YOLOv7 model, as shown in Figure 3.

In Figure 3, we need to explain that CBS is the Convolution + Batch Normalization(BN) + Sigmoid Linear Unit(SiLU), SPPCSPC is the Spatial Pyramid Pooling + Cross-Stage Partial Connection, EM-ELAN & EM-ELAN-H are the improved Efficient Layer Aggregation Network modules, P-BD Head is the Partial-Bottleneck Decoupling Predictor Head, MP is the Maxpooling Layer, Upsample is the Upsampling Module, and Rep is the Reparameterized Convolutional Module.

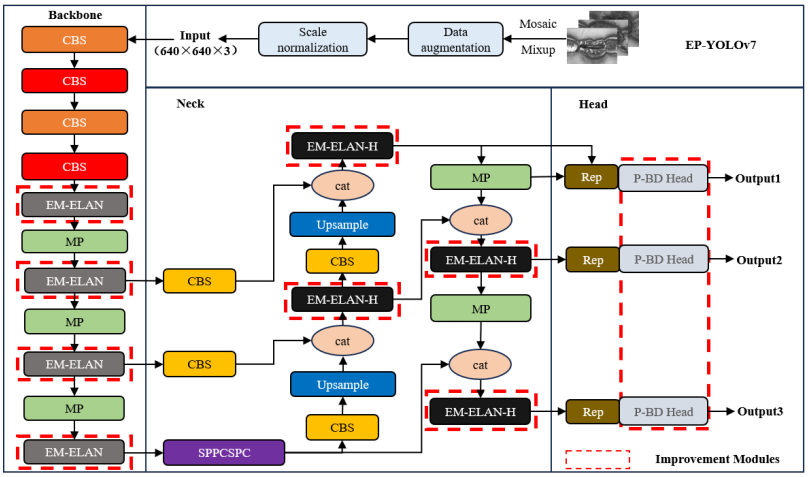


Fig. 3. Structure of EP-YOLOv7

Unlike the conventional YOLOv7, the proposed model incorporates three improvements to enhance the ability to detect weld proximity defects. First, an Efficient Multiscale Channel Attention (EMCA) module is introduced to enhance multiscale feature perception. By combining fixed multiscale and dynamic adaptive convolution branches, EMCA improves the model's sensitivity to subtle weld proximity defect features. Second, EMCA is embedded into the original ELAN and ELAN-H structures to form EM-ELAN and EM-ELAN-H modules. This integration strengthens feature fusion and improves the extraction of dense and overlapping defect regions. Finally, a Partial-Bottleneck Decoupling Predictor Head (P-BD Head) is proposed to reduce task conflict in the prediction stage. With a dual-branch bottleneck and PConv structure, it enhances localization accuracy and alleviates small-target missed detections with minimal computational overhead. Together, these improvements significantly enhance the capability of YOLOv7 in detecting complex weld proximity defects while maintaining lightweight model complexity.

3.1. EMCA Module

In convolutional neural networks, the attention mechanism filters extract key features from large amounts of information by assigning different weights to each part of the input feature map. The Efficient Channel Attention (ECA) module [37] achieves cross-channel interaction through adaptively tuned one-dimensional convolution, significantly improving detection accuracy. However, due to the use of a single convolutional kernel and the

neglect of multi-scale variations among channel features, ECA often struggles to capture key defect features.

To address this limitation, we propose an Efficient Multiscale Channel Attention mechanism (EMCA), which employs two parallel one-dimensional convolutional branches: a fixed multiscale branch and a dynamic adaptive branch, as shown in Figure 4.

The fixed branch contains 1×3, 1×5, and 1×7 convolution kernels to capture local, medium, and long-range channel dependencies, thereby covering channel correlations at different receptive-field ranges. The dynamic branch adaptively adjusts the convolutional kernels according to the number of channels of the input features, enabling flexible allocation of attention weights. Through the synergistic interaction of these two branches, EMCA enhances the representation of global contextual information. It allows the network to more accurately highlight channels associated with defect boundaries and suppress redundant background responses, leading to improved bounding-box regression precision and reduced false localization in weld proximity defects.

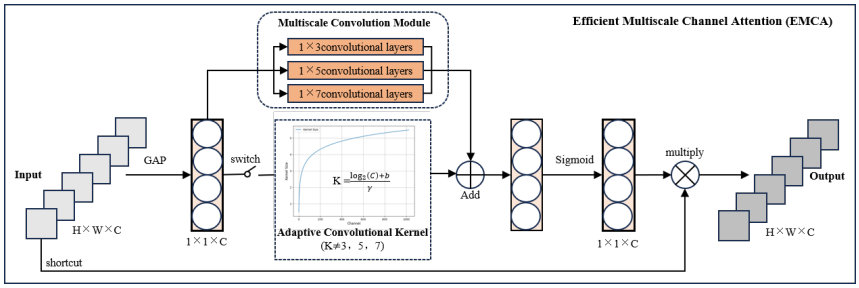


Fig. 4. Structure of the EMCA module

First, for the feature map with input dimension $H \times W \times C$, global average pooling is performed to aggregate spatial information within each channel and obtain a global channel of size $1 \times 1 \times C$, as shown in equation 1.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{c,i,j} \,. \tag{1}$$

where z_c denotes the global average pooling value of the channel C, and $X_{c,i,j}$ denotes the eigenvalue of the channel C at position (i, j).

Then, two parallel branches are constructed to perform one-dimensional convolution for the new feature vector, including an adaptive convolution kernel branch and a fixed multi-scale convolution branch. A switch mechanism is introduced to dynamically control branch activation. If both branches adopt the same kernel size, the adaptive branch is deactivated to reduce redundant computation. Otherwise, their feature extraction vectors are summed. The kernel size of the adaptive convolution ensures flexible modeling of cross-channel dependencies for different feature scales, as computed as shown in equation 2.

$$K = \left| \frac{\log_2(C) + b}{\gamma} \right|_{odd} . \tag{2}$$

where b and γ denote hyperparameters, and odd denotes the convolution kernel size taken as an odd number.

Next, the output vector obtained from the 1D convolution layers is activated by a non-linear function to generate the attention weights of each channel, allowing the network to highlight informative channels while suppressing less relevant ones, as shown in equation 3.

$$s_c = \sigma(Conv1D(z)) \,. \tag{3}$$

where σ denotes the Sigmoid activation function, $Conv1D(z)$denotes the one-dimensional convolution operation, and s_c denotes the final channel weight.

Finally, the learned attention weights are multiplied with the original input feature map through element-wise channel multiplication, realizing the selective enhancement of critical weld defect features, as shown in equation 4

$$\hat{X}_c = s_c \cdot X_c \,. \tag{4}$$

where X_c denotes the channel characteristic of the input feature, and $\hat{X}_c$ denotes the channel characteristics of the weighted output feature.

3.2. EM-ELAN And EM-ELAN-H Modules

ELAN and ELAN-H serve as efficient layer aggregation networks in the YOLOv7, employing multi-branch stacking and feature fusion to extract multi-level features. However, after fusion, they still lack fine-grained intra-channel weight allocation, which leads to insufficient feature refinement and limits the representation of weld proximity defects. To address these issues, we propose the EM-ELAN and EM-ELAN-H modules, as shown in Figure 5.

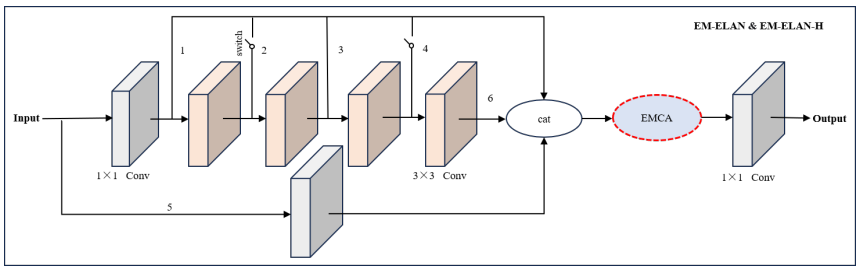


Fig. 5. Structure of EM-ELAN and EM-ELAN-H modules

In Figure 5, we need to explain that EM-ELAN and EM-ELAN-H follow the original multi-branch fusion structure but incorporate the proposed EMCA module after feature aggregation. The key difference lies in their fusion scope: EM-ELAN fuses partial branches(*i.e.,* Branches 1, 3, 5, and 6) for lightweight shallow-layer extraction in the backbone, while EM-ELAN-H fuses all branches (*i.e.,* Branches 1-6)in the neck to integrate richer semantic and multi-scale features.

Specifically, the input features are first processed through the ELAN/ELAN-H backbone to obtain deep-level weld proximity defect representations via multi-branch stacking and fusion. Then, the EMCA module adaptively assigns channel weights to the fused feature maps, enhancing both intra- and inter-channel interactions. Finally, a 1×1 convolution generates the output feature map. This enhanced design improves feature fusion quality and significantly strengthens the model's capability for representing and extracting complex weld proximity defects.

3.3. P-BD Head Module

In the traditional YOLOV7 architecture, the coupled detection head predicts classification, regression, and confidence simultaneously through a shared 1×1 convolution. This structure often causes conflicts between classification and localization objectives, leading to suboptimal performance in weld proximity defect detection. To address this limitation, we propose a Partial-Bottleneck Decoupling Predictor Head (P-BD Head), as shown in Figure 6.

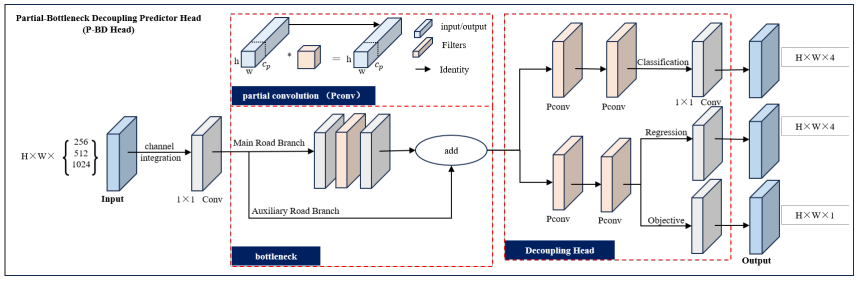


Fig. 6. P-BD Head structure

First, the decoupling prediction structure is employed to replace the original coupling detection head. By compressing channels and assigning independent convolutional branches for classification, regression, and confidence estimation, this architectural separation alleviates task interference and improves the training convergence speed[38].

Second, a dual-branch bottleneck layer is introduced after the input stage to enhance feature quality. The main branch performs channel reduction, local feature extraction, and channel restoration. Using a 1×1 convolution to reduce channels lowers computational cost while suppressing redundant information. The following convolutional layers focus on key local regions and capture deeper correlations, after which the channels are restored to maintain sufficient representation ability. Meanwhile, the auxiliary branch directly retains the original features, reducing gradient attenuation and preventing information loss during transmission.

Finally, the conventional 3×3 convolution is replaced with the PConv module in the overall structure. PConv extracts features from only part of the input channels while passing the remaining channels unchanged, and the outputs are fused through channel concatenation. This design strengthens local feature extraction for small weld defects while retaining global contextual information.

In summary, the P-BD Head introduces a decoupled architecture with a dual-branch bottleneck and Pconv. Together, these enhancements effectively reduce task conflicts, improve feature quality, and reinforce local detail representation, thereby boosting both localization accuracy and detection performance for weld proximity defects.

4. Experiments

4.1. Experimental Environment

The experimental environment is based on the Windows 10 operating system, the CPU is Intel Core i7-13700KF, the GPU is NVIDIA GeForce RTX 4090 (24 GB video memory), the Python version is 3.7, the CUDA version is 11.0, and the input image is 640×640 pixels.

The total number of training epochs is set to 100, divided into two phases: a frozen phase and an unfrozen phase, each lasting for 50 epochs. During the frozen training phase, the backbone network is frozen, and a batch size of 16 is used. In the unfrozen phase, all layers are trainable, and the batch size is reduced to 8. The Stochastic Gradient Descent (SGD) optimizer is employed with a momentum of 0.937 and a weight decay of 0.0005. The initial learning rate is set to 0.01, and a cosine annealing learning rate schedule is applied to gradually reduce the learning rate.

4.2. Stell Thin Plate Weld Defect Image Acquisition And Dataset Construction

In the welding process of steel thin plate, due to the limited number of samples of weld proximity defects, the main dataset in this paper is constructed as a real-life shooting homemade dataset, and the overall process is shown in Figure 7. The image acquisition location is the key laboratory of Anhui Province for intelligent manufacturing of construction machinery, and this experimental acquisition platform consists of a mechanical arm bracket, an industrial focal length, an industrial camera, a computer, and a steel thin plate. The welding samples mainly come from the real steel thin plate in the daily industrial welding process. The industrial focal length adopts a computar-fixed focus 45 million F2524-MPT industrial lens, and the industrial camera adopts Daheng Image ME2P-900-13GC/GM-P Mercury II Pro 9 megapixel color industrial camera. In this paper, image acquisition is carried out by mounting the industrial camera on the robotic arm, and the sampling platform is shown in Figure 7(a).

The collected defect data contains four kinds of weld proximity defects: Weld Tumor (WT), Burn-Through (BT), Shrinkage Cavity (SC), and Black-gray Oxidation (BO). The specific diagram of each defect classification is shown in Figure 7(b).

We select the 2,100 sample images in JPG format as the original dataset. Further, to enhance the generalization ability and robustness in detecting various types of weld defects, this dataset is augmented through geometric transformations, resulting in a total of 5,400 images. The final dataset is labeled using LabelImg software, and text data for training can be generated by drawing rectangular boxes in the defective regions in the images. Each image corresponds to a text file, and the labeling results are saved in XML format. Figure 7(c)shows the data labeling interface and the target box information generated after labeling.

In this experiment, 5400 images are divided into 4374 training sets, 486 validation sets, and 540 test sets. Among them, the number of weld proximity defects is 29124, including 14361 WT defects, 2400 BT defects, 9148 SC defects, and 3215 BO defects. The statistics of the number of defects in each category are shown in Figure 7(d).

In addition, the open-source weld defect detection dataset Welding-Defect-Final0 on the Robflow website is selected for generalization experiments, including five types of weld defects: Geometric defect, Non-fusion defect, crack, porosity, and spatters. Considering the balance of defect samples and data quality, a total of 1,000 images covering all five types of weld defects are selected, such as 200 samples of each category.

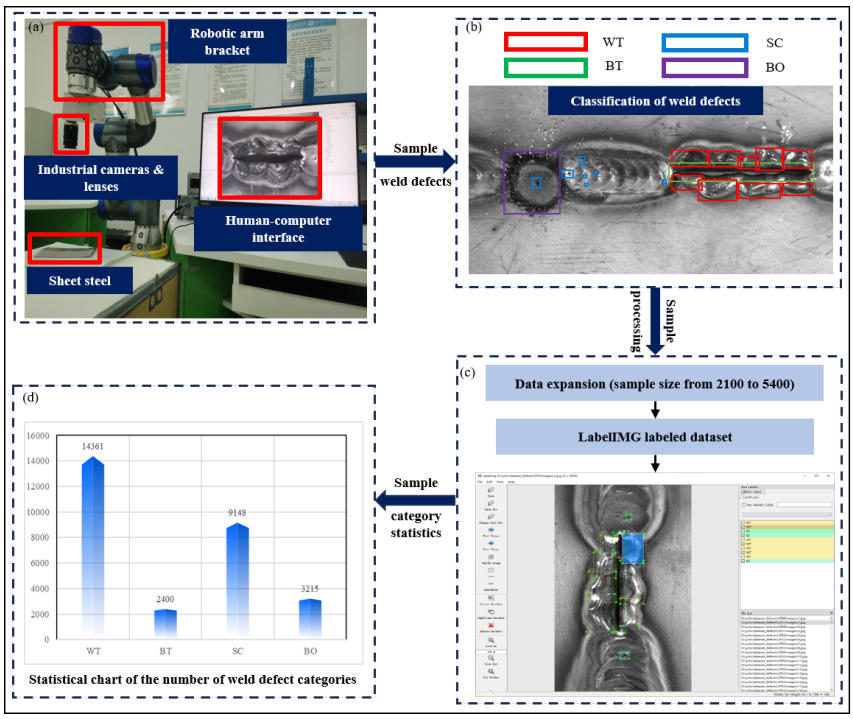


Fig. 7. The overall process of dataset construction. (a) Data collection platform for weld defects in steel sheet (b) Classification of weld defects in steel sheet (c) Expansion of weld defect samples and labeling of dataset (d) Statistical graph of the number of weld defects in each category

4.3. Evaluation Metrics

In this study, the following metrics are introduced: Precision (P), Recall (R), F1 Score (F1), mean Average Precision (mAP), model Parameters (Params), Average (Ave), and Inference Times(I-Time) for the evaluation of the model.

P characterizes the proportion of correctly identified positive samples among all predicted positives, as shown in equation 5. R characterizes the proportion of correctly detected defect samples among all actual defects, as shown in equation 6. F1 characterizes the harmonic mean of precision and recall, providing a balanced measure that considers both detection completeness and accuracy, as shown in equation 7.

$$P = \frac{TP}{TP + FP}\,. \tag{5}$$

$$R = \frac{TP}{TP + FN}\,. \tag{6}$$

$$F1 = \frac{2 \times P \times R}{P + R}\,. \tag{7}$$

For the weld dataset, four categories of proximity defects are considered, namely WT, BT, SC, and BO. For each category, the area under the Precision–Recall curve is calculated to obtain the AP, as shown in equation 8. The mAP is then obtained by averaging the AP values across all defect categories, serving as a comprehensive indicator of overall detection accuracy, as shown in equation 9.

$$AP = \int_0^1 P(R)dR\,. \tag{8}$$

$$mAP = \frac{\sum_1^4 AP}{4}\,. \tag{9}$$

To further evaluate the holistic performance of the proposed model, the Ave is introduced, which aggregates the key evaluation indicators to intuitively reflect the model's overall detection capability, as shown in equation 10.

$$Ave = \frac{P + R + F1 + mAP}{4}\,. \tag{10}$$

TP, FP, and FN denote the number of true positives, false positives, and false negatives, representing cases where a positive sample is correctly identified, incorrectly identified, or missed by the detector.

4.4. Ablation Experiments For EP-YOLOV7

This section aims to analyze the effect of each improved module on weld proximity defect detection. To this end, ablation experiments are performed using the original YOLOv7 as the baseline, and the improved modules are gradually added. The specific experimental results are shown in Table 2.

In Table 2, we need to explain that A is the improved efficient layer aggregation network module and B is the P-BD Head module.

Compared with the original YOLOv7, replacing the original module with the improved efficiency layer aggregation network leads to increases of 2.3%, 2.1%, 1.7%, and 1.4% in mAP, P, F1, and R. This improvement is achieved with nearly the same number

Table 2. Performance comparison of ablation experiments

Model	Epoch (/)	mAP (%)	P (%)	F1 (%)	R (%)	Params (M)	I-Time(s)
YOLOv7	100	81.7	89.1	77.5	68.6	37.3	0.0119
YOLOv7+A	100	84.0	91.2	79.2	70.0	37.3	0.0135
YOLOv7+B	100	84.8	89.5	80.0	72.1	37.9	0.0125
YOLOv7+A+B	100	**85.2**	90.7	**80.3**	**72.1**	37.9	0.0131

of parameters and only a slight increase in inference time, demonstrating that the EMCA module enhances multi-scale channel interaction and aggregation, thereby improving the model's sensitivity to weld surface defects and strengthening its feature extraction capability.

When replacing the original detection head with the P-BD Head, the performance is further improved, with mAP, P, F1, and R increased by 3.1%, 0.4%, 2.5%, and 3.5%. Although the P-BD Head introduces additional computation, its optimized decoupled design contributes to more accurate localization and detection of weld defects.

After integrating both modules, the overall improvement becomes more significant. Compared with the baseline, the proposed EP-YOLOv7 achieves improvements of mAP, P, F1, and R by 3.5%, 1.6%, 2.8%, and 3.5%. The parameter increases by only 0.6M, and inference time (0.0131s) shows only a minor rise, indicating that the proposed improvements achieve a better balance between accuracy and efficiency.

4.5. Efficient Layer Aggregation Network Module Comparison Experiment

This section aims to investigate the impact of different attention mechanisms on the performance of the ELAN and ELAN-H modules. To this end, we introduce improved ELAN and ELAN-H modules that integrate EMCA, ECA, SimAM, CBAM, and SE for comparative experiments. The specific experimental results are shown in Table 3.

Table 3. Comparison of efficient layer aggregation networks with different attention mechanisms

Model	Epoch (/)	mAP (%)	P (%)	F1 (%)	R (%)	Params (M)
YOLOv7	100	81.7	89.1	77.5	68.6	37.3
YOLOv7+ECA	100	82.9	89.1	79.8	72.3	37.3
YOLOv7+CBAM	100	83.3	89.1	79.6	71.9	37.3
YOLOv7+SimAM	100	83.4	89.6	78.3	69.6	39.5
YOLOv7+SE	100	82.2	88.1	78.4	70.7	38.4
YOLOv7+EMCA	100	**84.0**	**91.2**	79.2	70.0	37.3

Table 3 shows that both mAP and P are significantly improved in YOLOv7 when incorporating the EMCA into the module, compared to other attention-based variants, without increasing the overall network complexity. In Figure 8, we visualize and compare the two attention mechanisms with the highest mAP scores, CBAM and the EMCA. The results show that, compared to CBAM, EMCA effectively reduces missed detections and enables more accurate localization of weld proximity defects. The combined results of

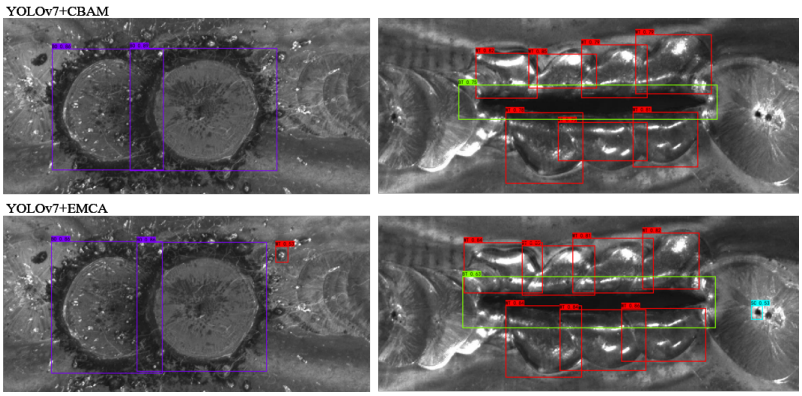


Fig. 8. Comparative visualization of different attention mechanisms

the above experiments show that EMCA strengthens the ability to focus on the information of weld proximity defects by adding fixed convolutional kernels. Furthermore, when combined with the efficient layer aggregation module, it markedly improves the model's feature extraction capability, thereby enabling accurate localization and identification of defects.

4.6. Ablation Experiment For P-BD Head

This section aims to investigate the contribution of each improved component within the P-BD Head to the overall detection performance. To this end, ablation experiments are carried out using the original YOLOv7 as the baseline model, where the decoupling head, bottleneck layer, and Pconv module are sequentially integrated for comparative analysis. The specific experimental results are shown in Table 4.

Table 4. P-BD head module ablation experiments

Model	Epoch (/)	mAP (%)	P (%)	F1 (%)	R (%)	Params (M)
YOLOv7	100	81.7	89.1	77.5	68.6	37.3
YOLOv7+C	100	83.3	89.3	76.0	66.1	44.8
YOLOv7+D	100	84.0	90.1	76.7	66.7	45.4
YOLOv7+B	100	**84.8**	89.5	**80.0**	**72.1**	37.9

In Table 4, we need to explain that C is an improved detection header with only the decoupling header added, D is an improved detection header with the bottleneck layer and decoupling header added, and B is an improved detection header with the bottleneck layer, decoupling header, and Pconv module added.

Compared to the original YOLOv7, the YOLOv7 with the decoupling head improves metrics mAP and P by 1.6% and 0.2%. The experimental results show that the decoupling head enhances detection accuracy by enabling each prediction task to be processed independently. However, due to the subdivision of the prediction task by the decoupling head,

the model's ability to detect low-confidence targets is slightly weakened, resulting in a minor decrease in R. Additionally, the introduction of a 3×3 convolutional layer increases the model size by 7.5 MB.

Furthermore, when both the bottleneck layer and the decoupling head are integrated into YOLOv7, mAP and P show further improvements. The bottleneck layer refines the input features to provide more feature information for subsequent decoupling processing.

Finally, after incorporating the P-BD Head into YOLOv7, mAP, F1, and R are improved by 0.8%, 3.3%, and 5.4%. While the model parameters are reduced by 7.5 MB compared with the version containing both the bottleneck layer and the decoupled head, with only a slight decrease in P. The ablation experiments show that with only a minimal parameter increase, the P-BD Head substantially enhances the localization and detection accuracy for the weld proximity defects, while significantly improving the model's recall capability.

4.7. Performance Comparison Of Different Target Detection Models

This section aims to comprehensively evaluate the overall performance of the proposed EP-YOLOv7 model. To this end, a series of comparative experiments is conducted on object detection algorithms, including two-stage detectors (*e.g.,*Faster R-CNN) and single-stage detectors (*e.g.,*SSD and the YOLO series), as well as recent References[12], [18], and transformer-based models (*e.g.,* RT-DETR). The specific results are shown in Table 5 and Figure 9, where Figure 9(a) shows loss function curves for the training and validation sets of the EP-YOLOv7 model, and Figure 9(b) shows the change curves of each evaluation index.

Table 5. Performance comparison of different target detection models

Model	Epoch (/)	mAP (%)	P (%)	F1 (%)	R (%)	Ave (%)	Params (M)
SSD	100	66.8	84.0	61.0	47.9	64.9	24.1
Faster-RCNN	100	73.1	62.9	69.5	77.7	70.8	28.4
YOLOv5	100	71.5	85.8	61.5	47.9	66.7	46.7
YOLOv7	100	81.7	89.1	77.5	68.6	79.2	37.3
YOLOv7-Tiny	100	73.5	86.9	66.5	53.8	70.2	6.0
[18]	100	74.5	87.5	67.2	54.6	71.0	8.6
YOLOv8n	100	80.2	90.4	77.7	68.2	79.1	3.0
YOLOv8l	100	80.7	88.1	77.5	69.1	79.0	43.6
YOLOv10n	100	77.7	90.7	73.7	62.1	76.1	2.3
YOLOv10l	100	81.3	90.4	78.8	69.9	80.1	24.3
YOLOv12	100	79.6	90.1	76.7	66.8	78.3	2.5
[12]	100	81.1	88.1	77.6	69.3	79.0	5.6
RT-DETR	100	81.9	76.7	75.8	74.9	77.3	28.5
EP-YOLOv7	100	**85.2**	**90.7**	**80.3**	72.1	**82.1**	37.9

Table 5 shows that the proposed EP-YOLOv7 model outperforms several mainstream detection models across multiple evaluation metrics(*e.g.,* mAP:85.2%, P:90.7%, F1:80.3%, Ave:82.1%). Compared to the original YOLOv7 model, the number of parameters in EP-YOLOv7 increases by only 0.6 MB, while the detection accuracy improves by 3.5%.

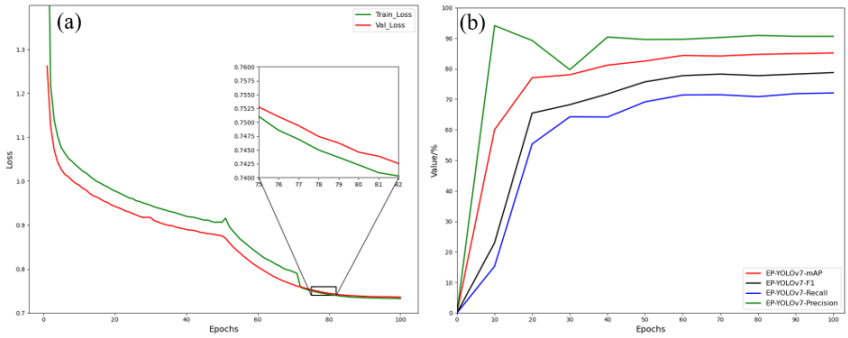


Fig. 9. EP-YOLOv7 effect plots.(a) training set and validation set loss function curves (b) P, R, mAP, and F1 variation curves

Although the R and parameter values are not the highest among all models, the Ave value surpasses those of the compared models, demonstrating its stronger overall performance in accurate weld proximity defect detection.

Meanwhile, according to Figure 9, both the training and validation loss functions exhibit a clear convergence trend, and all evaluation metrics (mAP, precision, recall, and F1) stabilize in the later training stages, indicating that the model achieves reliable convergence. In summary, the experimental results confirm that the proposed EP-YOLOv7 model achieves stable convergence and superior performance in the weld proximity defect detection task.

4.8. Comparison Chart Of Visualization Model Effect Detection

This section aims to provide an intuitive comparison of the detection performance among different models. We select three models with the highest Ave value(YOLOv7, YOLOv10l, and the proposed EP-YOLOv7)to visualize and compare their weld proximity defect detection results.

Due to the weld defects in this dataset of a variety of defects densely occupy the position, so this paper selects several representative defect combinations for comparison. The corresponding detection results are shown in Figure 10, where Figure 10(a)–(d) represent the combinations WT&BT&SC, WT&BO&SC classes, WT&BT&BO class, WT&BT&BO&SC classes. As shown in the figure, the proposed EP-YOLOv7 model outperforms both YOLOv7 and YOLOv10l in detecting weld proximity defects on thin steel plates.

Specifically, YOLOv7 and YOLOv10l exhibit noticeable missed detections across multiple scenarios. For instance, both models fail to WT defects in Figure 10(a). In Figure 10(b), the YOLOv7 model misses the WT and SC defects, while YOLOv10l misses the BO defects. Similarly, in Figure 10(c), YOLOv10l fails to detect BO defects. In Figure 10(d), YOLOv7 and YOLOv10l miss SC defects, and YOLOv10l additionally fails to detect BO defects. These results indicate that existing models struggle with dense multi-defect environments.

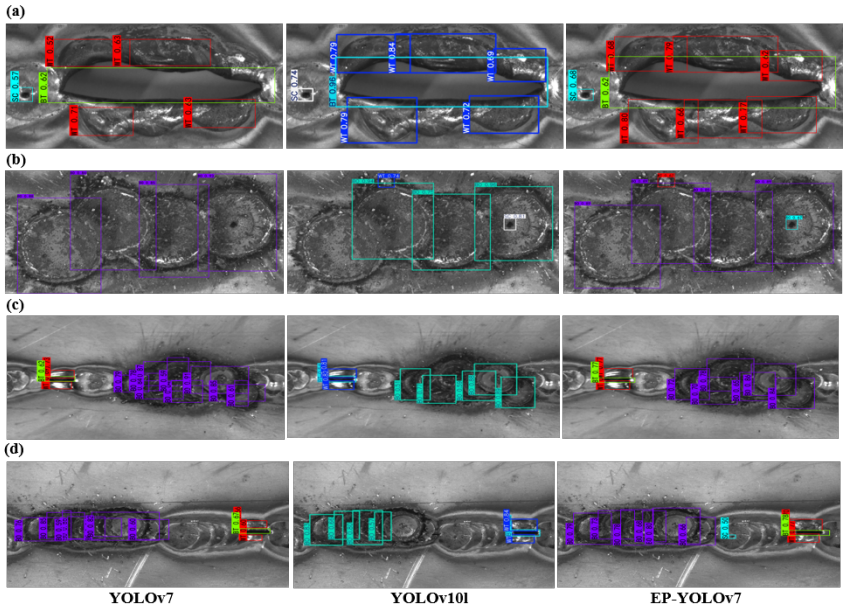


Fig. 10. Comparison of the detection effect of weld proximity defects (a) WT&BT&SC classes (b) WT&BO&SC classes (c) WT&BT&BO classes (d) WT&BT&BO&SC classes

In addition to missed detections, YOLOv7 exhibits inaccurate localization. In Figure 10(a), the predicted bounding boxes for WT and BT defects fail to fully cover the target regions. In Figure 10(b), the bounding box for the BO defect significantly exceeds the actual defect area. In contrast, the proposed EP-YOLOv7 model effectively addresses both missed detection and localization inaccuracies, achieving more accurate and reliable detection of weld proximity defects in steel thin plates.

4.9. Generalizability Experiments

This section aims to evaluate the generalization performance of the proposed EP-YOLOv7 model. To this end, we select the three models with the highest Ave value in the comparative experiments (YOLOv7, YOLOv10l, and the proposed EP-YOLOv7) and conduct experiments on the public dataset Welding-Defect-Final0. As shown in Table 6, EP-YOLOv7 achieves an mAP of 56.2%, representing improvements of 4.6% and 1.8% over the baseline YOLOv7 and YOLOv10l models.

Meanwhile, Figure 11 shows the detection results for the five weld defect categories in the dataset. Overall, the proposed model exhibits fewer missed detections and higher confidence scores, confirming the superior generalization ability of EP-YOLOv7 in weld defect detection.

Table 6. Results of generalization experiment data

Model	Epoch (/)	mAP (%)	P (%)	F1 (%)	R (%)	Params (M)
YOLOv7	100	51.6	88.0	40.0	25.7	37.3
YOLOv10l	100	54.4	89.8	41.4	26.9	24.3
EP-YOLOv7	100	**56.2**	**91.3**	**43.3**	**28.4**	37.9

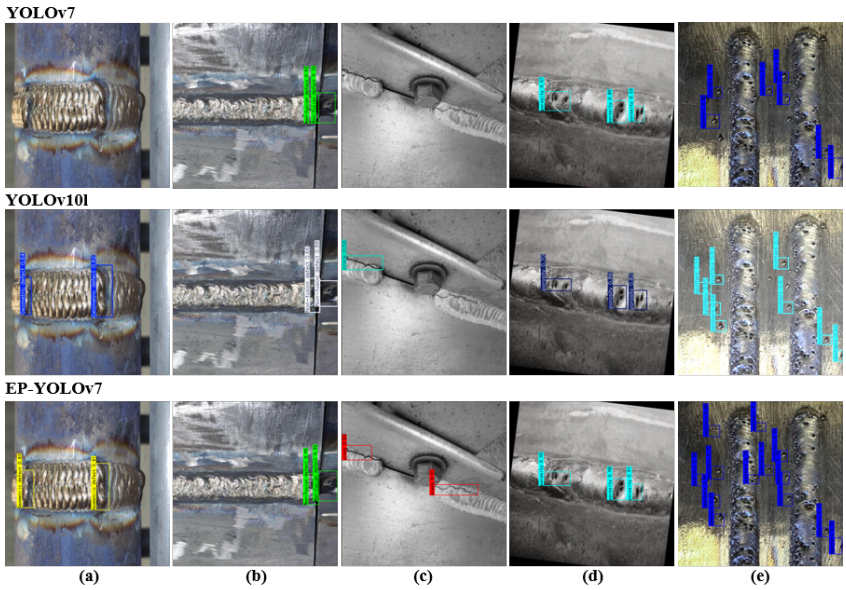


Fig. 11. Generalized experimental visualization results (a) Geometric defect (b) Non-fusion defect (c) crack (d) porosity (e) spatters

5. Conclusion

In this paper, we propose the EP-YOLOv7 model to achieve accurate localization and identification of weld proximity defects in steel thin plates. In the EM-ELAN and EM-ELAN-H modules, we incorporate the proposed EMCA module based on multi-scale convolution into the original module to effectively optimize the allocation of the feature weights in the channels and to improve the sensitivity of the model to the weld proximity defects and the feature extraction capability. In the P-BD Head module, the optimized decoupled head structure is used to effectively improve the detector's positioning and detection accuracy for weld proximity defects. Experimental results show that EP-YOLOv7 achieves superior detection performance compared with mainstream models. With only a 0.6MB increase in parameters and a minor rise in inference time, it achieves a 3.5% improvement in mAP over the baseline YOLOv7, along with notable gains in precision, F1 score, and recall. The model also exhibits good generalizability in publicly available datasets.

In the future, we will further expand the weld defect dataset to include additional materials (*e.g.,*aluminum, stainless steel, and galvanized steel) and more diverse industrial application scenarios (*e.g.,* shipbuilding, pressure vessel manufacturing, and rail transit equipment production) to enhance the model's adaptability and generalization. Moreover, we will continue to optimize EP-YOLOv7 through lightweight structural design, network pruning, and quantization strategies, aiming to improve detection efficiency and facilitate deployment in resource-constrained industrial environments.

Acknowledgments. This work was supported in part by the Provincial Natural Science Research Programs for Universities in Anhui Province (2024AH050234), in part by the open subject of Anhui Jianzhu University research platform (IMCM2023KF05), in part by the Chang'an University Special Funds for Basic Research Operating Expenses of the Central Universities (300102254501-202412), in part by the Scientific Research Program of Higher Education Institutions in Anhui Province (2024AH050245), in part by the Anhui Jianzhu University Research Reserve Project (2022XMK03), and in part by the Corporate Commissioned Project (JR202242).

References

1. Matthew Rumsa, Michele John, and Wahidul Biswas. Global steel decarbonisation roadmaps: Near-zero by 2050. *Environmental Impact Assessment Review*, 112:107807, 2025.
2. Xiaopeng Wang, Uwe Zscherpel, Paolo Tripicchio, Salvatore D'Avella, Baoxin Zhang, Juntao Wu, Zhimin Liang, Shaoxin Zhou, and Xinghua Yu. A comprehensive review of welding defect recognition from x-ray images. *Journal of Manufacturing Processes*, 140:161–180, 2025.
3. Changlu Xu, Linsheng Li, Jiwei Li, and Chuanbo Wen. Surface defects detection and identification of lithium battery pole piece based on multi-feature fusion and pso-svm. *Ieee Access*, 9:85232–85239, 2021.
4. Ruichen Ma, Jinglong Chen, Yong Feng, Zitong Zhou, and Jingsong Xie. Ela-yolo: An efficient method with linear attention for steel surface defect detection during manufacturing. *Advanced Engineering Informatics*, 65:103377, 2025.
5. Dorra Zaibi, Maroua Salhi, Khaoula Tbarki, and Riadh Ksantini. An incremental software defect detection model based on support vector machine. *Engineering Computations*, 42(1):76–95, 2025.

6. Yirga Kene Molla and Emebet Abeje Mitiku. Cnn-hog based hybrid feature mining for classification of coffee bean varieties using image processing. *Multimedia Tools and Applications*, 84(2):749–764, 2025.
7. Qianwen Ye, Yiwei Dong, Xiaoxin Zhang, Duo Zhang, and Siyi Wang. Robustness defect detection: improving the performance of surface defect detection in interference environment. *Optics and Lasers in Engineering*, 175:108035, 2024.
8. Xianghe Zou, Chongyang Wu, Hongen Liu, Zhangwei Yu, and Xianyan Kuang. An accurate object detection of wood defects using an improved faster r-cnn model. *Wood Material Science & Engineering*, 20(2):413–419, 2025.
9. Zhige He and Yuanqing He. As-faster-rcnn: An improved object detection algorithm for airport scene based on faster r-cnn. *IEEE Access*, 2025.
10. Xin Weng, Qihua Ma, Qilin Li, and Wenchao Wang. Improved mask r-cnn algorithm: Multi-ore detection and positioning based multi-sensor fusion in complex field environment. *Measurement*, 246:116602, 2025.
11. Ling Tan, Hui Wu, Zifeng Xu, and Jingming Xia. Multi-object garbage image detection algorithm based on sp-ssd. *Expert Systems with Applications*, 263:125773, 2025.
12. Beilong Chen, Mingjun Wei, Jianuo Liu, Hui Li, Chenxu Dai, Jinyun Liu, and Zhanlin Ji. Efs-yolo: a lightweight network based on steel strip surface defect detection. *Measurement Science and Technology*, 35(11):116003, 2024.
13. Wenqi Cui, Zhenye Li, Anning Duanmu, Sheng Xue, Yiren Guo, Chao Ni, Tingting Zhu, and Yajun Zhang. Ccg-yolov7: A wood defect detection model for small targets using improved yolov7. *IEEE Access*, 12:10575–10585, 2024.
14. Keji Mao, Runhui Jin, Lingkang Ying, Xingda Yao, Guanglin Dai, and Kai Fang. Sc-yolo: Provide application-level recognition and perception capabilities for smart city industrial cyber-physical system. *IEEE Systems Journal*, 17(4):5118–5129, 2023.
15. Yongbin Chen, Jingran Wang, and Guitang Wang. Intelligent welding defect detection model on improved r-cnn. *IETE Journal of Research*, 69(12):9235–9244, 2023.
16. Rijun Wang, Fulong Liang, Xiangwei Mou, Lintao Chen, Xinye Yu, Zhujing Peng, and Hongyang Chen. Development of an improved yolov7-based model for detecting defects on strip steel surfaces. *Coatings*, 13(3):536, 2023.
17. Quan Lu, Kehong Lin, and Linfei Yin. 3d attention-focused pure convolutional target detection algorithm for insulator defect detection. *Expert Systems with Applications*, 249:123720, 2024.
18. Qingqi Zhang, Xinbo Wang, Hao Wang, and Jie Li. Vehicle small target detection algorithm for remote sensing images based on improved yolov7-tiny. In *2024 5th International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pages 183–187. IEEE, 2024.
19. Jinlai Zhang, Wenjie Yang, Yumei Chen, Mingkang Ding, Huiling Huang, Bingkun Wang, Kai Gao, Shuhan Chen, and Ronghua Du. Fast object detection of anomaly photovoltaic (pv) cells using deep neural networks. *Applied Energy*, 372:123759, 2024.
20. Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: chasing higher flops for faster neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12021–12031, 2023.
21. Zhijin Li, Jinfeng Yan, Jie Zhou, Xiaozhen Fan, and Jiahui Tang. An efficient smd-pcba detection based on yolov7 network model. *Engineering Applications of Artificial Intelligence*, 124:106492, 2023.
22. Yi Zhang and Qingjian Ni. A novel weld-seam defect detection algorithm based on the s-yolo model. *Axioms*, 12(7):697, 2023.
23. Shunxiang Zhang, Meng Chen, Kuan-Ching Li, Hua Wen, and Liang Sun. Fire detection models based on attention mechanisms and multiscale features. *Computer Science and Information Systems*, (00):59–59, 2025.
24. Zihua Chen, Runmei Zhang, Meng-Yen Hsieh, Alireza Souri, and Kuan-Ching Li. Average sigmoid-tanh attention and multi-filter partially decoupled mechanism via yolov7 for detecting weld proximity defects. *Metallurgical and Materials Transactions B*, pages 1–15, 2025.

25. Yushan Zhao, Kuan-Ching Li, Shunxiang Zhang, and Tongzhou Ye. Hyperbolic graph attention network fusing long-context for technical keyphrase extraction. *Information Fusion*, 120:103061, 2025.
26. Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
27. Junpeng Xu, Xiangbo Zhu, Lei Shi, Jin Li, and Ziman Guo. Squeeze-and-excitation attention and bi-directional feature pyramid network for filter screens surface detection. *Journal of Electronic Imaging*, 33(4):043044–043044, 2024.
28. Zhaochao Li, Linxuan Xiao, Meiling Shen, and Xiya Tang. A lightweight yolov8-based model with squeeze-and-excitation version 2 for crack detection of pipelines. *Applied Soft Computing*, page 113260, 2025.
29. Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
30. Hajar Zoubir, Mustapha Rguig, Mohamed El Aroussi, Rachid Saadane, and Abdellah Chehri. Pixel-level concrete bridge crack detection using convolutional neural networks, gabor filters, and attention mechanisms. *Engineering Structures*, 314:118343, 2024.
31. Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.
32. Liang Tian Jiao, Pan Wei Guo, Bao Hong, and Pan Feng. Vehicle wheel weld detection based on improved yolo v4 algorithm. , 46(2):271–279, 2022.
33. Moyun Liu, Youping Chen, Jingming Xie, Lei He, and Yang Zhang. Lf-yolo: A lighter and faster yolo for weld defect detection of x-ray image. *IEEE sensors journal*, 23(7):7430–7439, 2023.
34. Zihua Chen, Yu Zheng, Ling-Huey Li, Tien-Hsiung Weng, Kuan-Ching Li, and Aneta Poniszewska-Maranda. An improved dilated-transposed convolution detector of weld proximity defects. *IEEE Access*, 2024.
35. Xiangqian Xu and Xing Li. Research on surface defect detection algorithm of pipeline weld based on yolov7. *Scientific reports*, 14(1):1881, 2024.
36. Yong Ren, Dong Liu, and Sanhong Gu. A lightweight defect classification method for latex gloves based on image enhancement. *Computer Science and Information Systems*, (00):7–7, 2025.
37. Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020.
38. Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

Runmei Zhang is a professor in the School of Electronic and Information Engineering at Anhui Jianzhu University. She received her PhD degree from the Hefei University of Technology in 2015. Her current research interests are robotics, artificial intelligence, and big data.

Jingwei Fan is a graduate student in the School of Mechanical and Electrical Engineering at Anhui Jianzhu University. His research interests include deep learning and image processing.

Zihua Chen (corresponding author) is a lecturer in the School of Electronic and Information Engineering at Anhui Jianzhu University. He received his PhD degree in the College

of Mechanical Engineering, Anhui University of Science and Technology (AUST), in 2022. His current research interests are the Internet of Things, multi-agent cooperative control, and computer vision.

Zhong Chen is a senior engineer in the School of Mechanical and Electrical Engineering at Anhui Jianzhu University. He received his PhD degree from the Hefei University of Technology in 2015. His current research interests are artificial intelligence, electrical control.

Bin Yuan is a lecturer in the School of Mechanical and Electrical Engineering at Anhui Jianzhu University. He received his PhD degree from the Hefei University of Technology in 2018. His current research interests are intelligent manufacturing and construction technology.

Received: September 22, 2025; Accepted: December 14, 2025.

Computer Science and Information Systems 23(1):277–297 https://doi.org/10.2298/CSIS250923008L

Adaptive Bandwidth Allocation via Uncertainty-Constrained Deep Reinforcement Learning

Li Wei, Wu Yong and Yan Dong

Suzhou Suneng Group Co., LTD
215004 Suzhou, China
{liwei_sz,wuy11,yandong_sz}@js.sgcc.com.cn

Abstract. With the rapid growth of network services, traditional static bandwidth allocation schemes can no longer meet the demands of multi-user, dynamic, and QoS-sensitive applications. Ensuring both efficiency and stability in bandwidth allocation remains a significant challenge, especially under high variability and uncertainty conditions. To address this, we propose a novel algorithm named Uncertainty-Constrained Stability-aware Deep Reinforcement Learning (UCS-DRL) for dynamic bandwidth allocation. UCS-DRL adopts a dual-policy architecture: a task policy that learns optimal bandwidth allocation decisions, and a stability policy guided by uncertainty-aware value estimation to identify and mitigate potential risky or unstable behaviors during deployment. Furthermore, the framework incorporates a curiosity-driven exploration mechanism based on Random Network Distillation, which enhances exploration efficiency by encouraging the agent to visit informative and under-explored states. Experimental results show that UCS-DRL achieves high bandwidth utilization and service quality while reducing policy volatility and risky actions, balancing performance and robustness in dynamic bandwidth allocation.

keywords: Dynamic Resource Allocation, Reinforcement Learning, Uncertainty Estimation, Stability-aware Control, Dual-policy Framework.

1. Introduction

In the context of the rapid development of smart substations, power systems are placing higher demands on the efficiency of bandwidth allocation and the intelligence of scheduling in communication networks[1]. The variety of services deployed within substations is continuously increasing[2]: on one hand, applications such as remote monitoring, relay protection signal transmission, and real-time equipment status reporting require stable and continuous bandwidth, with high sensitivity to transmission delay and reliability; on the other hand, event-driven services such as fault alarms and emergency command dispatch often generate large volumes of data in a short time, exhibiting strong time-criticality and transient high-bandwidth consumption. With emerging applications such as 4K/8K high-definition video[3], VR/AR-assisted inspection, and remote interactive maintenance, network traffic volume and dynamics are further intensified. Under such an environment characterized by concurrent multi-service operations and significantly different traffic patterns, how to achieve efficient, precise, and dynamic bandwidth allocation—while ensuring quality of service for diverse applications and overall system reliability—has become a key challenge for communication networks in smart substations[4].

Traditional bandwidth allocation strategies exhibit significant limitations in such complex service environments[5]. However, they cannot adapt in real time to competition among services and frequent traffic fluctuations, often leading to inefficient utilization or imbalanced allocation of resources. When sudden traffic surges occur, these methods usually fail to adjust promptly, resulting in congestion or interruption of critical services; conversely, in low-load scenarios, they often lead to bandwidth redundancy and waste. Meanwhile, modern networks, particularly smart substations, impose multiple constraints on bandwidth scheduling[6], such as those of latency, throughput, packet loss, and fairness, which are often conflicting. Traditional methods typically optimize a single objective, making achieving comprehensive trade-offs across multiple metrics difficult. More critically, real-world networks are subject to diverse uncertainties, including sudden traffic demands, link quality fluctuations, and external disturbances, and traditional strategies often fail to maintain stability under abnormal conditions. This shortcoming is particularly severe in critical infrastructures such as intelligent substations, where failure of allocation strategies at crucial moments may directly disrupt the stable and continuous operation of the power grid[7].

Recently, reinforcement learning (RL) has been widely applied in network scheduling and resource management[8]. It gradually optimizes strategies through interaction with the environment and has strong adaptability and generalization capabilities. In particular, policy gradient methods (such as PPO[9]) and value iteration methods (such as DQN[10]) demonstrate good learning capabilities in complex state spaces. However, most existing RL methods generally neglect to model the stability and uncertainty of policy deployment[11], leading to potential uncontrollable decisions by the trained policy when faced with unexpected scenarios such as sudden traffic spikes or abnormal states, thereby impacting service stability and network stability.

To address these challenges, we propose Uncertainty-Constrained Stability-aware Deep Reinforcement Learning (UCS-DRL) that explicitly incorporates uncertainty constraints to ensure stable and reliable policy deployment. UCS-DRL constructs a dual-module collaborative training framework comprising a task and stability policies. In UCS-DRL, the task policy employs the Deep Deterministic Policy Gradient(DDPG) method to achieve efficient bandwidth utilization. It features a pluggable interface to accommodate other high-performance policy structures. The stability policy incorporates a multi-scenario uncertainty bounding mechanism that integrates optimistic, pessimistic, and most likely behavioral scenarios. UCS-DRL enables a more comprehensive characterization of the potential fluctuation range in policy outputs and allows for proactive identification of extreme risk boundaries before action execution. Additionally, UCS-DRL combines a curiosity mechanism to expand the policy exploration space and enhance generalization capabilities in unknown states. Dynamically deploying UCS-DRL in the controller effectively improves throughput and service reliability while controlling end-to-end latency and deployment risks, achieving stable, efficient, and intelligent bandwidth resource allocation. Experimental results show that our method outperforms baselines in key metrics such as throughput, latency reduction, and adaptation capability, demonstrating superior performance and practical engineering value.

Our contributions are summarized as follows:

- We propose the UCS-DRL that achieves efficient and stable dynamic bandwidth allocation through a collaborative training framework of task and stability policies under uncertainty constraints.
- We design a pluggable task policy module to enhance resource allocation efficiency while maintaining training stability and convergence performance.
- We construct a stability policy based on multi-scenario uncertainty modeling that considers optimistic, pessimistic, and most likely behavioral estimates, balancing constraint and adaptability in policy output space.
- We combine internal policy uncertainty evaluation with external exploration signals from random network distillation to improve environmental awareness, policy generalization, and deployment robustness under distribution shifts.

The remainder of this paper is organized as follows: Related works are given in Section 2. The problem is modeled in Section 3. The details of UCS-DRL are explained in Section 4. The experiments and results are presented and analyzed in Section 5. Eventually, the paper ends in Section 6 with conclusions.

2. Related Work

2.1. Traditional Bandwidth Scheduling Methods

In early network bandwidth scheduling, allocation strategies were primarily based on static rules that divided bandwidth by user type, service priority, service level, or traffic weight[12]. For example, static scheduling mechanisms such as Weighted Round Robin (WRR)[13] and Weighted Fair Queuing (WFQ)[14] are widely used in traditional network devices, offering advantages such as simple implementation, low computational complexity, and controllable scheduling overhead. However, these methods heavily rely on manually designed rules, making it challenging to maintain efficient resource utilization and service quality assurance in complex business models and dynamic network conditions. To overcome the shortcomings of static methods in adapting to environmental changes, researchers have proposed a series of dynamic scheduling algorithms based on heuristic ideas. For example, Shortest Path First (SPF)[15] and Fastest Bandwidth Lowest Delay[16] strategies are applied in the joint selection of paths and bandwidth to improve overall network throughput and task completion efficiency. Meanwhile, fuzzy logic control has also been introduced into network resource scheduling, enhancing the decision-making flexibility of schedulers in uncertain environments. A typical example is the Fuzzy Bandwidth and Delay Guaranteed Routing Algorithm(FBDRA)[17] models bandwidth requirements and delay tolerance through a fuzzy inference system, thereby improving the satisfaction rate of QoS metrics while maintaining system stability. Although these heuristic methods demonstrate good practicality and performance in small-to-medium-sized network environments, they exhibit poor generalization capabilities and scalability when faced with real-world scenarios such as high-dimensional state spaces, heterogeneous multi-service requirements, and complex topological dynamic changes[18]. This has provided research motivation for developing more intelligent and adaptive scheduling algorithms.

2.2. Traffic Prediction and Proactive Resource Allocation

To address the traffic dynamics caused by concurrent multi-service operations in smart substations, researchers increasingly focused on proactive resource allocation methods[19] based on traffic prediction in recent years. Unlike traditional scheduling mechanisms that rely on real-time status feedback, proactive scheduling anticipates future traffic trends over a certain period and preemptively reserves bandwidth. It adjusts resources, enhancing the capability to guarantee service quality for critical applications. For instance, periodically reported equipment status data and scheduled video inspection tasks exhibit strong temporal regularity, making them suitable for high-precision traffic prediction through time series modeling[20]. For unexpected events such as fault alarms and protective actions, their occurrence probabilities and data volumes can be predicted by combining historical fault patterns with operational conditions, using statistical learning or machine learning methods. Previous studies have employed traditional time-series analysis methods such as ARIMA and Hidden Markov Models (HMM) for industrial communication traffic prediction. Recently, deep learning models including Long Short-Term Memory (LSTM)[21], Gated Recurrent Units (GRU), and Transformer[22] have demonstrated superior capabilities in capturing complex nonlinear traffic characteristics. Nevertheless, most existing methods still rely on offline training, making them difficult to adapt to the dynamic changes of substation operating environments. Moreover, they seldom consider the impact of prediction errors on the robustness of resource allocation, which limits their practical effectiveness in real-world systems.

2.3. Reinforcement Learning

With the development of deep reinforcement learning technology, its application in intelligent network control has received increasing attention[23]. Reinforcement learning has the ability to "learn through interaction," enabling it to gradually approach optimal strategies through trial-and-error exploration mechanisms[24]. It is naturally suited to network scheduling problems characterized by high-dimensional state spaces, complex system dynamics, and sparse rewards. In existing research, value-based methods such as DQN (Deep Q Network) have been widely used in discrete bandwidth allocation and routing selection scenarios[25]. For example, by encoding path and bandwidth allocation actions, a Q-network is trained to obtain the policy with the maximum cumulative reward, thereby optimizing system throughput or load balancing effects. Researchers have adopted policy gradient methods such as PPO (Proximal Policy Optimization) for scheduling scenarios with continuous actions to achieve refined resource control and rapid response adjustments[26]. In addition, to improve exploration efficiency and generalization capabilities, some studies have combined intrinsic incentive mechanisms such as Random Network Distillation(RND)[27] and Intrinsic Curiosity Module(ICM)[28] to guide agents to focus on insufficiently explored state regions and avoid falling into local optima. In multi-objective scheduling scenarios, some studies have introduced multi-task reinforcement learning architectures, incorporating multiple metrics such as bandwidth utilization, latency, service quality, and stability risks into a unified reward function to improve the system's overall performance[29].

In real-world network deployments, ensuring policy safety and robustness is critical. Safe reinforcement learning addresses this through constraint-based optimization and

uncertainty-aware methods[11]. Representative approaches like Penalized Proximal Policy Optimization (P3O)[30] formalize safety as cost constraints, while Robust RL frameworks [31,32] enhance robustness by explicitly accounting for model uncertainty or adversarial disturbances. Building on these foundations, our work introduces an uncertainty-constrained mechanism to jointly optimize performance and safety in network control.

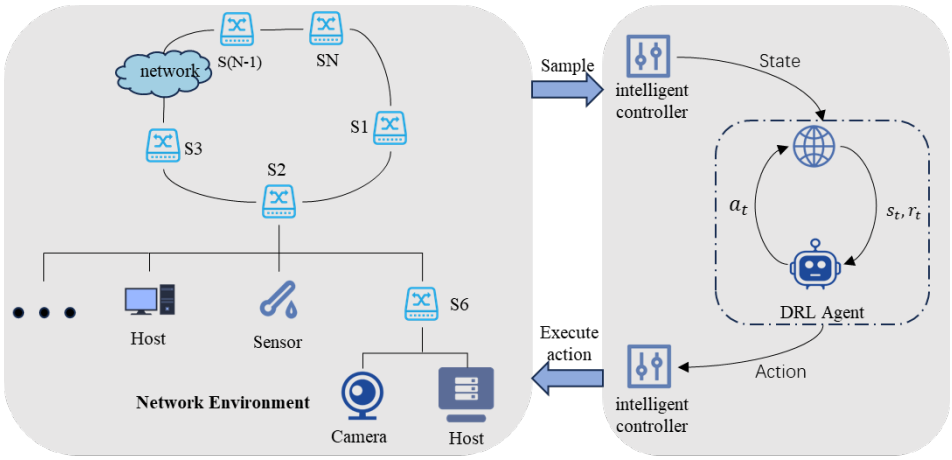


Fig. 1. Network topology: a ring backbone with tree-structured access networks, highlighting one representative access node and its connected end devices

3. System Model and Problem Formulation

3.1. System Model

We investigate the problem of dynamic bandwidth allocation at the access layer in smart substation communication networks. Considering that highly reliable ring-based backbone networks are widely adopted in practical engineering to ensure network connectivity, while actual bandwidth contention primarily occurs within the tree-structured subnetworks connected to nodes on the ring, we focus on the local bandwidth scheduling scenario under a single node. In this architecture, a communication node on the ring connects to multiple end devices—such as high-definition video cameras, protection relays, condition monitoring sensors, and local monitoring hosts—via local access links. These devices share the upstream link bandwidth of the node, forming a typical "multi-source to single-sink" traffic aggregation pattern.

To more realistically simulate such scenarios, we construct the network topology shown in Fig. 1 in a simulation environment. The topology consists of a ring-based backbone network, with several end-user nodes connected under each switching node, forming a ring-tree hybrid structure. We focus on one representative node in the network, connecting multiple end devices — including high-priority protection devices, high-bandwidth

video surveillance equipment, medium-priority status sensors, and a local monitoring host capable of data aggregation and command dispatch. These devices share an upstream link via a local switch to access the ring network, with the total link bandwidth set to $B_{total} = 100$ Mbps. Due to significant differences among services in delay sensitivity, data volume, and transmission continuity, the absence of an effective scheduling mechanism may lead to delays or packet loss for critical services due to bandwidth contention.

During the simulation, the system periodically collects real-time status information of each traffic flow, including transmission rate, queue length, end-to-end delay, and packet loss, serving as input for the bandwidth allocation policy. This sensing mechanism is implemented by deploying lightweight monitoring modules at local nodes, without requiring global topology information, thus meeting substations' localized and low-overhead scheduling requirements. Based on these local state feedbacks, the scheduler dynamically adjusts the bandwidth allocation for each traffic flow to ensure quality of service for high-priority applications while improving overall resource utilization efficiency. At each time step, the intelligent controller can obtain the following network state information: each user's bandwidth request volume $d_i(t)$; each link's delay $q_l^{delay}(t)$; the user's actual bandwidth usage in the previous cycle $u_i(t-1)$; and the user's service priority p_i. These observations are obtained through port queues, forwarding table statistics, or probe mechanisms via the switching device interface, with specific calculations as follows:
Bandwidth request estimation: Use the queue rate to estimate user i instantaneous bandwidth request:

$$d_i(t) \approx \frac{\Delta Q_i(t)}{\Delta t} = \frac{Q_i(t) - Q_i(t-\Delta t)}{\Delta t}\,. \tag{1}$$

Link delay estimation: Use the controller to actively detect the delay of link l:

$$q_l^{delay}(t) = \frac{1}{2} \cdot RTT_l(t), \tag{2}$$

where $RTT_l(t)$ measured by ICMP or probe packets.

Estimated actual bandwidth usage: The user's actual bandwidth usage in the previous cycle is estimated based on the byte change recorded in the forwarding table:

$$u_i(t-1) = \frac{\Delta Bytes_i(t-1)}{\Delta t}\,. \tag{3}$$

Through the above system design, we establish a simulation network environment that supports multi-user access, features dynamic link states, and reflects realistic bandwidth scheduling requirements. This environment serves as a foundational platform for training and evaluating reinforcement learning algorithms. Based on the typical "ring-tree" hybrid topology of smart substations, this model fully incorporates the service heterogeneity, traffic burstiness, and delay sensitivity present in practical communication scenarios, effectively capturing the complexity and uncertainty inherent in bandwidth scheduling at the access layer of power systems.

3.2. Problem Formulation

Based on the aforementioned system model, we formalize the bandwidth allocation task as a reinforcement learning problem and models it as a Markov Decision Process (MDP)[33].

At each discrete time step *t*, the controller needs to make a round of bandwidth allocation decisions based on the current network state to achieve optimal resource scheduling while ensuring network stability and user service quality. Let the set of terminal users be $U = 1, 2, ..., N$.

State space S: The system state at time step t is s_t, which includes user bandwidth requirements, link quality, historical usage, and priority, i.e.:

$$s_t = \{d(t), q(t), u(t), p\} \ . \tag{4}$$

Action space A: The controller selects bandwidth allocation actions based on the current state.

$$a_t = [a_1(t), a_2(t), ..., a_N(t)], \quad s.t. \sum_{i=1}^{N} b_i(t) \leq B_{total}, b_i(t) \geq 0 \, . \tag{5}$$

Reward function R: Measures the effectiveness of current bandwidth allocation, satisfies user bandwidth requirements (satisfaction), maintains good link quality, and utilizes bandwidth resources as efficiently as possible (resource utilization):

$$R_t = \lambda_1 \cdot R_{th}(t) - \lambda_2 \cdot R_{lat}(t) + \lambda_3 \cdot R_{ada}(t) \, , \tag{6}$$

where:

$$R_{th}(t) = \frac{1}{N} \sum_{i=1}^{N} \frac{b_i(t)}{d_i(t) + \varepsilon} \tag{7}$$

represents throughput score (positive). A value of this indicator closer to 1 denotes that it better meets user's need.

$$R_{lat}(t) = \frac{1}{L} \sum_{i=1}^{L} \frac{q_l^{delay}(t)}{q^{max}} \tag{8}$$

indicates delay penalty (negative), where L represents the number of links that are actively probed by the controller. This item is a value between 0 and 1. The higher the value, the more severe the delay.

$$R_{ada} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(a_i(t) \in A_i^{stable}(t)) \tag{9}$$

denotes the adaptation score (positive). A higher score indicates better adaptation, which is defined by two key behaviors: bandwidth allocation aligned with actual link capacity, and allocation actions remaining within predefined policy confidence boundaries under dynamic network conditions.

State transition function P: determined by dynamic changes in the network, including link load fluctuations, traffic release, and sudden congestion, and exhibits a certain degree of uncertainty.

Discount factor $\gamma \in (0, 1)$: indicates the discount for future rewards.

The ultimate objective is to solve for that optimal policy that maximizes the desired long-term cumulative discount reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum \gamma^t r_t\right] \, . \tag{10}$$

The complexity of this problem lies in the fact that the user demand changes rapidly, the link state fluctuates frequently, and the traditional static bandwidth allocation policy is difficult to adapt. The introduction of a reinforcement learning framework can utilize the continuous interaction process between the controller and the environment to autonomously learn adaptive dynamic strategies and achieve fine-grained and efficient allocation of bandwidth resources.

4. Methodology

4.1. Overview

To address dynamic bandwidth allocation challenges, we propose UCS-DRL, a stability-aware deep reinforcement learning algorithm based on uncertainty constraints, designed for intelligent and robust scheduling of network resources. The algorithm adopts a dual-policy coordination framework, integrating a task policy for performance optimization with a stability policy for risk-constrained decision-making. By jointly evaluating task rewards and behavioral risks, UCS-DRL ensures high network performance while enhancing deployment robustness. The task policy leverages high-performance reinforcement learning architectures to generate efficient bandwidth allocation sequences. The stability policy employs a multi-scenario uncertainty bounding mechanism[34], adapted from project risk analysis, to estimate policy behavior under optimistic, pessimistic, and most likely conditions - ensuring performance remains within stable operational margins. To mitigate over-constraint and encourage exploration of informative states, we integrate a curiosity-driven exploration mechanism that enriches the data space without compromising stability. Guided by the task policy, the stability policy, enhanced with uncertainty-aware risk screening and curiosity exploration, serves as the final decision module. The two policies are functionally complementary and trained asynchronously, forming the core of UCS-DRL. The algorithm's workflow is illustrated in Fig. 2, and its pseudocode is presented in Algorithm 1.

4.2. Task Policy

A performance-oriented efficient policy optimization mechanism. We use the deep deterministic policy gradient algorithm (DDPG)[27] as the core policy framework to achieve dynamic perception and response control of network status. DDPG is an actor-critic-based reinforcement learning algorithm suitable for high-dimensional, continuous control tasks, making it ideal for bandwidth allocation with adjustable granularity.

For modeling, we encode the current network state as a vector s_t, including key variables such as user demand, link quality, previous allocation records, and user priority. The actor network takes s_t as input and outputs a continuous action vector a_t, i.e., the bandwidth allocation recommendation. To ensure the feasibility of actions, a softmax operation is applied and multiplied by the total bandwidth B_{total} to satisfy the bandwidth constraint.

DDPG improves training stability via target networks and experience replay. The critic network is updated using the Bellman equation to estimate the Q-value function:

$$Q(s_t, a_t) = r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1}))\,, \tag{11}$$

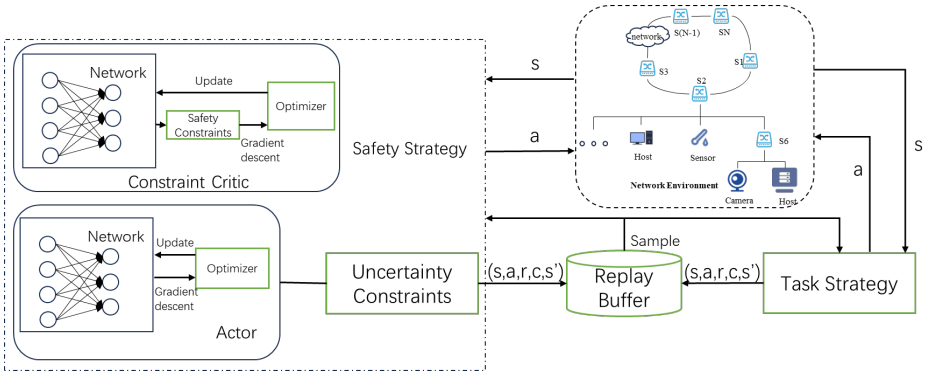


Fig. 2. Illustration of overall architecture of UCS-DRL that primarily consists of four components: the environment,the task policy utilizing the off-policy algorithm, the stability policy employing the actor-critic framework, and the data space

where $\mu'(s)$ and $Q'(s,a)$ are the target actor and critic networks, respectively, and r_t is the immediate reward based on bandwidth matching and link stability. The actor network is optimized by maximizing the expected Q-value gradient:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}\left[\nabla_a Q(s,a|\theta^Q)|_{a=\mu(s)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)\right] . \tag{12}$$

UCS-DRL effectively enhances the performance of bandwidth allocation strategies while maintaining action continuity.

4.3. Stability Policy

Stability-oriented value constraint policy mechanism. After the reinforcement learning agent has constructed a policy with high task performance but possible stability risks, the next step is to further guide it to generate a policy with higher stability while ensuring performance. In the bandwidth allocation problem proposed in this paper, stability is reflected in the resource allocation process by avoiding key link overload and ensuring the service continuity of high-priority tasks. Therefore, we introduce a stability policy framework to impose soft stability constraints during the training process to constrain the direction of policy updates. Specifically, the stability policy acts as an auxiliary critic that estimates the long-term risk of state-action pairs, penalizing decisions that could lead to future instability (e.g., persistent congestion or service interruption). The penalty term is dynamically weighted and incorporated into the policy gradient update, allowing the agent to continuously trade off between maximizing task rewards and minimizing stability violations during training. The stability policy framework is designed based on a value function optimization architecture, utilizing a dual-network structure: the actor network outputs deterministic actions, while the evaluator network assesses the value of state-action pairs. Additionally, we introduce a constrained evaluator network[35] that integrates real-time stability costs into the traditional value function, i.e.:

$$Q_{rc} = r + \gamma \max_{a'} Q_{rc}(s', a') - c\,, \tag{13}$$

where r is the immediate reward, c is the immediate cost incurred due to violations of bandwidth constraints (such as overload, conflicts, or unmet priorities), and γ is the discount factor. The constraint value function influences the actor's policy through gradient updates, enabling it to perceive and avoid potential stability risks during the decision-making process. During training, both the constraint commentator network and the target network adopt soft update strategies to ensure training stability. The agent policy is optimized based on the gradient of the constraint value function, thereby achieving a resource allocation policy that balances efficiency and stability. The resulting policy effectively controls the expansion direction of the data space, making the reinforcement learning model more controllable and stable during bandwidth scheduling, particularly suitable for dynamic and complex network environments.

4.4. Uncertainty Constraints and Diversity

In reinforcement learning-driven bandwidth allocation problems, uncertainty in the environment and policy may lead to risky behaviors such as resource conflicts, performance degradation, or policy overfitting. To address this, we introduce an uncertainty constraint mechanism that combines the multi-scenario uncertainty bounding method in project management to dynamically evaluate the lower bound threshold of task performance, thereby balancing the agent's pursuit of performance while constructing a secure data space. The mechanism bounds performance uncertainty by integrating optimistic, pessimistic, and most-likely scenario evaluations, providing a probabilistic safeguard against over-optimistic policy updates. Specifically, we use the Beta distribution model to calculate the task performance threshold μ_r, which is updated every K episodes during training. Based on the maximum, minimum, and average values of episodic returns within that cycle, we derive three scenario-based estimates: T_o(optimistic) reflects the maximum achievable throughput under ideal network conditions such as minimal interference and full resource availability; T_p(pessimistic) represents the minimum sustainable performance under worst-case disturbances including congestion or traffic bursts; T_m(most likely) captures the expected performance under typical operational loads and serves as the baseline for stable allocation. These estimates are then used to compute the following formula:

$$\mu_r = \frac{T_o + 4T_m + T_p}{6}\,. \tag{14}$$

This dynamic threshold is used to filter samples in the experience pool, controlling the quality of the data space and guiding the policy to maintain high bandwidth allocation efficiency under stability constraints. To alleviate the inhibition of policy exploration capabilities caused by excessive constraints, we introduce an intrinsic motivation mechanism to encourage agents to actively explore unknown state spaces.There have been many excellent results in exploration, such as the curiosity-driven mechanism ICM [36] and the never give up (NGU) [37]. The way of intrinsic reward-driven exploration in reinforcement learning mainly realizes these exploration algorithms. In our experiments, we employ Random Network Distillation (RND) [38] to score states for novelty. RND consists of a fixed set of random feature extraction networks $f(s)$ and a learnable network $\hat{f}(s)$, where the prediction error represents the unfamiliarity of the state:

$$r_t^{\text{curiosity}} = \|f(s_t) - \hat{f}(s_t)\|^2\,. \tag{15}$$

Curiosity rewards are linearly weighted into task rewards to form the final optimization goal:

$$r_t^{\text{final}} = r_t^{\text{task}} + \lambda_{\text{curiosity}} \cdot r_t^{\text{curiosity}}. \tag{16}$$

The task policy dominates sampling and updates the policy network, while the stability policy performs offline constraint screening on behavior samples to form candidate strategies for deployment. The curiosity module is embedded in the task policy in the form of rewards, encouraging agents to move beyond their normal behavior areas and form a healthy data flow cycle. Ultimately, the controller will use only strategies that meet both performance and stability requirements for actual network bandwidth allocation tasks.

Algorithm 1 Uncertainty-Constrained Stable Deep Reinforcement Learning

1: **Input:** Network topology E
Initialize constraint critic network $Q_c(s, a|\theta^{Q_c})$ and actor $\mu(s|\theta^{\mu})$
Initialize target constraint network Q_c' and μ'
Replay buffer R, the task policy π_{task}, the stability policy π_{stable}, the list $ep_min_return_list$, the task performance constraint threshold ep_min_return in Algorithm 2
2: **Output:**π_{stable}
3: **for** $episode$ = 1,N **do**
4: Initialize a random process $\mathcal{N}$ for action exploration
5: Receive initial observation state s_0
6: **for** t = 1,T **do**
7: Select action $a_t = \mu(s_t|\theta^{\mu}) + \mathcal{N}_t$ according to the current policy π_{stable} and exploration noise
8: Execute action a_t and observe reward r_t, cost c_t, next state s_{t+1}
9: Calculate curiosity rewards: $r_t^{\text{curiosity}} = \|f(s_t) - \hat{f}(s_t)\|^2$
10: Calculate total rewards: $r_{total} = r_t + \beta \cdot r_{curiosity}$
11: **if** $r_t \geq ep_min_return$ **then**
12: $\Pi_{stable} = \Pi_{stable} \cup (s_t, a_t, r_{total}, c_t, s_{t+1})$
13: **end if**
14: Set $y_i = r_{total} + \gamma * Q_c'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q_c'}) - c_i$
15: Update constraint critic by minimizing the loss: $L = \frac{1}{N}\sum_i \left(y_i - Q_c\left(s_i, a_i \mid \theta^{Q_c}\right)\right)^2$
16: Update the actor policy π_{stable} using the sampled policy gradient
17: Update the target network:
$\theta^{Q_c'} = \tau\theta^{Q_c} + (1-\tau)\theta^{Q_c'}$
$\theta^{\mu'} = \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$
18: **end for**
19: **end for**

In the Algorithm 1, it's crucial to highlight that we utilize the task performance threshold calculated with uncertainty as a screening criterion (Line 9-11) for the target data space. Additionally, we incorporate the immediate cost as a soft stability constraint within the expected Q value in the stability policy(Line 12). This process culminates in the generation of a stability data space that takes task performance considerations into account.

Algorithm 2 Uncertainty Constraint Mechanism

Input: The list $ep_min_return_list$ consisting of the minimum per-episode returns of K episodes is used as input
Output: The expected minimum return per episode ep_min_return as a task performance constraint for the next K episodes
Sort the list $ep_min_return_list$ in ascending order in place
$T_p = ep_min_return_list[0]$
$T_o = ep_min_return_list[-1]$
$T_m = \text{avg}(ep_min_return_list)$
$ep_min_return = (T_p + T_m \times 4 + T_o)/6$

5. Experiments and Results

5.1. Experimental Setups

To validate the effectiveness and robustness of the proposed uncertainty-constrained stable deep reinforcement learning bandwidth allocation algorithm (UCS-DRL) in real-world network scenarios, we conducted systematic simulation experiments. All experiments were conducted on a unified hardware platform and simulation environment to ensure fairness in comparison and reproducibility of the results. The experiments were conducted in two phases: training and deployment. During the training phase, we conducted training in cycles of 100 steps per round, totaling 1,000 training cycles, and recorded algorithm performance changes every 10 rounds to observe convergence trends. During the deployment phase, we selected the converged strategies from the algorithms, conducted independent tests, recorded the average metric results per round, and statistically analyzed the final stable performance to demonstrate practical deployability.

In terms of hyperparameter settings, the discount factor γ is set to 0.98, and the batch size is set to 128. The soft update coefficient τ of the target network is set to 0.005, while the learning rate of the actor network is set to 3×10^{-4} and that of the critic network is set to 3×10^{-3}. Additionally, the number of hidden layers for both the actor and critic networks is 2. A curiosity exploration mechanism based on Random Network Distillation (RND) is introduced to assist in expanding the training data space for the policy, with the exploration reward ratio set to 0.01.

5.2. Baselines

In order to comprehensively evaluate the performance of the UCS-DRL algorithm, we have selected three representative comparison methods, covering static resource allocation strategies, fuzzy logic control methods, and typical deep reinforcement learning algorithms. These methods are widely used in bandwidth scheduling and QoS routing optimization and other related research fields, and have a good representative comparison.
Static:A static scheduling policy with fixed proportional bandwidth allocation.
DDPG[27]: A classical deep reinforcement learning method.
IFRA-GLB[39]: A method based on fuzzy logic and reinforcement learning.
UCS-DRL: An innovative algorithmic framework proposed in this paper.

5.3. Results and Analysis

Reward Curve Analysis. Fig. 3 illustrates the cumulative reward curves over training episodes for the proposed UCS-DRL method and two baseline algorithms, DDPG and IFRA-GLB. In the initial stages of training, all methods exhibit substantial reward fluctuations, which is attributed to the agent's exploratory behavior in an uncertain environment. As training progresses, however, the proposed UCS-DRL approach demonstrates a significantly faster convergence rate and achieves a higher asymptotic reward plateau compared to the baselines. This superior performance highlights the effectiveness of incorporating uncertainty-based constraints into the decision-making process, enabling the agent to avoid high-variance or sub-optimal actions early in training and thus accelerate learning.Moreover, the stability of the reward curve in the later episodes aligns closely with the stabilization of key network performance metrics such as throughput and latency, indicating that the designed reward function accurately reflects real-world system performance. The shaded regions around each curve represent the standard deviation across multiple runs, further confirming that UCS-DRL not only achieves higher average rewards but also maintains greater consistency and robustness during training. In contrast, both DDPG and IFRA-GLB show slower convergence and lower final performance, underscoring the advantage of the uncertainty-aware framework in guiding more efficient and reliable policy optimization.

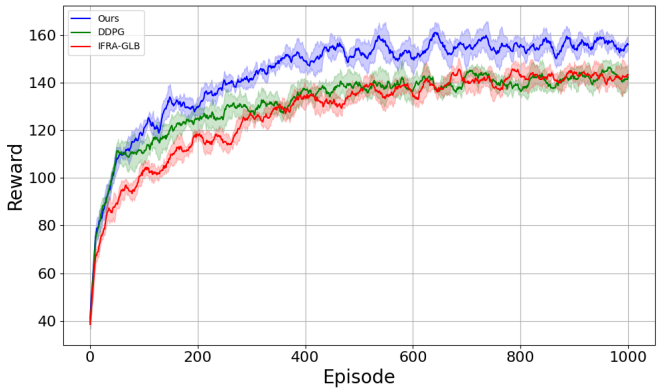


Fig. 3. Reward curves of different algorithms during training

Throughput performance. UCS-DRL demonstrates stable and continuous performance improvement throughout the training process. As shown in Fig. 4, the average network throughput increases steadily from an initial 20 Mbps to a final stabilized value of 94.5 Mbps, achieving approximately 94.5 percent of the total link capacity of 100 Mbps. This result significantly outperforms the baseline methods. DDPG and IFRA-GLB stabilize at around 82 and 86 Mbps, respectively. At the beginning of training, the agent's policy is suboptimal due to lack of experience. This leads to inefficient bandwidth allocation, underutilized links, and low throughput. However, UCS-DRL exhibits faster convergence and smaller fluctuations in the later stages compared to the benchmark algorithms.

This behavior indicates that the curiosity-driven exploration mechanism effectively encourages the agent to explore underutilized state action spaces, helping to avoid premature convergence to local optima. The task oriented policy framework guides the DDPG based actor critic structure to prioritize high impact actions. These include allocating sufficient bandwidth to high priority protection flows while opportunistically serving high bandwidth video streams. This guidance accelerates policy optimization. The integration of uncertainty aware risk control further prevents over allocation to bursty or unreliable flows, maintaining high link utilization without inducing congestion. The sustained high throughput achieved by UCS-DRL reflects its ability to dynamically balance exploration and exploitation, efficiently utilize available bandwidth, and adapt to time-varying traffic patterns. This makes it particularly suitable for smart substation networks, where both high resource efficiency and service reliability are critical.

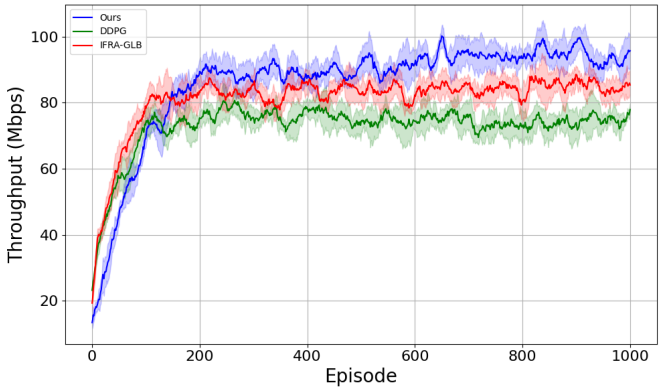


Fig. 4. Throughput performance comparison under different methods

Latency performance. UCS-DRL effectively avoids over-scheduling of highly loaded links by introducing uncertainty constraints and a risk-aware control mechanism. As illustrated in Fig. 5, the average latency of UCS-DRL decreases steadily from an initial 100 ms to 25 ms during the early training phase. Eventually, it converges to approximately 22.3 ms with minimal fluctuation. This smooth convergence indicates that the agent quickly learns an efficient bandwidth allocation policy that prioritizes delay-sensitive services while maintaining system stability under uncertain traffic conditions. In contrast, DDPG and IFRA-GLB exhibit significant oscillations in latency during the late stages of training. DDPG, which relies solely on reward shaping without explicit risk modeling, tends to over-exploit high-throughput paths, leading to temporary congestion and bursty latency spikes. IFRA-GLB, although it considers global load balancing, fails to account for traffic uncertainty and sudden bursts (e.g., fault alarms or video stream initiation), resulting in unstable performance and delayed response to dynamic changes. Moreover, UCS-DRL's uncertainty-aware design enables proactive bandwidth reservation for critical services based on predicted traffic patterns and confidence intervals. This not only prevents buffer overflow in high-priority queues but also reduces unnecessary waiting

time for time-sensitive data, which are essential for mission-critical applications in smart substations.

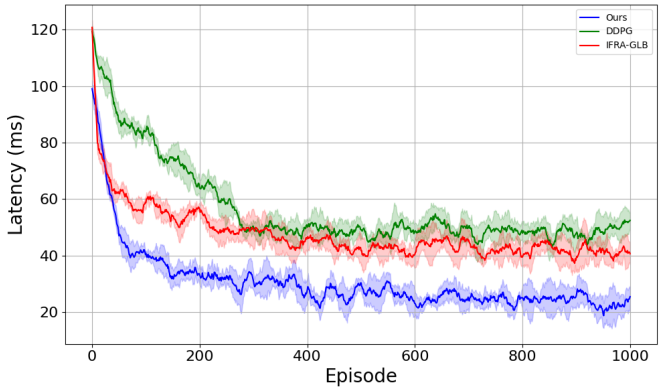


Fig. 5. Latency performance comparison under different methods

Adaptation performance. A higher Adaptation Rate indicates better system capability in dynamically adjusting to network conditions while adhering to stability and reliability constraints. These include avoiding high packet loss links, respecting bandwidth limits, and preventing congestion. UCS-DRL achieves outstanding performance in this metric. By incorporating a multi-scenario uncertainty bounding mechanism and a risk lower bound filtering operation during policy evaluation, UCS-DRL effectively suppresses actions with high uncertainty. This ensures that the selected actions are not only efficient but also compliant with operational stability requirements. As shown in Fig. 6, UCS-DRL maintains an average Adaptation Rate of approximately 97.5% in the test deployment phase, significantly outperforming DDPG (87.9%) and IFRA-GLB (91.4%). Further analysis reveals that the embedded risk control mechanism not only enhances the stability of the learned policy during deployment but also guides the training process toward the "high benefit low risk" region of the solution space. This demonstrates the effective synergy between reinforcement learning and stability-aware constraints. Unlike DDPG, which often generates aggressive actions under uncertainty, UCS-DRL prioritizes reliable and feasible decisions, making it well suited for mission-critical power communication environments.

5.4. Ablation study

Ablation study analysis on the curiosity-driven module reveals that removing the RND module leads to significant performance degradation, manifested through reduced exploration efficiency, slower convergence speed, and deteriorated policy quality. As shown in Fig. 7, without RND, the model exhibits delayed exploration progress during early training (0-400 episodes), with insufficient state-space coverage causing the policy to converge to local optima. While the complete model stabilizes after approximately 400 episodes, the ablated version fails to achieve comparable performance even after 1000 episodes,

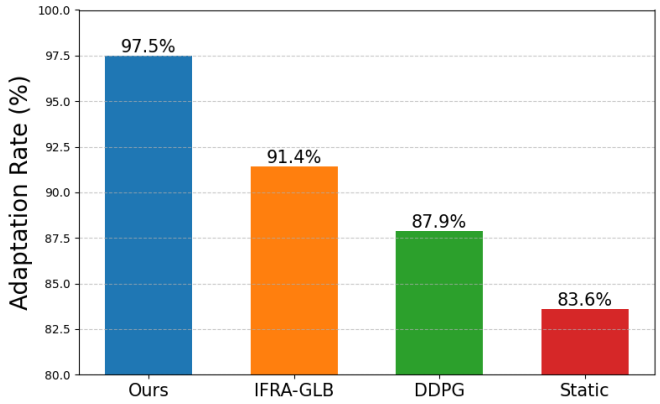


Fig. 6. Adaptation rate comparison in the deployment phase

demonstrating higher decision variance and lower adaptability. These results verify that in dynamic bandwidth allocation scenarios, the RND module effectively promotes comprehensive exploration of the network state space through its state novelty scoring mechanism, thereby preventing the resource allocation strategy from becoming overly conservative due to stability constraints. The prediction error-driven intrinsic reward provides essential exploratory momentum for optimizing bandwidth allocation strategies. By discovering non-typical resource allocation patterns, it significantly enhances the policy's adaptability and robustness in complex network environments, establishing itself as a key component in maintaining an efficient data flow cycle for bandwidth allocation.

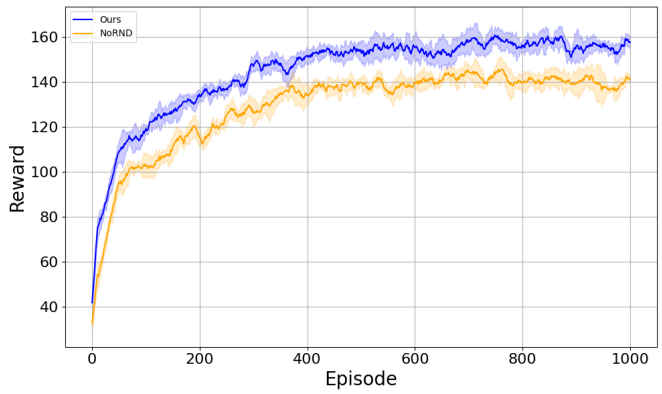


Fig. 7. Performance comparison of ablation studies

5.5. Deployment Evaluation Results

To further evaluate the generalization and stability of the trained policies, we conduct a comprehensive evaluation in the test phase, where the learned models are applied to previously unseen traffic scenarios. The performance is assessed across four key metrics: throughput, average latency, bandwidth utilization, and adaptation rate. Results are summarized in Table 1. Our proposed UCS-DRL algorithm outperforms all baseline methods across all metrics. It achieves a throughput of 91.3 Mbps and a bandwidth utilization of 91.3%, demonstrating efficient resource scheduling under diverse traffic patterns. More importantly, UCS-DRL achieves an adaptation rate of 95.2%, significantly higher than DDPG (87.9%) and IFRA-GLB (91.4%), indicating its strong ability to avoid risky actions—such as overloading links or violating bandwidth limits—while maintaining high performance. In terms of latency, UCS-DRL achieves the lowest average delay of 23.7 ms, which is critical for time-sensitive services in smart substations. This reflects its effectiveness in prioritizing high-priority flows and avoiding congestion through proactive bandwidth allocation. Fig. 8 presents a bar chart comparing the performance of all methods across the four evaluation metrics in the test phase. While static policy methods achieve moderate results in known scenarios, they lack adaptability and fail to respond effectively to dynamic and unseen conditions. In contrast, learning-based methods, particularly UCS-DRL, exhibit superior generalization and robustness due to its uncertainty-aware design and risk-constrained learning framework. These results confirm that UCS-DRL not only learns an efficient scheduling policy during training but also generalizes well to new scenarios, achieving a favorable balance between performance and stability. This makes it highly suitable for intelligent bandwidth management in complex and dynamic substation communication environments.

The above experimental results show that UCS-DRL has superior performance in terms of throughput efficiency, service delay control and adaptability, which proves the adaptability, stability and engineering practicability of the proposed algorithm in the complex dynamic network environment. Its task-stability dual-policy cooperative mechanism and uncertainty-aware design jointly construct a set of bandwidth allocation policy framework with learning capability, stability control, and stable output capability, which has good potential for promotion in future networks.

Table 1. Comparison of Algorithm Performance

Algorithm	Throughput (Mbps)	Latency (ms)	Utilization (%)	Adaptation (%)
Ours	**91.3±1.2**	**18.6±0.8**	**87.9±0.9**	**97.5±0.6**
IFRA-GLB	78.4±2.1	26.7±1.5	74.1±1.7	91.4±1.2
DDPG	83.7±1.8	22.3±1.2	81.5±1.4	87.9±1.5
Static	86.1±0.9	20.1±0.7	84.2±0.8	83.6±1.8

6. Conclusion

In this paper, we address the key challenge of balancing high performance with operational stability in dynamic bandwidth allocation by proposing a stable deep reinforce-

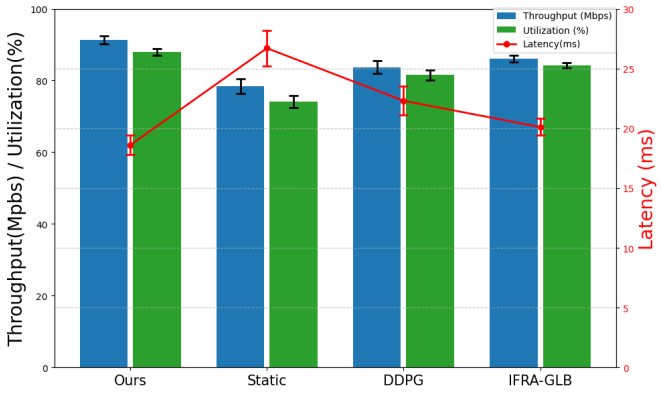


Fig. 8. Algorithm performance comparison in the deployment phase

ment learning algorithm. Our method employs a pluggable dual-policy framework: a performance-oriented task policy that maximizes bandwidth utilization and allocation efficiency, and a stability policy that constrains high-risk decisions via multi-scenario uncertainty modeling (optimistic, pessimistic, and most likely) and a value-based conservative critic, thereby directly mitigating the instability risks inherent in RL-based control. To enhance exploration and avoid local optima, we integrate a curiosity-driven mechanism that guides the agent toward unfamiliar states in early training, improving policy generalization. The reward function jointly optimizes throughput, delay, and stability adaptation rate, enabling balanced learning between performance and deployment robustness.

We evaluate our approach in a simulated multi-user environment. Extensive experiments demonstrate that our method outperforms baselines in both task performance and stability maintenance, significantly reducing policy violations during resource allocation. Ablation studies confirm the critical contributions of the curiosity module and uncertainty estimation mechanism. Importantly, our framework imposes minimal constraints on the base policy and is highly compatible with diverse task settings. We also acknowledge limitations including simulation-based validation and focus on single-node scheduling. Future work will explore multi-agent reinforcement learning for cross-node coordination and test the method on physical network hardware and more complex topologies.

Acknowledgments. This work was supported by Science and Technology Project of State Grid Jiangsu Electric Power Co., Ltd., and Science and Technology Project of Suzhou Suneng Group Co., Ltd. under Grant SGSZSNJTKJJS2500979.

References

1. José Gaspar, Tiago Cruz, Chan-Tong Lam, and Paulo Simões. Smart substation communications and cybersecurity: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 25(4):2456–2493, 2023.
2. Weiti Lv. Research on network application automation system based on computer artificial intelligence technology. In *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, pages 1934–1938. IEEE, 2023.

3. Xiaodong Qiao, Naichen Yan, and Lu Zhang. Design and implementation of substation monitoring system based on harmonyos and broadband-narrowband wireless communication technology. In *2025 2nd International Conference on Smart Grid and Artificial Intelligence (SGAI)*, pages 1484–1488. IEEE, 2025.
4. R Mohandas and D John Aravindhar. An intelligent dynamic bandwidth allocation method to support quality of service in internet of things. *International Journal of Computing*, 20(2):254–261, 2021.
5. G Senthilkumar, KN Madhusudhan, Y Jeyasheela, and P Ajitha. A novel blockchain enabled resource allocation and task offloading strategy in cloud computing environment. *Automatika*, 65(3):973–982, 2024.
6. Nurshazlina Suhaimy, Nurul Asyikin Mohamed Radzi, Wan Siti Halimatul Munirah Wan Ahmad, Kaiyisah Hanis Mohd Azmi, and MA Hannan. Current and future communication solutions for smart grids: A review. *IEEE Access*, 10:43639–43668, 2022.
7. Lan-Huong Nguyen, Van-Linh Nguyen, Ren-Hung Hwang, Jian-Jhih Kuo, Yu-Wen Chen, Chien-Chung Huang, and Ping-I Pan. Toward secured smart grid 2.0: Exploring security threats, protection models, and challenges. *IEEE Communications Surveys & Tutorials*, 27(4):2581–2620, 2025.
8. Hoa Tran-Dang, Sanjay Bhardwaj, Tariq Rahim, Arslan Musaddiq, and Dong-Seong Kim. Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues. *Journal of Communications and Networks*, 24(1):83–98, 2022.
9. John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, pages 1–12, 2017.
10. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
11. Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11216–11235, 2024.
12. Liqing Liu, Xiaoming Yuan, Decheng Chen, Ning Zhang, Haifeng Sun, and Amir Taherkordi. Multi-user dynamic computation offloading and resource allocation in 5g mec heterogeneous networks with static and dynamic subchannels. *IEEE Transactions on Vehicular Technology*, 72(11):14924–14938, 2023.
13. Ahmed Mohammed, Nor Fadzilah Abdullah, Sameer Alani, Othman S Alheety, Mohammed Mudhafar Shaker, Mohammed Ayad Saad, and Sarmad Nozad Mahmood. Weighted round robin scheduling algorithms in mobile ad hoc network. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–5. IEEE, 2021.
14. Mays A Mawlood and Dhari Ali Mahmood. Performance analysis of weighted fair queuing (wfq) scheduler algorithm through efficient resource allocation in network traffic modeling. *Journal of Communications Software and Systems*, 20(3):266–277, 2024.
15. Mohit Mittal, Celestine Iwendi, Suleman Khan, and Abdul Rehman Javed. Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using levenberg-marquardt neural network and gated recurrent unit for intrusion detection system. *Transactions on Emerging Telecommunications Technologies*, 32(6):1–16, 2021.
16. Inayat Ali, Seungwoo Hong, and Taesik Cheung. Quality of service and congestion control in software-defined networking using policy-based routing. *Applied Sciences*, 14(19):1–13, 2024.
17. Jianhu Gong and Hamed Nazari. A fuzzy bandwidth and delay guaranteed routing algorithm for performance enhancement of video conference over mpls networks. *Journal of Ambient Intelligence and Humanized Computing*, 14(6):7079–7090, 2023.
18. Yingnan Deng. A reinforcement learning approach to traffic scheduling in complex data center topologies. *Journal of Computer Technology and Software*, 4(3):1–6, 2025.

19. Torana Kamble, Sanjivani Deokar, Vinod S Wadne, Devendra P Gadekar, Hrishikesh Bhanudas Vanjari, and Purva Mange. Predictive resource allocation strategies for cloud computing environments using machine learning. *Journal of Electrical Systems*, 19(2):68–77, 2023.
20. Bo Wu and Yifan Hu. Analysis of substation joint safety control system and model based on multi-source heterogeneous data fusion. *IEEE Access*, 11:35281–35297, 2023.
21. Wei Sun, Pengyu Li, Zhi Liu, Xue Xue, Qiyue Li, Haiyan Zhang, and Junbo Wang. Lstm based link quality confidence interval boundary prediction for wireless communication in smart grid. *Computing*, 103(2):251–269, 2021.
22. Yang Chen, Jia Hao, Yu Peng, and Hongyan Xia. Transformer-based performance prediction and proactive resource allocation for cloud-native microservices. *Cluster Computing*, 28(9):568–590, 2025.
23. Linqiang Huang, Miao Ye, Xingsi Xue, Yong Wang, Hongbing Qiu, and Xiaofang Deng. Intelligent routing method based on dueling dqn reinforcement learning and network traffic state prediction in sdn. *Wireless Networks*, 30(5):4507–4525, 2024.
24. Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, pages 16828–16847. PMLR, 2022.
25. Jiawei Xu, Yufeng Wang, Bo Zhang, and Jianhua Ma. A graph reinforcement learning based sdn routing path selection for optimizing long-term revenue. *Future Generation Computer Systems*, 150:412–423, 2024.
26. Abdelhak Bentaleb, Mehmet N Akcay, May Lim, Ali C Begen, and Roger Zimmermann. Bob: Bandwidth prediction for real-time communications using heuristic and reinforcement learning. *IEEE Transactions on Multimedia*, 25:6930–6945, 2022.
27. David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 387–395, Bejing, China, 22–24 Jun 2014. PMLR.
28. Hongliang Zeng, Ping Zhang, Fang Li, Chubin Lin, and Junkang Zhou. Ahegc: Adaptive hindsight experience replay with goal-amended curiosity module for robot control. *IEEE Transactions on Neural Networks and Learning Systems*, 35(11):16602–16615, 2024.
29. Yuqing Cheng, Zhiying Cao, Xiuguo Zhang, Qilei Cao, and Dezhen Zhang. Multi objective dynamic task scheduling optimization algorithm based on deep reinforcement learning. *The Journal of Supercomputing*, 80(5):6917–6945, 2024.
30. Linrui Zhang, Li Shen, Long Yang, Shixiang Chen, Bo Yuan, Xueqian Wang, and Dacheng Tao. Penalized proximal policy optimization for safe reinforcement learning. In *International Joint Conference on Artificial Intelligence*, pages 1–7, 2022.
31. Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 34:7193–7206, 2021.
32. Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:16082–16097, 2022.
33. Davide Maran, Pierriccardo Olivieri, Francesco Emanuele Stradi, Giuseppe Urso, Nicola Gatti, and Marcello Restelli. Online markov decision processes configuration with continuous decision space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14315–14322, 2024.
34. Abdullah Al Hayajneh, Hasnain Nizam Thakur, and Kutub Thakur. The evolution of information security strategies: A comprehensive investigation of infosec risk assessment in the contemporary information era. *Computer and Information Science*, 16(4):1–20, 2023.
35. Claire Chen, Shuze Liu, and Shangtong Zhang. Efficient policy evaluation with safety constraint for reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, pages 1–22, 2025.

36. Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 488–489, 2017.
37. Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martin Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, pages 1–28.
38. Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, pages 1–17, 2019.
39. Zhiqun Wang, Zikai Jin, Zhen Yang, Wenchao Zhao, and Mahdi Mir. An intelligent fuzzy reinforcement learning-based routing algorithm with guaranteed latency and bandwidth in sdn: Application of video conferencing services. *Egyptian Informatics Journal*, 27:1–11, 2024.

Li Wei is an Engineer at Suzhou Suneng Group Co., LTD of China. He has rich experience on power grid technology and application. His research interests include power system technology, substation network, and flow control.

Wu Yong is an Engineer at Suzhou Suneng Group Co., LTD of China. He has rich experience on communication technology and application. His research interests include communication technology, substation network, electric power information network Quality of Service, and flow control.

Yan Dong a Senior Engineer at Suzhou Suneng Group Co., LTD of China. His research interests include intelligent operation and maintenance, information and communication, electric power information network Quality of Service, network security, and artificial intelligence.

Received: September 23, 2025; Accepted: December 14, 2025.

Computer Science and Information Systems 23(1):299–320 https://doi.org/10.2298/CSIS251015009H

Robust QoS-Aware Network Scheduling for Smart Substations via Multi-Agent Adversarial Reinforcement Learning

Ping He, Dongsheng Jing, Baozhen Qi, Yu Yang, and Jingsong Xue

State Grid Suzhou Power Supply Company
Suzhou Jiangsu 215004, China
{hep_sz, jds_sz, qibz, yang_yu, js.xue }@js.sgcc.com.cn

Abstract. With the rapid development of modern power systems, traditional scheduling and reinforcement learning methods often fail to meet stringent Quality of Service (QoS) demands for low latency, high reliability, and stable bandwidth under large-scale bursty traffic. To address this problem, we propose a QoS-driven routing optimization approach based on Adversarial Reinforcement Learning, referred to as Adversarial Critic-Cooperative Actor (ACCA). By introducing adversarial agents that model worst-case perturbations, ACCA establishes a multi-agent game framework that enhances policy robustness and adaptability in dynamic network environments. Furthermore, a multi-dimensional state representation and a QoS-aware cost function are designed to capture metrics such as delay, bandwidth utilization, queue length, and packet loss. Experiments demonstrate that ACCA outperforms traditional routing protocols and standard reinforcement learning algorithms in terms of end-to-end delay, load balancing, and throughput, thereby providing an effective solution for QoS assurance in intelligent power communication networks.

Keywords: Bandwidth Management, Deep Reinforcement Learning, Adversarial Learning, Quality of Service.

1. Introduction

As the core hub of the current power system, smart substations not only perform critical functions such as data acquisition, processing, and intelligent control but also achieve deep integration among primary equipment, protection devices, and communication networks through the deployment of intelligent electronic devices[1,2]. Smart substations can support predictive maintenance by leveraging artificial intelligence, thereby enhancing operational efficiency and grid resilience.

However, smart substations impose stringent communication requirements. Remote control is the most critical service, as it directly affects grid safety and stability. In contrast, auxiliary services, such as real-time monitoring, state awareness, video transmission, and online analysis, also demand low latency, high reliability, and stable bandwidth [3,4,5]. In practice, power communication networks face substantial and bursty traffic, especially during fault detection or control command dispatch, and wireless links at distribution and sensing layers further limit bandwidth and stability [6,7]. Effectively managing high-volume, bursty, and wireless-constrained traffic is therefore critical for safe and stable grid operation. Conventional scheduling strategies, such as Open Shortest Path

First (OSPF), rely on static link costs and lack global state awareness, making them unable to adapt to topology changes, resource contention, or sudden congestion [8,9]. Reinforcement Learning (RL) has recently been explored for network scheduling, showing promise in improving adaptability and resource utilization. For example, in Deterministic Networking (DetNet), A3C-MSO frameworks leverage Graph Convolutional Networks (GCNs) to learn multiple routing and scheduling policies with improved generalization [10,11,12]. However, most RL-based approaches still overlook disturbances such as link failures, traffic surges, and environmental noise [13], which limit the robustness of policies in industrial networks.

To address these limitations, we propose a novel QoS-driven routing optimization approach based on Adversarial Reinforcement Learning [14], referred to as Adversarial Critic-Cooperative Actor (ACCA). The ACCA approach is specifically designed for heterogeneous industrial networks under the intelligent control paradigm [15]. It integrates global network state perception, intelligent routing, and disturbance modeling into a unified training framework, aiming to enhance the policy's generalization ability and robustness under quality of service constraints through adversarial interactions. Specifically, ACCA consists of a target actor agent and two adversarial agents. These adversaries simulate realistic disturbances in task conditions (e.g., bandwidth fluctuations, latency jitter) and cost conditions (e.g., malicious link injections, packet drops). Through adversarial training, the target policy learns to cope with dynamic and hostile conditions, leading to enhanced robustness and adaptability. For state modeling, the intelligent control plane is abstracted as a graph structure. Protocols like the Link Layer Discovery Protocol (LLDP) and the OpenFlow control interface (OpenFlow) are employed to collect multidimensional metrics—such as delay, bandwidth utilization, queue length, and packet loss—to construct a high-dimensional state space suitable for deep reinforcement learning [16,17]. Disturbance modeling is achieved via adjustable perturbation coefficients, enabling controlled simulation of task-level and cost-level noise. During path optimization, a QoS-driven cost function and reward design are employed to ensure a balance between performance and safety.

The main contributions of this study are summarized as follows:

- We propose ACCA, a robust QoS optimization framework based on adversarial reinforcement learning, tailored explicitly for intelligent control–based industrial networks, which significantly enhances the reliability of traffic scheduling under complex dynamics.
- We design a multi-agent adversarial training architecture that models both task and cost disturbances to enhance policy robustness against uncertain environmental factors.
- We construct a multi-dimensional network state representation and a QoS-aware path cost function to support fine-grained decision-making and adaptive control.
- We evaluate the proposed method on delay, load balancing, and throughput, and perform ablation studies to assess the impact of perturbation mechanisms on learning outcomes.

In summary, the ACCA algorithm introduces adversarial disturbance modeling into RL-based scheduling frameworks, leveraging centralized control and programmability of intelligent control to achieve robust and adaptive QoS optimization. This method not

only enhances the intelligence and cost-effectiveness of industrial network scheduling but also provides a novel technical approach and theoretical foundation for optimizing communication systems in intelligent manufacturing.

The remainder of this paper is organized as follows: Section 2 reviews existing approaches for QoS routing optimization and robustness enhancement. Section 3 introduces the proposed framework, which includes reinforcement learning preliminaries, a formal problem definition, and an adversarial multi-agent architecture. Section 4 presents the QoS optimization architecture built upon Adversarial Reinforcement Learning and formulates the corresponding optimization model. Section 5 describes the experimental design and evaluation metrics. Section 6 reports and analyzes the results. Section 7 concludes the paper and outlines future research directions.

2. Related Work

To comprehensively understand the research progress in QoS routing optimization and robustness enhancement, this section provides a structured review from three perspectives: QoS Routing Based on SPF and Heuristic Enhancements, Reinforcement Learning for Adaptive QoS Routing, and Adversarial Learning for Robust Routing Policy Optimization.

2.1. QoS Routing Based on SPF and Heuristic Enhancements

Shortest Path First algorithms and their variants have been widely adopted in traditional IP networks. Representative protocols such as the Open Shortest Path First and Routing Information Protocol [18] typically rely on single network metrics (e.g. , delay or bandwidth) for path selection. While these approaches perform well under simple topologies or controllable traffic conditions, they often lead to traffic concentration on specific links in complex or high-load networks, causing congestion, unbalanced resource utilization, and increased transmission delay[19].

To overcome these limitations, researchers have proposed several enhancements. For instance, Smart OSPF adapts routing decisions by sensing global network states at the destination, thereby achieving improved traffic distribution. In addition, heuristic optimization algorithms—such as Ant Colony Optimization [20], Simulated Annealing [21], and Genetic Algorithms [22] —have been introduced to improve path quality, reduce packet loss, and balance network load [23]. However, these methods often suffer from high computational complexity and low execution efficiency, making them difficult to apply in latency-sensitive industrial networks with strict QoS requirements.

2.2. Reinforcement Learning for Adaptive QoS Routing

Reinforcement learning has emerged as a promising framework for sequential decision-making problems and has been widely applied in network routing and traffic management. By continuously interacting with the environment, RL agents autonomously learn policies that adapt to dynamic network conditions and QoS constraints.

Prior work includes the algorithm proposed by Casas-Velasco et al. [24], which incorporates link state information to mitigate congestion caused by traffic aggregation. Similarly, Sendra et al. employed RL to predict underlying network behaviors and improve

bandwidth scheduling under the intelligent control architectures [25]. Nevertheless, traditional Q-learning methods encounter scalability issues in large state-action spaces, as the Q-table quickly becomes prohibitively large, limiting practical deployments in large-scale networks.

To address these limitations, Deep Q-Networks [26,27] utilize deep neural networks to approximate Q-values, enabling efficient learning in large-scale topologies such as NSFNET and ARPANET [28]. However, DQN is prone to overestimating the value function, leading to sub-optimal policy decisions and reduced training stability.

2.3. Adversarial Learning for Robust Routing Policy Optimization

Adversarial Learning [29,30], initially inspired by Generative Adversarial Network (GAN) [31,32], has gained significant traction in improving model robustness and generalization. In classical GAN frameworks, a generator and a discriminator are trained in opposition: the generator aims to produce realistic samples, while the discriminator distinguishes real from fake data. This adversarial training paradigm enables models to resist external perturbations better and adapt to diverse inputs.

In recent years, adversarial mechanisms have been increasingly applied in reinforcement learning to enhance policy robustness in dynamic and potentially adversarial environments [14,33]. Specifically in routing optimization, adversarial training allows the simulation of environmental disturbances and policy deviations, encouraging agents to learn resilient and adaptive strategies [34,35]. For example, adversarially trained policies demonstrate more stable performance and enhanced QoS preservation in latency manipulation or deceptive path guidance scenarios. In network control and scheduling, Adversarial Reinforcement Learning (ARL) employs adversarial agents or discriminators to simulate disturbances and non-stationary dynamics, enabling the learning agent to develop more robust and adaptive strategies. For example, the Generative Adversarial Reinforcement Learning Scheduling (GARLSched) algorithm [36] integrates a task embedding–based discriminator to guide policy learning, thereby improving convergence stability and adaptability to unseen workloads.

3. Methodology

This section presents the proposed framework, beginning with an overview of reinforcement learning theory. We formalize the problem by defining the state space, action space, and reward function. Then, we detail the adversarial multi-agent architecture, elaborating on the learning mechanisms and interactions among the target policy agent, task-level attacker, and cost-level attacker. This framework aims to achieve robust scheduling optimization under multi-dimensional disturbances, providing theoretical support for subsequent experimental validation.

3.1. QoS-Aware Routing Optimization via Reinforcement Learning

Reinforcement learning is an interactive learning paradigm based on the Markov Decision Process (MDP)[37]. The MDP problem is typically defined as a tuple:

$$\xi = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle . \tag{1}$$

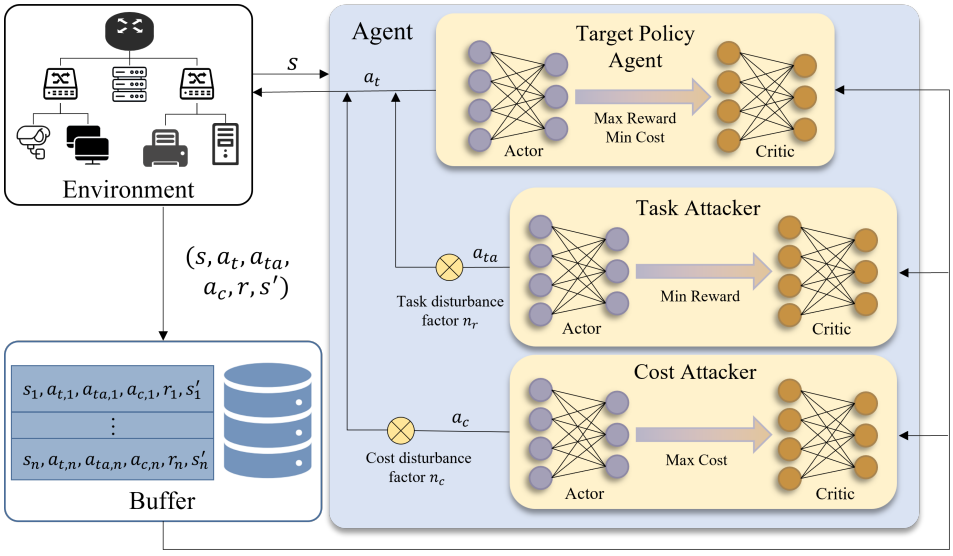


Fig. 1. Overview of the Adversarial Critic-Coordinated Actor (ACCA) architecture for secure deep reinforcement learning. The framework comprises a target policy agent and two adversarial agents (task attacker and cost attacker), all interacting with the intelligent control plane via an actor-critic learning scheme

where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to R$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor. At time t, the agent observes state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ based on policy π, transitions to s_{t+1} according to $\mathcal{P}$, and receives reward $r_t = \mathcal{R}(s_t, a_t)$. The objective is to learn an optimal policy π^* that maximizes the expected cumulative discounted reward:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] . \tag{2}$$

The above Markov decision process formulation provides a general mathematical foundation for bandwidth regulation for wireless power communication networks. To apply it effectively, we need to instantiate its core components — namely, the state space, action space, and reward function — in accordance with the characteristics of data transmission and remote control tasks in smart substations.

State: In the network, the state reflects multi-dimensional performance metrics and flow scheduling information. At time t, the global state is defined as:

$$s_t = \{x_e^t \mid e \in \mathcal{E}\}, \quad x_e^t = [d_e^t, U_e^t, p_e^t, q_e^t] . \tag{3}$$

where d_e^t denotes link delay, U_e^t the bandwidth utilization, p_e^t the packet loss rate, and q_e^t the queue length at link e. These are collected by the intelligent controller via OpenFlow statistics and LLDP probes, ensuring comprehensive network observability.

Action: The action defines a selection of routing paths for multiple concurrent flow requests. Given a request d_k with candidate path set $\mathcal{P}_k = \{p_k^1, p_k^2, \ldots, p_k^K\}$, the action at

time t is:

$$a_t = \{p_k \mid p_k \in \mathcal{P}_k,\ k = 1, 2, \ldots, M\}\,. \tag{4}$$

where M is the number of active flows. The agent learns optimal path assignments to balance QoS requirements and system robustness. For example, a video stream from host H1 to H8 may have two candidate paths p_1^1 = H1–S1–S3–S6–H8, offering low latency but high load, and p_1^2 = H1–S2–S5–S6–H8, offering higher stability but slightly longer delay. The agent learns to balance such trade-offs to satisfy QoS requirements while maintaining network robustness.

Reward: The reward encourages the agent to select paths that meet QoS constraints while being robust against disturbances. We define the reward as the negative total path cost under the perturbed network state $\tilde{s}_t$:

$$J(p_k, \tilde{s}_t) = \sum_{e \in p_k} \left(\lambda_1 d_e^t + \lambda_2 (1 - U_e^t) + \lambda_3 p_e^t + \lambda_4 q_e^t\right)\,. \tag{5}$$

$$r_t = -\sum_{k=1}^{M} J(p_k, \tilde{s}_t) = -\sum_{k=1}^{M} \sum_{e \in p_k} \left(\lambda_1 d_e^t + \lambda_2 (1 - U_e^t) + \lambda_3 p_e^t + \lambda_4 q_e^t\right)\,. \tag{6}$$

In the above formulation, $J(p_k, \tilde{s}_t)$ represents the weighted cost of flow k along the selected path p_k, incorporating s_t. The parameters λ_1–λ_4 control the relative importance of these metrics in the overall cost, balancing multiple performance objectives such as latency, bandwidth efficiency, and reliability. By taking the negative of the total path cost, the reward function encourages the agent to select routing paths with lower delay, reduced packet loss, and higher bandwidth utilization.

The actual network state under the influence of perturbations is defined as:

$$\tilde{s}_t = s_t + \eta_r \cdot \delta_t^T + \eta_c \cdot \delta_t^S\,. \tag{7}$$

Here, δ_t^T and δ_t^S represent task and cost perturbations, with η_r and η_c as their respective intensity coefficients. Weights λ_i reflect the relative importance of each QoS metric.

3.2. Adversarial Reinforcement Learning Architecture

To enhance the robustness of network scheduling strategies, we develop a multi-agent adversarial reinforcement learning framework. The overall framework mainly consists of three components: the environment, the agents, and the replay buffer. The agent module comprises a target policy agent and two adversarial agents — a task-level attacker and a cost-level attacker, as illustrated in Fig. 1. All agents are constructed using actor-critic architectures and are jointly deployed within the control layer to enable coordinated adversarial training:

- **Task Attacker:** Mimics performance degradation by injecting perturbations (e.g., traffic bursts), aiming to minimize the long-term reward.
- **Cost Attacker:** Simulates cost threats such as packet drops or congestion injection, aiming to maximize a long-term cost.
- **Target Policy Agent:** Learns robust routing decisions to maximize the cumulative reward under adversarial conditions.

The adversarial training process among the three agents drives the target policy to progressively adapt to dynamic perturbations, thereby ensuring resilient and stable traffic scheduling in the presence of adversarial disruptions.

Task Attacker: The task attacker is modeled using an actor-critic framework, where the actor approximates the task perturbation policy π_{task} to generate effective disturbances that degrade task performance. The critic network estimates the long-term discounted task reward function Q_r as:

$$Q_r(s_t, a_{ta}) = E_b \left[\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{ta'}) \right] . \tag{8}$$

where E_b denotes the empirical expectation over trajectories sampled from the experience buffer.

The critic is trained by minimizing the mean squared error (MSE) loss:

$$\mathcal{L}_{Q_r} = \text{MSE}\left(Q_r(s_t, a_{ta}), r_t + \gamma Q'_r(s_{t+1}, \pi'_{\text{task}}(s_{t+1}))\right) . \tag{9}$$

where Q'_r and π'_{task} respectively represent the target critic and target actor networks.

The actor is updated by minimizing the negative value estimated by the critic, with the loss function defined as:

$$\mathcal{L}_{\pi_{\text{task}}} = E_b \left[Q_r(s_t, \pi_{\text{task}}(s_t)) \right] . \tag{10}$$

The optimal task perturbation policy is obtained by:

$$\pi^*_{\text{task}} = \arg\min_{\pi_{\text{task}}} \mathcal{L}_{\pi_{\text{task}}} . \tag{11}$$

Cost Attacker: The cost attacker also adopts an actor-critic framework, where the actor learns a perturbation policy π_{cost} aimed at maximizing long-term cost through adversarial perturbations.

The critic network estimates the discounted cumulative cost Q_c as:

$$Q_c(s_t, a_c) = E_b \left[\sum_{t'=t}^{T} \gamma^{t'-t} \, c(s_{t'}, a_{c'}) \right] . \tag{12}$$

where $c(s_t, a_c)$ denotes the penalty for violating constraints at time t, and E_b is the empirical expectation over sampled trajectories.

The critic is optimized by minimizing the mean squared error (MSE) between predicted and target costs:

$$\mathcal{L}_{Q_c} = \text{MSE}\big(Q_c(s_t, a_c), \quad c_t + \gamma \, Q'_c(s_{t+1}, \pi'_{\text{cost}}(s_{t+1}))\big) . \tag{13}$$

where Q'_c and π'_{cost} are the target critic and actor networks, respectively. The actor updates its policy to maximize the expected long-term cost estimated by the critic:

$$\mathcal{L}_{\pi_{\text{cost}}} = E_b \left[Q_c(s_t, \pi_{\text{cost}}(s_t)) \right] . \tag{14}$$

The optimal cost attack policy is defined as:

$$\pi^*_{\text{cost}} = \arg\max_{\pi_{\text{cost}}} \mathcal{L}_{\pi_{\text{cost}}} . \tag{15}$$

Target Policy Agent: The target policy agent also follows an actor-critic framework, where the actor learns a routing policy π_{target} to generate path selection actions, and the critic estimates the long-term discounted difference between cumulative reward and safety cost, denoted as Q_{rc}:

$$Q_{rc}(s_t, a_t) = E_b \left[\sum_{t'=t}^{T} \gamma^{t'-t} \left(r(s_{t'}, a_{t'}) - c(s_{t'}, a_{t'}) \right) \right] . \tag{16}$$

Here, $r(s_{t'}, a_{t'})$ is the immediate reward, $c(s_{t'}, a_{t'})$ is the safety cost, γ is the discount factor, and $\mathbb{E}_b[\cdot]$ denotes the expectation over the policy distribution. This evaluation function provides long-term optimization signals to the actor, enabling the agent to select robust paths that satisfy both QoS requirements and network safety in dynamic environments.

The critic network is trained by minimizing the mean squared error between predicted and target values:

$$\mathcal{L}_{Q_{rc}} = \text{MSE}\left(Q_{rc}(s_t, a_t), r_t - c_t + \gamma Q'_{rc}(s_{t+1}, \pi'_{\text{target}}(s_{t+1})) \right) . \tag{17}$$

The actor is optimized to maximize the expected utility output from the critic, using the following loss:

$$\mathcal{L}_{\pi_{\text{target}}} = -E_b \left[Q_{rc}(s_t, \pi_{\text{target}}(s_t)) \right] . \tag{18}$$

The optimal target policy is thus obtained by:

$$\pi^*_{\text{target}} = \arg\max_{\pi_{\text{target}}} E_b \left[Q_{rc}(s_t, \pi_{\text{target}}(s_t)) \right] . \tag{19}$$

Final Action Composition: In actual agent-environment interaction, the action executed by the system is the combination of the target policy and perturbations introduced by the task and cost attackers:

$$a_t = \pi_{\text{target}}(s_t) + \eta_r \cdot \pi_{\text{task}}(s_t) + \eta_c \cdot \pi_{\text{cost}}(s_t) . \tag{20}$$

Here, η_r and η_c are the scaling coefficients for the task and cost perturbations, respectively. The influence of these perturbations on system performance will be further investigated through ablation studies in subsequent sections.

Algorithm 1 Adversarial Reinforcement Learning-based QoS Optimization (ACCA)

Input: Network topology $G(V, E)$, initial QoS parameters θ, convergence threshold ϵ.
Output: Optimized routing policy π^*
Initialize actor network π_θ and critic network Q_ω
Initialize attacker agents $\mathcal{A}_t$ (task attacker), $\mathcal{A}_s$ (cost attacker)
Initialize experience replay buffers $\mathcal{D}_\pi$, $\mathcal{D}_t$, $\mathcal{D}_s$
for each training episode **do**
    Initialize episode cumulative reward R and cost C
    Initialize trajectory buffer b
    ****
    Data Collection Stage
    ****
    for each time step t **do**
        Observe current network state s_t
        Sample actions $a_t = \pi_\theta(s_t)$, $a_{ta} = \mathcal{A}_t(s_t)$, $a_c = \mathcal{A}_c(s_t)$
        Combined action: $a_t^{total} = a_t + \eta_r a_{ta} + \eta_c a_c$
        Execute a_t^{total} and observe next state s_{t+1}, reward r_t, and cost c_t
        Update episode totals: $R = R + r_t$, $C = C + c_t$
        Store $(s_t, a_t, a_{ta}, a_c, s_{t+1}, r_t, c_t)$ in trajectory buffer b
    end for
    Add trajectory b to experience buffers $\mathcal{D}_t$, $\mathcal{D}_{ta}$, $\mathcal{D}_c$
    ****
    Actor-Critic Training Stage
    ****
    Sample mini-batches from $\mathcal{D}_t$, $\mathcal{D}_{ta}$, $\mathcal{D}_c$
    ******** Task-level Attacker Update ********
    Update critic Q_r with loss L_{Q_r} via Eq.8
    Update task attacker $\mathcal{A}_t$ with loss $L_{\pi_{task}}$ via Eq.10
    ******** Cost-level Attacker Update ********
    Update critic Q_c with loss L_{Q_c} via Eq.12
    Update cost attacker $\mathcal{A}_c$ with loss $L_{\pi_{cost}}$ via Eq.14
    ******** Target Policy Update ********
    Update combined critic Q_{rc} with loss $L_{Q_{rc}}$ via Eq.17
    Update actor π_θ with loss $L_{\pi_{target}}$ via Eq.18
    if convergence criterion $|\Delta J(\pi_\theta)| < \epsilon$ is satisfied **then**
        break
    end if
end for
Return Optimized policy $\pi^* = \pi_\theta$

The process of learn robust scheduling policies under multi-dimensional network disturbances is shown in Algorithm 1. The algorithm consists of two main stages: Data Collection and Actor-Critic Training. During the data collection stage, the target policy agent generates actions based on the current network state, while the task-level and cost-level attacker agents introduce perturbations. The combined actions are applied to the network environment, and the resulting state transitions, rewards, and costs are recorded in trajectory buffers. These trajectories are stored in experience replay buffers for subsequent training.

In the actor-critic training stage, the task-level and cost-level attacker agents are updated to maximize adversarial objectives, enhancing the network's robustness against disturbances. Simultaneously, the target policy agent is updated using the combined critic to optimize reward and cost objectives. This iterative training process continues until the policy converges, yielding an optimized routing policy that achieves robust QoS performance under dynamic and uncertain network conditions.

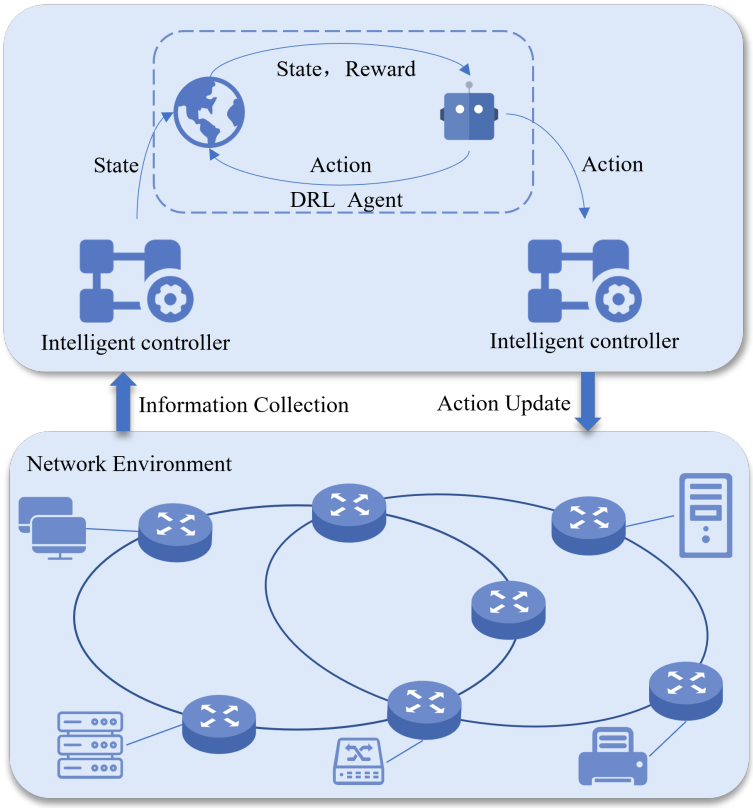


Fig. 2. The QoS optimization architecture for heterogeneous factory networks under intelligent control

4. QoS Optimization with ACCA

This section proposes a QoS optimization architecture based on Adversarial Reinforcement Learning, aiming to address the challenges of maintaining robust bandwidth control in wireless power communication networks. First, we present the architectural design and define the key network components. Then, a formal QoS optimization model is developed to provide the theoretical foundation for intelligent and resilient bandwidth scheduling under dynamic channel conditions and network disturbances.

4.1. ARL-Driven QoS Optimization Architecture

With the rapid evolution of wireless power communication networks, ensuring stable and efficient data transmission has become a fundamental requirement for modern smart grids and power systems. Diverse traffic types coexist in such networks, including real-time monitoring signals, control commands, and high-volume measurement data. These services impose heterogeneous Quality of Service requirements, where latency, bandwidth utilization, and reliability are especially critical since they directly affect grid stability and system responsiveness. However, traditional QoS-aware routing and bandwidth allocation strategies often fail to cope with the dynamic wireless environment, interference, and potential adversarial disruptions. This motivates the design of intelligent and robust optimization architectures, where adversarial reinforcement learning can provide adaptive decision-making for QoS-driven bandwidth regulation in WPCNs.

Conventional static scheduling strategies and basic reinforcement learning methods often fail to maintain robust performance in highly dynamic network states and the presence of multiple sources of disturbances (e.g., traffic bursts, link anomalies, malicious attacks). To address this, we design a multi-agent QoS optimization architecture based on ARL, as illustrated in Fig. 2, which integrates the benefits of heterogeneous factory networks, intelligent centralized control, and adversarial learning.

The architecture consists of three layers:

- **Data Plane:** Composed of OpenFlow-enabled virtual switches, responsible for efficient packet forwarding and traffic processing.
- **Control Plane:** Consists of intelligent controllers that periodically transmit Link Layer Discovery Protocol packets and use echo mechanisms to monitor network topology and link states dynamically. Port statistics are collected to extract key metrics such as bandwidth utilization, packet loss rate, and queue length.
- **Decision Plane:** Hosts ARL-based agents that leverage global network state information from the control plane to learn robust path selection and traffic scheduling strategies. This layer includes one target policy network and two adversarial networks that simulate task-level and cost-related disturbances. The adversarial outputs are weighted and injected into the network state to guide robust policy learning under complex and dynamic environments. Finally, flow table rules are generated by the intelligent controller based on the agent's outputs, enabling unified encapsulation and dynamic configuration across the data link (e.g., MAC address, VLAN ID), network (e.g., IP address), and transport layers (e.g., TCP/UDP ports), forming a flat and intelligent factory network.

4.2. QoS Optimization Modeling

In wireless power communication networks, the dynamic nature of spectrum usage and energy constraints leads to constantly evolving network states, while the QoS demands of different data flows vary significantly. Particularly in mission-critical scenarios such as industrial monitoring and autonomous unmanned systems, the network must achieve efficient and robust bandwidth allocation under constraints like interference, delay, and limited wireless resources.

Algorithm 2 QoS Optimization with ACCA

Require: Network graph $G(V, E)$, flow request $d_k = (s_k, t_k)$
Require: Disturbance coefficients η_r, η_c
Require: Learning rates $\alpha_p, \alpha_T, \alpha_C$
Ensure: Robust routing policy π_θ

Initialize policy network π_θ, task attacker A_T, cost attacker A_C
Initialize replay buffer $\mathcal{D}$
for each training episode **do**
    Observe network state s_t
    Adversarial disturbance generation
    $\delta_t^T \leftarrow A_T(s_t)$ via Eq.8
    $\delta_t^S \leftarrow A_C(s_t)$ via Eq.12
    $\tilde{s}_t \leftarrow s_t + \eta_r \delta_t^T + \eta_c \delta_t^S$ via Eq.23
    QoS cost evaluation
    for each path $p_k \in \mathcal{P}_k$ **do**
        Compute link metrics $\{d_e, U_e, p_e, q_e\}$
        Evaluate path cost $J(p_k, \tilde{s}_t)$
    end for
    Policy decision
    Sample/select path $p_k \sim \pi_\theta(\tilde{s}_t)$
    Apply forwarding rule to data plane switches
    Environment feedback
    Observe reward $r_t \leftarrow -J(p_k, \tilde{s}_t)$ and next State s_{t+1} via Eq. 5
    Store transition $(s_t, \tilde{s}_t, p_k, r_t, s_{t+1})$ into $\mathcal{D}$
    Joint update
    Sample minibatch $B \subset \mathcal{D}$
    $\theta_T \leftarrow \theta_T + \alpha_T \nabla_{\theta_T} L_T(B)$
    $\theta_C \leftarrow \theta_C + \alpha_C \nabla_{\theta_C} L_C(B)$
    $\theta \leftarrow \theta - \alpha_p \nabla_\theta L_\pi(B)$
end for
Return π_θ

We model the wireless power communication network as an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_M\}$ denotes the set of nodes representing wireless power communication devices (e.g., access points, relays, or terminals), and $E = \{e_1, e_2, \ldots, e_N\}$ denotes the wireless links subject to fading and interference. The network controller leverages spectrum sensing and power feedback to monitor the topology and link-level states, obtaining key metrics such as signal-to-interference-plus-noise ratio, residual energy, throughput, and queue lengths.

Each link $e \in E$ at time t is described by a four-dimensional vector:

$$\mathbf{x}_e^t = [d_e^t, U_e^t, p_e^t, q_e^t]\,. \tag{21}$$

where d_e^t is the LLDP-measured delay, p_e^t is the packet loss rate, and q_e^t is the queue length as a measure of congestion. U_e^t is the bandwidth utilization calculated as:

$$U_e^{used} = \frac{\Delta\mathrm{Rx} + \Delta\mathrm{Tx}}{\Delta t}, \quad U_e^t = \frac{U_e^{used}}{B_e}\,. \tag{22}$$

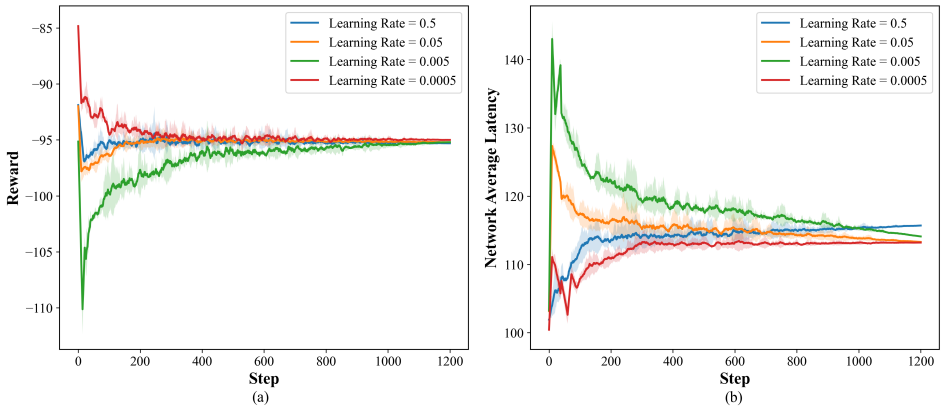


Fig. 3. Performance of the proposed algorithm under the same disturbance coefficients η_r and η_c with different learning rates α. (a) and (b) represent the reward and the average network delay, respectively

Here, B_e is the maximum bandwidth of the link, and ΔRx, ΔTx denote the bytes received and transmitted during interval Δt. The overall network state at time t is via Eq. 3.

For a given flow request d_k, with source s_k and destination t_k, a set of candidate paths $\mathcal{P}_k = \{p_1, p_2, \ldots, p_K\}$ is considered. A cost function quantifies the QoS of a path via Eq. 5.

4.3. Adversarial Disturbance Modeling

In wireless power communication networks, link states are highly susceptible to various disturbances, primarily stemming from spectrum dynamics and security threats. These disturbances can be categorized into communication-level fluctuations and adversarial cost attacks. The former, denoted by the vector δ_t^T, captures uncertainties such as channel fading, spectrum contention, and traffic bursts. The latter, denoted by δ_t^S, represents malicious behaviors including jamming, interference injection, and bandwidth occupation attacks. Under such disturbances, the actual network state can be expressed as:

$$\tilde{s}_t = s_t + \eta_r \cdot \delta_t^T + \eta_c \cdot \delta_t^S \,. \tag{23}$$

where η_r and η_c are disturbance coefficients controlling the intensity of task and cost perturbations. These perturbations may increase delay, reduce bandwidth availability, increase loss rates, and lengthen queues, thus impacting routing decisions and QoS performance.

Optimization Objective: Given disturbed state modeling, the objective is to learn a robust routing policy π that minimizes the expected path cost under adversarial conditions.

$$\min_{\pi} \; E_{\delta_t} \left[J(p_k, \tilde{s}_t) \right] \,. \tag{24}$$

subject to:

- Valid probability distribution over path selection: $\sum_{k=1}^{K} w_k = 1, \quad w_k \geq 0$
- QoS constraints: $J(p_k, \tilde{s}_t) < L_{\max}$
- Link capacity constraint: $U_e^t < 1, \forall e \in p_k$
- Robustness requirement: $|J(p_k, \tilde{s}_t) - J(p_k, s_t)| \leq \epsilon$

The algorithm 2 outlines the core inference procedure of ACCA, detailing the complete operational sequence from state acquisition to adversarial interaction and final path selection. This optimization framework provides the theoretical foundation for learning robust ARL-based bandwidth regulation policies, enhancing both the reliability and security of resource allocation in wireless power communication networks.

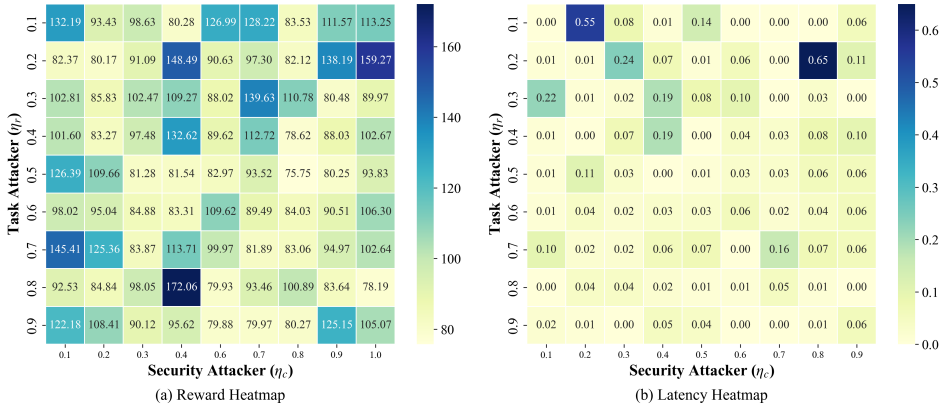


Fig. 4. Sensitivity analysis of the proposed algorithm under varying task perturbation coefficient η_r and cost perturbation coefficient η_c: (a) average episodic reward heatmap; (b) average episodic cost cost heatmap

5. Experiments

This section presents a comprehensive set of experiments designed to evaluate the effectiveness of the proposed ACCA approach. The experiments aim to analyze three aspects: the convergence behavior of the algorithm, its sensitivity to perturbation parameters, and its performance compared with baseline methods. All evaluations are conducted in a unified simulation environment using standard QoS-related metrics, including reward value, end-to-end latency, load balancing, and throughput.

5.1. Experimental Environment

To evaluate the effectiveness of the proposed method in the context of bandwidth regulation for wireless power communication networks, we developed a simulation testbed using the Mininet network emulator in combination with the Ryu intelligent controller. The simulated environment represents a smart substation scenario, where heterogeneous

devices, including intelligent electronic devices, sensors, actuators, and control terminals, are interconnected. The network adopts a hybrid wired/wireless topology supporting multiple communication protocols, including IEC 61850, Modbus, Wi-Fi, and TCP/IP.

Small-scale Mininet topology: The network topology consists of 6 switches and 8 hosts, forming a hybrid wired/wireless architecture that emulates the heterogeneous communication environment of a power communication network. Among the hosts, different roles are assigned to reflect real-world scenarios, including phasor measurement units (PMUs), protection relays, intelligent electronic devices (IEDs), and remote monitoring terminals. Flow request rates range from 10 Mb/s to 120 Mb/s, supporting both low-latency monitoring signals and high-throughput control or protection data streams.

Large-scale Mininet topology: To further assess scalability and robustness, additional experiments were conducted on an extended topology with up to 20 switches and 32 hosts, introducing more complex interconnections and traffic dynamics.

This multi-scale experimental configuration enables the simulation of diverse traffic conditions under different operational states—such as routine monitoring, demand response, fault detection, and emergency control—providing a realistic and comprehensive basis for assessing bandwidth regulation and QoS optimization strategies in wireless power communication networks.

Table 1. Routing Performance Comparison on Small-scale and Large-scale Mininet Topologies at 120 Flows/s

Method	Load Imbalance(%)	Throughput (Mbps)	network latency(ms)
Small-scale Mininet topology			
OSPF	0.75	60	116.25
DQN	0.55	63	115.25
A3C-MSO	0.50	68	114.64
ACCA	0.31	72	113.14
Large-scale Mininet topology			
OSPF	0.83	68	131.32
DQN	0.62	73	128.79
A3C-MSO	0.57	78	124.98
ACCA	0.40	84	122.18

5.2. Experimental Setup

The proposed ARL algorithm is implemented in PyTorch. The key hyperparameters are set as follows: learning rate $\alpha = 0.001$, discount factor $\gamma = 0.95$, exploration rate $\epsilon = 0.4$, and replay buffer size of 2000. To evaluate the performance and convergence of the model, the average reward is recorded every 100 training episodes as a measure of policy improvement. Each experiment is performed with a single random seed over 1200 training steps, allowing the algorithm to interact with the simulated network environment and progressively optimize the scheduling policy.

To comprehensively evaluate the approach, three primary performance metrics are used:

- **End-to-end latency** – the average per-flow transmission delay, reflecting the efficiency of data delivery.
- **Load balancing** – measured by the standard deviation of link utilization, indicating the distribution of traffic across the network.
- **Network throughput** – the total amount of successfully transmitted data over the network, representing bandwidth utilization.

During training, the ARL agent interacts with the network environment, selects actions based on the learned policy, and receives feedback in the form of rewards. This process enables the agent to adaptively adjust routing decisions, reducing latency, balancing load, and maximizing throughput.

5.3. Baseline Methods

To evaluate the performance of the proposed ACCA algorithm, we compare it against the following baseline approaches:

- **OSPF** – a widely used static routing protocol that determines paths based on link-state information. OSPF does not adapt dynamically to varying traffic loads, serving as a baseline for conventional routing performance.
- **DQN** – a reinforcement learning–based routing method that leverages deep Q-networks to learn optimal forwarding policies. By approximating the Q-function with neural networks, DQN can handle high-dimensional state spaces more effectively than traditional Q-learning. However, it often suffers from unstable convergence and limited generalization in large-scale or highly dynamic network scenarios, serving as a representative baseline of value-based RL approaches.
- **A3C-MSO** – an advanced RL-based QoS optimization approach designed for deterministic networking. It employs an asynchronous advantage actor–critic architecture with multipolicy learning, enabling joint routing and scheduling optimization. A3C-MSO improves modeling and generalization in complex network environments by incorporating graph convolutional networks for topology representation.

These baselines enable us to systematically evaluate the algorithm's advantages in terms of load balancing, latency reduction, and throughput maximization under dynamic network conditions.

6. Results and Analysis

6.1. Convergence and Robustness Evaluation

To evaluate the training stability and convergence capability of the proposed algorithm under perturbation environments, we first conducted a systematic assessment of the impact of different learning rates on training performance under fixed disturbance coefficients ($\eta_r = 0.5$, $\eta_c = 0.5$). Experiments were performed with a single random seed over 1200 training steps, recording the trends of average episodic reward and link latency to measure the convergence speed and stability of the model. As shown in Fig. 3, the algorithm achieves optimal performance at a learning rate of $\alpha = 0.0005$, exhibiting faster convergence, lower final network latency, and more balanced load distribution. During the initial

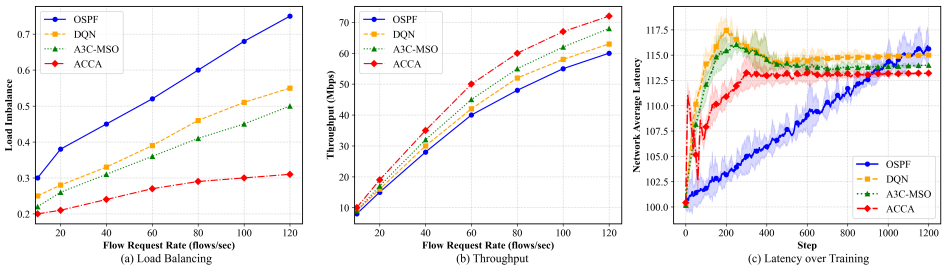


Fig. 5. Comparative performance evaluation of routing algorithms under varying traffic demand and training progress: (a) Network load imbalance versus traffic demand; (b) Network throughput versus traffic demand; (c) Average network latency during training versus training steps

training stage (approximately steps 0–50), transient congestion caused by traffic surges resulted in increased latency and decreased rewards; subsequently, the model progressively captured the network dynamics and optimized its policy. The reward curve stabilized between steps 400 and 600, with a significant reduction in latency, indicating that the policy had essentially converged and maintained stable performance thereafter.

After determining the optimal learning rate, we conducted a perturbation sensitivity study to systematically evaluate the robustness of the proposed algorithm under various combinations of the task perturbation coefficient η_r and the cost perturbation coefficient η_c, each ranging from 0.1 to 0.9. For each parameter pair, experiments were repeated three times with different random seeds, training for 500 episodes per run. Both the average episodic reward and average cost were recorded and visualized using heatmaps.

As shown in Fig. 4, the reward heatmap indicates that high-reward regions primarily cluster near the anti-diagonal of the perturbation coefficient grid, particularly when $\eta_r \leq 0.3$ or $\eta_r \geq 0.7$. This suggests that the policy performs stably and achieves superior performance under relatively weak or strong task perturbations. In contrast, moderate perturbations (e.g., $\eta_r \approx 0.5$) tend to disrupt learning, resulting in degraded policy performance. This phenomenon provides a possible explanation for the observed results: under weak perturbations, the environment is close to its nominal state, allowing the agent to learn the optimal policy with stable convergence. Under moderate perturbations, the agent receives conflicting signals—perturbations are strong enough to interfere with learning but insufficient to induce robust policy adaptation, resulting in suboptimal performance. Under strong perturbations, the agent is compelled to account for a broader range of potential environment variations during training, thereby enhancing the robustness of the learned policy and yielding higher rewards.

The cost heatmap further shows that the algorithm maintains low or near-zero average cost across most perturbation combinations, indicating robust cost defense. Elevated costs mainly occur at the boundaries $\eta_c \leq 0.4$ and $\eta_c \geq 0.8$, where the former corresponds to insufficient perturbation for effective threat recognition, and the latter reflects excessive perturbation that destabilizes training. Overall, these results highlight the nonlinear relationship between perturbation intensity and policy performance, suggesting that both very

weak and extreme perturbations can be beneficial for learning stable and robust policies. In contrast, moderate perturbations may negatively affect performance.

In summary, comprehensive analysis identifies the optimal disturbance parameter configuration as task perturbation coefficient $\eta_r = 0.8$ and cost perturbation coefficient $\eta_c = 0.4$. Under this configuration, the algorithm strikes a balance between high reward acquisition and low cost, demonstrating robust and effective performance in uncertain environments.

6.2. Comparative Performance Evaluation

After tuning key hyperparameters, we benchmark the proposed method against several baseline approaches under both small-scale and large-scale network settings. As summarized in Table 1, in the large-scale topology (20 switches and 32 hosts), all methods exhibit consistent performance trends with the small-scale case (6 switches and 8 hosts). Specifically, at a flow rate of 120 flows/s, **ACCA** achieves the lowest load imbalance at 0.40%, while A3C-MSO shows a higher imbalance of 0.57%. Similarly, **ACCA** attains the highest throughput of 84 Mbps, exceeding the 78 Mbps achieved by A3C-MSO, and maintains the lowest network latency at 122.18 ms compared to 124.98 ms for A3C-MSO. Compared to the small-scale topology, absolute values such as load imbalance and latency are slightly higher, and throughput is somewhat lower due to increased network depth and path diversity. Despite this, the relative performance advantages of **ACCA** remain consistent across topologies. Therefore, for clarity, only the small-scale Mininet results are illustrated in Fig. 5.

As shown in Fig. 5(a), the degree of load imbalance increases with traffic demand across all methods. However, the growth rate differs significantly. OSPF exhibits a sharp rise in imbalance under high loads, indicating its inability to balance traffic adaptively due to its static nature. The DQN-based method alleviates imbalance more effectively than OSPF, as it leverages deep Q-learning to explore adaptive routing decisions. However, its value-based learning framework often leads to unstable convergence, and the achieved balance is still inferior to that of actor–critic approaches. The A3C-MSO method achieves a better balance than both OSPF and DQN across all flow rates, demonstrating stronger adaptive capabilities. In contrast, the algorithm consistently maintains the lowest load imbalance, even under extreme conditions (e.g., 120 flows/sec), validating its superior congestion awareness and dynamic routing capability.

Fig. 5(b) further compares the throughput performance of the four methods under increasing traffic. OSPF shows saturation in throughput at medium to high load levels, while the DQN method achieves improved bandwidth utilization but suffers from unstable performance under bursty traffic. The A3C-MSO approach outperforms both OSPF and DQN, but the proposed ACCA method delivers the highest throughput across all flow levels, with the most pronounced improvements observed in the 60–120 flows/sec range. This confirms its effectiveness in handling complex network dynamics and adversarial perturbations, maximizing resource usage, and enhancing overall QoS.

Fig. 5(c) illustrates the evolution of the average network latency during training under the four routing strategies. As a static protocol, OSPF exhibits a relatively smooth but steadily increasing delay trend throughout the training process, indicating its limited adaptability. The DQN method exhibits significant fluctuations during training, reflecting its unstable convergence behavior. Although it can reduce latency compared to OSPF, it

fails to achieve consistent long-term improvements. In contrast, the A3C-MSO method shows a rapid increase in latency during the initial training stage, followed by partial stabilization in later stages; however, it still maintains a relatively high average delay. Notably, the algorithm demonstrates a brief fluctuation in the early stage, but quickly converges after approximately 400 steps and consistently maintains the lowest average latency thereafter. Throughout the entire training process, ACCA outperforms OSPF, DQN, and A3C-MSO, highlighting its superior capability in adapting to network dynamics and learning efficient routing policies, thereby effectively alleviating network congestion and enhancing data transmission efficiency.

7. Conclusion

We addressed the problem of bandwidth regulation in wireless power communication networks by proposing an adaptive routing optimization method based on Adversarial Reinforcement Learning, achieving joint optimization of Quality of Service and network cost. By incorporating the task perturbation factor η_r and the cost perturbation factor η_c, the method simulates bandwidth fluctuations and potential attacks in dynamic and uncertain network environments, thereby enhancing the agent's capability to perceive network state changes and adapt its policies accordingly. A robust training mechanism is also designed to improve the stability of policy convergence and its generalization performance.

The proposed method demonstrates significant potential in optimizing key metrics such as bandwidth allocation, link load balancing, and latency, indicating that the adversarial mechanism can effectively improve network resource utilization and enhance policy robustness. Nonetheless, the current perturbation model does not fully capture complex attacks, such as time-varying disturbances and intelligence-induced perturbations. The single-agent training framework also faces scalability limitations in multi-controller and multi-path scenarios, which are typical of wireless power communication networks.

Future work will consider integrating Graph Neural Networks or Multi-Agent Reinforcement Learning frameworks to enhance the model's representational capacity and decision-making efficiency in large-scale heterogeneous networks. Moreover, the exploration of explainability mechanisms will be pursued to improve the transparency and auditability of routing policies, providing more reliable support for practical deployment in wireless power communication networks.

Acknowledgments. This work was supported by Science and Technology Project of State Grid Jiangsu Electric Power Co., Ltd., and Science and Technology Project of Suzhou Suneng Group Co., Ltd. under Grant SGSZSNJTKJJS2500979.

References

1. Juan C Lozano, Keerthi Koneru, Neil Ortiz, and Alvaro A Cardenas. Digital substations and iec 61850: A primer. *IEEE Communications Magazine*, 61(6):28–34, 2023.
2. José Gaspar, Tiago Cruz, Chan-Tong Lam, and Paulo Simões. Smart substation communications and cybersecurity: A comprehensive survey. *IEEE communications surveys & tutorials*, 25(4):2456–2493, 2023.

3. Tze-Yang Tung and Deniz Gündüz. Deepwive: Deep-learning-aided wireless video transmission. *IEEE Journal on Selected Areas in Communications*, 40(9):2570–2583, 2022.
4. S.C. Jayasinghe, M. Mahmoodian, A. Sidiq, T.M. Nanayakkara, A. Alavi, Sam Mazaheri, F. Shahrivar, Z. Sun, and S. Setunge. Innovative digital twin with artificial neural networks for real-time monitoring of structural response: A port structure case study. *Ocean Engineering*, 312:119187, 2024.
5. Wenqing Zheng, Hao Frank Yang, Jiarui Cai, Peihao Wang, Xuan Jiang, Simon Shaolei Du, Yinhai Wang, and Zhangyang Wang. Integrating the traffic science with representation learning for city-wide network congestion prediction. *Information Fusion*, 99:101837, 2023.
6. Xiang Cheng, Dongliang Duan, Shijian Gao, and Liuqing Yang. Integrated sensing and communications (isac) for vehicular communication networks (vcn). *IEEE Internet of Things Journal*, 9(23):23441–23451, 2022.
7. Lipeng Zhu, Wenyan Ma, and Rui Zhang. Movable antennas for wireless communication: Opportunities and challenges. *IEEE Communications Magazine*, 62(6):114–120, 2023.
8. G. BalaMurugan, N. Palanivel, and A. John. Application reliable traffic control method for efficient data management in wireless-aided computer applications. In *2023 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5, 2023.
9. Quan Chen, Lei Yang, Yong Zhao, Yi Wang, Haibo Zhou, and Xiaoqian Chen. Shortest path in leo satellite constellation networks: An explicit analytic approach. *IEEE Journal on Selected Areas in Communications*, 42(5):1175–1187, 2024.
10. Wenbin Tian, Chaojie Gu, Miao Guo, Shibo He, Jiawen Kang, Dusit Niyato, and Jiming Chen. Large-scale deterministic networks: Architecture, enabling technologies, case study, and future directions. *IEEE Network*, 38(4):284–291, 2024.
11. Sijin Yang, Lei Zhuang, Jianhui Zhang, Julong Lan, and Bingkui Li. A multipolicy deep reinforcement learning approach for multiobjective joint routing and scheduling in deterministic networks. *IEEE Internet of Things Journal*, 11(10):17402–17418, 2024.
12. Ying Ma, Haijie Lou, Ming Yan, Fanghui Sun, and Guoqi Li. Spatio-temporal fusion graph convolutional network for traffic flow forecasting. *Information Fusion*, 104:102196, 2024.
13. Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *ACM Comput. Surv.*, 58(2):53:1–53:37, September 2025.
14. Maxwell Standen, Junae Kim, and Claudia Szabo. Adversarial machine learning attacks and defences in multi-agent reinforcement learning. *ACM Computing Surveys*, 57(5):1–35, 2025.
15. Rana M. Sohaib, Oluwakayode Onireti, Yusuf Sambo, Rafiq Swash, Shuja Ansari, and Muhammad A. Imran. Intelligent resource management for embb and urllc in 5g and beyond wireless networks. *IEEE Access*, 11:65205–65221, 2023.
16. Chen Wu, Shikai Guo, Hui Li, Chenchen Li, and Rong Chen. Estimating uncertainty in line-level defect prediction via perceptual borderline oversampling. *ACM Trans. Softw. Eng. Methodol.*, June 2025. Just Accepted.
17. Shuhua Deng, Zhangping Yin, and Xieping Gao. Investigating vulnerabilities in openflow discovery protocol: Novel attacks and their defense. *IEEE Transactions on Dependable and Secure Computing*, 22(6):6376–6388, 2025.
18. M.Sahaya Sheela, R. Suganthi, S. Gopalakrishnan, T. Karthikeyan, K. Jyothi, and K. Ramamoorthy. Secure routing and reliable packets transmission in manet using fast recursive transfer algorithm. *Babylonian Journal of Networking*, 2024:78–87, 06 2024.
19. Muhammad Numan, Akif Zia Khan, Mansoor Asif, Sarmad Majeed Malik, and Kashif Imran. Exploiting the inherent flexibility in transmission network for optimal scheduling, wind power utilization, and network congestion management. *IEEE Access*, 9:88746–88758, 2021.
20. Christian Blum. Ant colony optimization: A bibliometric review. *Physics of Life Reviews*, 51:87–95, 2024.

21. Yusuf Alper Kaplan. Forecasting of global solar radiation: A statistical approach using simulated annealing algorithm. *Engineering Applications of Artificial Intelligence*, 136:109034, 2024.
22. Bushra Alhijawi and Arafat Awajan. Genetic algorithms: Theory, genetic operators, solutions, and applications. *Evolutionary Intelligence*, 17(3):1245–1256, 2024.
23. Ayodeji Olalekan Salau and Melesew Mossie Beyene. Software defined networking based network traffic classification using machine learning techniques. *Scientific Reports*, 14(1):20060, 2024.
24. Daniela M Casas-Velasco, Oscar Mauricio Caicedo Rendon, and Nelson LS da Fonseca. Intelligent routing based on reinforcement learning for software-defined networking. *IEEE Transactions on Network and Service Management*, 18(1):870–881, 2020.
25. Sandra Sendra, Albert Rego, Jaime Lloret, Jose Miguel Jimenez, and Oscar Romero. Including artificial intelligence in a routing protocol using software defined networks. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 670–674, 2017.
26. Xinqiang Chen, Ruiyang Hu, Kai Luo, Huafeng Wu, Salvatore Antonio Biancardo, Yiwen Zheng, and Jiangfeng Xian. Intelligent ship route planning via an a* search model enhanced double-deep q-network. *Ocean Engineering*, 327:120956, 2025.
27. Juan Moreno-Malo, Juan-Luis Posadas-Yagüe, Juan Carlos Cano, Carlos T Calafate, J Alberto Conejero, and Jose-Luis Poza-Lujan. Improving traffic light systems using deep q-networks. *Expert Systems with Applications*, 252:124178, 2024.
28. Yi-Ren Chen, Amir Rezapour, Wen-Guey Tzeng, and Shi-Chun Tsai. Rl-routing: An sdn routing algorithm based on deep reinforcement learning. *IEEE Transactions on Network Science and Engineering*, 7(4):3185–3199, 2020.
29. Hao Zhang, Linfeng Tang, Xinyu Xiang, Xuhui Zuo, and Jiayi Ma. Dispel darkness for better fusion: A controllable visual enhancer based on cross-modal conditional adversarial learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26477–26486, 2024.
30. Qiankun Zuo, Huisi Wu, C. L. Philip Chen, et al. Prior-guided adversarial learning with hypergraph for predicting abnormal connections in alzheimer's disease. *IEEE Transactions on Cybernetics*, 54(6):3652–3665, 2024.
31. Xuhui Liu, Bohan Zeng, Sicheng Gao, Shanglin Li, Yutang Feng, Hong Li, Boyu Liu, Jianzhuang Liu, and Baochang Zhang. Ladiffgan: Training gans with diffusion supervision in latent spaces. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1115–1125, 2024.
32. Willone Lim, Kelvin Sheng Chek Yong, Bee Theng Lau, and Colin Choon Lin Tan. Future of generative adversarial networks (gan) for anomaly detection in network security: A review. *Computers & Security*, 139:103733, 2024.
33. Chuyao Wang and Nabil Aouf. Explainable deep adversarial reinforcement learning approach for robust autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 10(4):2551–2563, 2025.
34. Ao Qu, Yihong Tang, and Wei Ma. Adversarial attacks on deep reinforcement learning-based traffic signal control systems with colluding vehicles. *ACM Trans. Intell. Syst. Technol.*, 14(6):113:1–113:22, November 2023.
35. Juan Chafla Altamirano, Mariem Guitouni, Hassan Hassan, and Khalil Drira. Routing optimization based on drl and generative adversarial networks for sdn environments. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, pages 1–5, 2024.
36. Jingbo Li, Xingjun Zhang, Jia Wei, Zeyu Ji, and Zheng Wei. Garlsched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems. *Future Generation Computer Systems*, 135:259–269, 2022.

37. Mansour Selseleh Jonban, Luis Romeral, Mousa Marzband, and Abdullah Abusorrah. A reinforcement learning approach using markov decision processes for battery energy storage control within a smart contract framework. *Journal of Energy Storage*, 86:111342, 2024.

Ping He is an Engineer at State Grid Suzhou Power Supply Company of China. His research interests include flow control and optimization, electric power information network Quality of Service, artificial intelligence, and machine learning.

Dongsheng Jing is a Senior Engineer at State Grid Suzhou Power Supply Company of China. He is majored in computer application technology. His research interests include computer network, electric power information network Quality of Service, intelligent operation and maintenance, artificial intelligence, and machine learning.

Baozhen Qi is a Senior Engineer at State Grid Suzhou Power Supply Company of China. He is majored in computer application technology. His research interests include substation network, flow control and optimization, artificial intelligence, and machine learning.

Yu Yang is a Senior Engineer at State Grid Suzhou Power Supply Company of China. He is majored in software engineering. His research interests include intelligent operation and maintenance, network security, information security, and artificial intelligence.

Jingsong Xue a Senior Engineer at State Grid Suzhou Power Supply Company of China. His research interests include intelligent operation and maintenance, information and communication, electric power information network Quality of Service, network security, and artificial intelligence.

Received: October 15, 2025; Accepted: November 30, 2025.

Computer Science and Information Systems 23(1):321–341 https://doi.org/10.2298/CSIS251027010R

Transformer Substation Network Disconnection Prediction via Semantic Reasoning with Causal Modeling

Jie Ren, Xiaojun Yao, and Hong Chen

Suzhou Suneng Group Co., LTD
215004 Suzhou Jiangsu, China
{rengj9, yaoxj1,ch_jt_sz}@js.sgcc.com.cn

Abstract. Reliable communication networks are indispensable for the stable operation of smart grids and substations. Currently, WAPI networks have been widely adopted in relevant scenarios. Nevertheless, WAPI networks are confronted with disconnection risks attributed to complex network topologies, dynamic traffic fluctuations, and external environmental disturbances. Most methods rely on correlation analysis and lack causal interpretability, which restricts their effectiveness in root-cause localization and preventive maintenance practices. To address the problem, we propose a disconnection prediction approach that integrates prompt-driven semantic reasoning with structured causal analysis. The approach constructs a causal event graph that models semantic, temporal, and topological dependencies across devices and alarm sequences after extracts heterogeneous information to unified event representation. Based on the established graph, an inference module combines causal path analysis, structural causal models, and counterfactual reasoning to assess the influence of events, predict emerging disconnection risks, and identify plausible root causes with coherent and interpretable justification. By tightly coupling semantic abstraction with causal reasoning, the proposed approach provides a proactive, explainable, and extensible mechanism for anticipating network disruptions and supporting informed maintenance decisions. Experiments demonstrate that the proposed approach improves prediction accuracy and interpretability, verifying its value for smart grid communication networks.

Keywords: Causal Inference, Network Disconnection Prediction, Root Cause Analysis, Incident Causality Graph, Substation, Disaster Recovery.

1. Introduction

The developing energy transition and the advancement of power system intelligence have made smart grids and substations vital infrastructures for secure and stable power system operation [23]. In these systems, communication networks enable device interaction and control signal transmission, forming the backbone of business continuity and reliability [14] . To meet the stringent security, reliability, and low-latency requirements of such environments, WAPI (Wireless Local Area Network Authentication and Privacy Infrastructure) network has been widely adopted due to its robust encryption, mutual authentication, resistance to common wireless attacks, and ability to maintain stable communication under high traffic and interference. As a result, WAPI-based devices are extensively deployed in substations, supporting secure and dependable operations. However, due to the complexity of topology, traffic fluctuations, and external disturbances, these devices

often experience disconnection events, leading to service interruptions and even cascading failures [25], which undermine both operational safety and economic efficiency.

To achieve high-assurance communication networks, traditional approaches generally rely on post-event response and static redundancy design [2]. However, these approaches are increasingly inadequate under highly dynamic and uncertain operating environments. In contrast, preventive maintenance mechanisms can proactively mitigate potential risks before failures occur, thus reducing the likelihood of network disconnections [18]. Within this framework, network disconnection prediction becomes a critical component: only when potential risks are predicted and explained in advance can operators take timely countermeasures, thereby enhancing the stability and resilience of the overall network. Existing mainstream approaches for network fault detection are predominantly based on correlation modeling, such as analyzing time-series fluctuations, delay drifts, and clustering of anomalous behaviors [21]. While effective to some extent, these approaches cannot generally capture causal mechanisms among variables, making it difficult to distinguish between "fault symptoms" and "fault root causes" [20]. This limitation directly constrains both the accuracy and interpretability of disconnection prediction.

Recently, the emergence of Large Language Models (LLMs) has opened new opportunities for knowledge modeling and semantic reasoning in complex systems [19]. LLMs can extract key semantic events from unstructured sources such as logs, configuration documents, and alarm texts [29]. Moreover, when integrated with causal discovery algorithms (e.g., the Peter-Clark algorithm, also known as PC) and prompt engineering, LLMs can help construct causal dependency graphs and uncover potential fault propagation paths [7]. Leveraging these capabilities, LLMs demonstrate strong potential in enhancing the generalization, interpretability, and timeliness of disconnection risk prediction.

To address the limitations of conventional approaches, we propose a novel network disconnection prediction approach that integrates LLMs-based semantic reasoning with causal modeling for WAPI network devices in smart grid and substation environments. The approach leverages LLMs to unify heterogeneous operational data and construct causal event graphs, while incorporating causal inference techniques such as path analysis, structural causal models, and counterfactual reasoning to identify root causes and predict potential disconnection risks. By combining the semantic strengths of LLMs with the interpretability of causal modeling, the proposed approach enhances adaptability to complex and unseen fault scenarios. Experimental results confirm its effectiveness in improving the accuracy and timeliness of disconnection prediction. Overall, we introduce a LLMs–causal modeling approach for proactive disconnection prediction, offering a practical paradigm for preventive maintenance in mission-critical communication systems of smart grids and substations. This study differs from existing correlation-based approaches by combining the semantic reasoning of LLMs with causal inference, enabling both accurate prediction and interpretable root-cause analysis. The proposed approach unifies heterogeneous data into causal structures and supports proactive disconnection prevention through scalable causal reasoning. The main contributions of our approach are summarized as follows:

(1) We present a unified event representation scheme that transforms heterogeneous network data into structured semantic events, providing a standardized basis for causal inference and prediction.

(2) We propose an LLMs-driven causal graph construction mechanism that leverages prompt engineering to extract causal dependencies among events, combining semantic reasoning with temporal and topological constraints to enhance accuracy and interpretability.
(3) We design a pluggable inference module that integrates causal path analysis, structural causal models, and counterfactual reasoning, enabling the identification of root causes and the prediction of high-confidence disconnection.
(4) We conduct experiments under representative smart grid network scenarios that demonstrate improved prediction accuracy, causal explainability, and response timeliness.

In the subsequent sections, section 2 reviews the relevant prior work, section 3 presents the methodology proposed in this study, section 4 details the experimental setup and configurations, section 5 provides a comprehensive analysis of the experimental results to demonstrate the effectiveness of the proposed approach, and section 6 concludes the study.

2. Related Work

2.1. Network Fault Diagnosis and Prediction

With the continuous growth of network scale and increasing diversity of services, traditional rule- or threshold-based network fault diagnosis struggles to handle the dynamic and complex modern networks effectively. Consequently, machine learning and deep learning have been widely applied to network disconnection prediction and root-cause diagnosis in recent years. They take advantage of network flow, delay, packet loss, and bandwidth utilization to identify potential anomalies or failures [17]. However, traditional machine learning models face challenges in high-dimensional, dynamic, and structured networks, including strong label dependence, insensitivity to topology changes, and limited ability to capture multi-failure dependencies, which restrict their practical effectiveness.

Deep learning approaches offer automatic feature extraction and complex pattern modeling. Recently, various neural networks have been applied to network disconnection prediction tasks. For instance, Alkaberi et al. used CNNs with MLPs to predict software faults [1]; Gupta et al. applied DNNs with hyper-parameter tuning to assess the impact of software faults [10]; and Cheng et al. introduced Attention-LSTM for time series prediction of intermittent faults [6]. While these approaches improve prediction accuracy, they largely remain black-box models, lacking interpretability and clear root-cause identification, and have limited ability to locate critical network links.

Recently, causal reasoning has been increasingly incorporated into network fault diagnosis to enhance interpretability and reasoning. Causal approaches model dependencies between variables and identify true trigger sources and propagation paths. For example, Ghosh et al. [9] proposed a cascading disconnection prediction approach using causal graphs for early warning in power transmission networks. Li et al. [12] designed a root-cause analysis approach for online service systems using causal Bayesian networks, which significantly improved response accuracy and speed.

These studies indicate that, although machine learning and deep learning show initial success in network fault diagnosis, they still struggle with modeling causal structures, handling multi-hop dependencies, and providing interpretable insights for practical operations.

2.2. Causal Analysis

Causal analysis provides an effective means to compensate for these deficiencies and has gradually become a significant research direction in the field of intelligent operation and maintenance. With the rapid development of general artificial intelligence tools, such as large models, causal analysis approaches are also gradually integrating with multimodal intelligence components, including natural language understanding and knowledge extraction. This integration provides a new technological path to realize network disconnection prediction systems with strong interpretability and generalization capabilities.

Causal analysis algorithms are classified into three categories: constraint-based, scoring-based, and non-Gaussian assumption-based. The PC algorithm is a typical constraint-based causal discovery approach, in which dependencies between variables are inferred through conditional independence tests. The approach consists of two stages: first, constructing an undirected causal skeleton graph, then determining the direction of some edges based on a series of rules, and finally forming a partially directed acyclic graph, which is suitable for data with both discrete and continuous variables [22][15].

In contrast, the GES algorithm (Greedy Equivalence Search) is a scoring-based search approach that does not require independence tests. It performs greedy optimization using scoring functions (e.g., Bayesian Information Criterion, also known as BIC) by performing structural modification operations (e.g., adding edges, deleting edges, reversing edges) in the equivalence class space to find the optimal Bayesian network structure. The approach is efficient but prone to fall into local optimality [11][16].

LiNGAM algorithm (Linear Non-Gaussian Acyclic Model) is a linear causal discovery approach based on structural equation modeling, which takes advantage of the non-Gaussian characteristics of the variables and recovers the directed acyclic structure among the variables by Independent Component Analysis (ICA), which breaks through the limitation that the direction of causality is not identifiable under the Gaussian assumption [24][4]. These approaches have been widely applied in the fields of biomedicine, social sciences, and economic modeling, providing important tools for understanding causal mechanisms in complex systems.

2.3. Large Language Models

In recent years, a large number of unstructured logs and heterogeneous alarms, such as syslog, SNMP Trap, NetFlow, etc., have been generated in network operation and maintenance scenarios, which are semantically diverse and confusing, and are often difficult to be uniformly abstracted and fused by traditional rules or keyword matching approaches. LLMs such as GPT, LLaMA, and Claude have powerful contextual understanding and linguistic reasoning capabilities [3], which can automatically map these unstructured texts into unified fault event labels, thus improving the quality of event normalization and cross-source fusion capabilities [5].

Through Chain-of-Thought (CoT) and Prompt Chaining techniques, LLMs can analyze temporal and semantic logical relationships in event sequences to help identify potential causal trigger paths. It has been demonstrated that the in-context learning approach using GPT-4 can significantly enhance the accuracy and readability of automated root cause analysis, eliminating the need for fine-tuning, in the context of fault analysis in cloud platforms [28]. In addition, studies have also designed reliability assessment

mechanisms, such as the PACE LM framework [27], to calibrate the confidence level of LLMs through the cue-enhancement and retrieval-enhancement (RAG) strategy, which effectively reduces the risk of generating "hallucinatory" information and improves the usability and security in real system scenarios.

LLMs have significant advantages in log abstraction, causal reasoning, disconnected prediction, and natural language interpretation, making them an important technical component of network fault diagnosis and prediction systems.

In addition to the above applications, recent research has explored the integration of LLMs with causal graph construction and preventive maintenance frameworks. By combining LLMs-based semantic reasoning with structural causal models, these approaches enable proactive disconnection prediction, root-cause identification, and explainable fault propagation analysis. Such integration not only enhances the interpretability of automated diagnostics but also provides practical guidance for network operators in large-scale and complex infrastructures, highlighting the potential of LLMs as a core component in intelligent network operation and maintenance systems.

3. Methodology

This section proposes a disconnection prediction approach for network devices that integrates a large model and a causal reasoning approach, aiming to solve the problems of strong dependence on structural assumptions, weak scalability, and limited predictive foresight in traditional causal modeling. With LLMs as the core, the proposed approach automatically generates causal event maps based on multi-source alarm data and structural information. It combines with a downstream pluggable root cause analysis module to realize real-time localization and prediction of disconnected faults.

3.1. Overview

As shown in Fig. 1, our proposed disconnected prediction approach comprises three core stages: data preprocessing and unified representation, large model-driven causal mapping construction, and a pluggable root cause localization and disconnection prediction mechanism. The goal of this process is to fully utilize the advantages of LLMs in processing unstructured data and reasoning about contextual relationships, while retaining the explanatory power and decision support capabilities of causal modeling in complex systems.

In the data preprocessing stage, we unify the abstraction and standardization of heterogeneous information, including network device logs, status monitoring indicators, and topology. By uniformly modeling information such as Syslog, SNMP alarms, and link state changes in the network, we establish a structured event flow that provides a consistent basis for subsequent causal modeling.

Subsequently, we design a prompt-based event recognition and causality extraction mechanism that leverages the semantic understanding and generation capabilities of LLMs. The large language model can generate a directed causal graph reflecting the dependencies between alarm events by receiving historical event context, a priori network structure, and system configuration information.

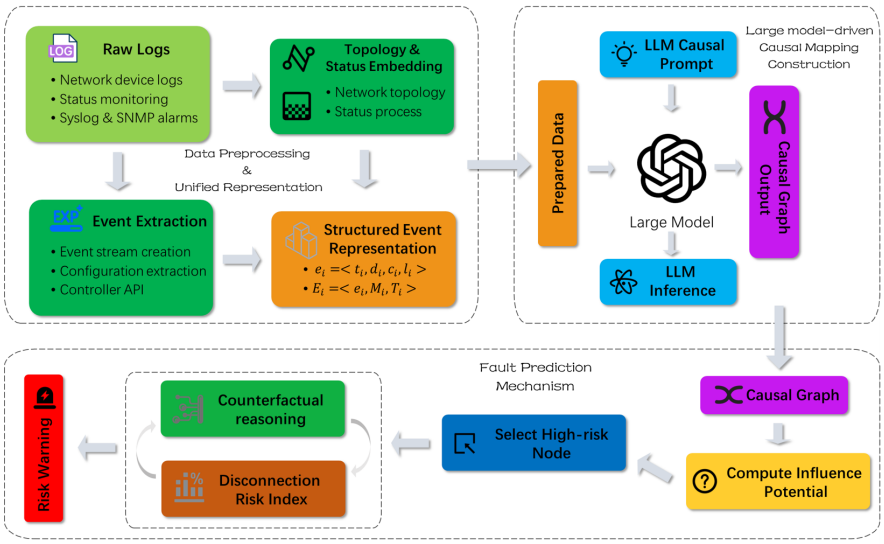


Fig. 1. The overview of our approach. The disconnected prediction approach proposed comprises three core stages: data preprocessing and unified representation, large model-driven causal mapping construction, and a pluggable root cause localization and disconnection prediction mechanism.

Finally, based on the completion of causal graph construction, we introduce a set of plugin causal inference modules, including multiple algorithms such as path analysis, structural causal modeling (also known as SCM), and counterfactual inference, among others, for identifying potential disconnection triggers and enabling high-confidence risk warnings in future windows. This multi-strategy fusion design ensures the approach's stability and migratability in different types of network environments.

Our approach constructs a network disconnection prediction framework that offers high generalization, interpretability, and predictive ability through the deep integration of large models and causal modeling. The following section introduces the design and implementation details of key modules, including causal event graph construction and the root cause inference mechanism.

3.2. LLMs-based causal event graph construction

In network operation, maintenance, and fault analysis, the first step requires accurately constructing a causal map between device anomalies. The traditional approach relies on statistical tests (e.g., conditional independence tests) and structural learning algorithms (PC, GES). These models typically require variables to satisfy specific distributional assumptions and are poorly suited for network environments with complex structures and sparse data. To this end, we propose an automatic causal event graph construction ap-

proach centered on the LLMs, which replaces the heavy dependency test and graph structure generation process in traditional algorithms.

Structured Event Representation. In the actual network operation and maintenance environment, fault data exists in various forms, including Syslog logs, SNMP Traps, NetFlow messages, and link state monitoring. These data sources have obvious heterogeneity. These data sources are obviously heterogeneous, comprising both structured metric information and a large amount of unstructured textual content, such as alarm descriptions and log records. To ensure the accuracy of subsequent causal inference and generation structure, we will construct a unified, structured event representation mechanism to support the approach's subsequent steps effectively.

We adopt a standardization approach based on event quaternions to uniformly abstract network operation and maintenance data from different sources into structured alarm event units. Each base alarm event is represented in the following form:

$$e_i =< t_i, d_i, c_i, l_i >, \tag{1}$$

where t_i denotes the timestamp of the event, which is used to indicate the timing information of the event, d_i denotes the device number or unique identifier number that the event belongs to, i.e., device ID; c_i denotes the type of the event or the alarm category, including "link interruption", "route drift", etc.; denotes the severity level of the event, which is graded from 1 to 5, with higher levels indicating more seriousness. c_i denotes the type of event or alarm category, including "link interruption", "route drift", etc.; l_i denotes the severity level of the event, which is divided into 1 to 5 levels, with the higher level denoting the more serious.

As an example, a real-world scenario in which device R1 detects a severe link anomaly at 13:12 can be represented as:

$$e =< 2024-07-15\ 13:12:15,\ R1, linkinterruption, 5 > . \tag{2}$$

To enhance the contextual connectivity of the event, we expansively introduce two additional fields M_i and T_i, the former denoting a snapshot of the device state at the time of this base event, such as the device's bandwidth usage and CPU utilization at the time of the event, etc., and the latter denoting the network topology neighbors at the time of the event, which form an expanded event with the base event, which can be represented as follows:

$$E_i =< e_i, M_i, T_i > . \tag{3}$$

Such an event representation preserves the important contextual features of network anomalies on the one hand, and facilitates inter-event comparisons, ordering, and relationship modeling on the other. The structured representation greatly reduces the ambiguity of language model inputs and provides a standard, unified semantic foundation for subsequent causal cue design, graph structure generation, and path inference.

Causal Edge Generation. Based on structured event representation, we further utilize LLMs to reason about potential causal relationships between events and construct a directed causal event graph accordingly. The core of the process lies in the use of prompt

The following are network events during a certain period of time, the event format is: $E_i =< e_i, M_i, T_i >, e_i =< t_i, d_i, c_i, l_i >$,

{Event Symbol Explanation},

determine which events may lead to other events:

[Event 1]:<<2024-07-15 13:02:10, R1,Link Outage,5>,{CPU:0.8,Interface Utilization:0.87,Packet Loss:0.02},[R2,R3]>>

[Event 2]:<<2024-07-15 13:03:19, R2,Route Drift,3>,{CPU:0.4,Interface Utilization:0.82,Packet Loss:0.0},[R1,R4]>

...

Please output the possible causal relationships between events in the format [event i] -> [event j].

Fig. 2. Prompt template used to convert structured event sequences into natural-language queries that guide the LLMs in inferring causal relationships between events.

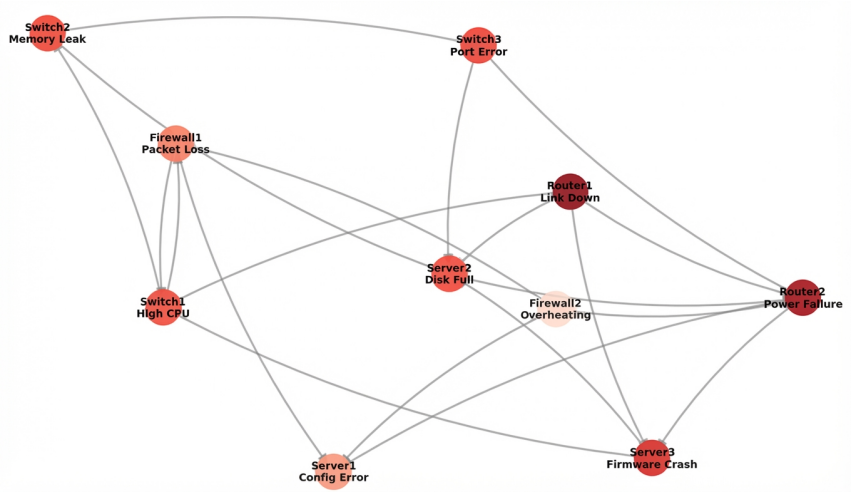


Fig. 3. Illustration of the resulting structural causal graph constructed from validated event dependencies.

engineering to guide the LLMs to identify causal dependencies between events, thus replacing traditional structural learning algorithms such as the PC approach. By combining temporal, semantic, and structural information, the generated causal graph can be more closely related to the fault propagation paths in real networks.

First, we transform the event sequences collected within the time window into a unified natural language input template, which is then fed to the LLMs for causal edge inference. Each event in the template is presented in either the base event format or the extended event format, as described in the previous section, ensuring that the model has sufficient contextual information. A sample prompt template is shown in Fig.2.

After receiving the cue, LLMs will output a set of candidate causal edges $E_{causal} = \{(e_i \rightarrow e_j)\}$ based on the context. To enhance the credibility of the edges and mitigate the noise in the graph structure, we have developed a three-stage causal edge scoring and screening mechanism.

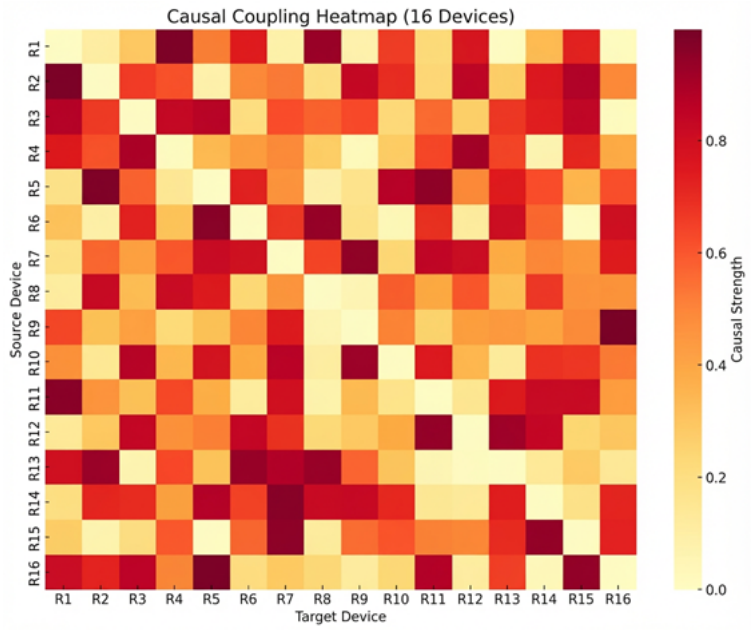


Fig. 4. Coupled heat map visualizing the pairwise causal influence scores between events in the final graph.

First, for each candidate edge, we define the following causal confidence scoring function:

$$\text{score}(e_i \to e_j) = \alpha \cdot \text{sim}(c_i, c_j) \quad + \beta \cdot e^{-\lambda(t_j - t_i)} \quad + \gamma \cdot \text{adj}(d_i, d_j), \tag{4}$$

where sim (c_i, c_j)denotes the cosine similarity of two events in the semantic space, which the LLMs calculate after generating the event embedding; $e^{-\lambda(t_j - t_i)}$ is the temporal decay factor, which ensures that the causal direction adheres to the principle of temporal sequentiality; adj (d_i, d_j), on the other hand, indicates the proximity of two event-generating devices on the network topology. The weights of the three terms α, β, γ can be adjusted according to the actual application to strengthen the model perception.

Second, for causal edges with scores higher than the threshold τ, we further guide the LLMs to generate explanatory text, which is used to verify whether the model outputs have logical rationality, e.g., ask the model, "Why does event e_i cause event e_j?", and if the model cannot reasonably explain a causal relationship, e.g., the reason is ambiguous or semantic conflict, the edge will be eliminated. This step enhances the interpretability of the causal graph and reduces the inference error caused by "language illusion".

Finally, to avoid structural conflicts, we perform consistency checking in conjunction with network device topology graphs to remove unreasonable edges. This procedure consists of two constraints: loop exclusion and topology direction constraints.

Loop exclusion means that if adding the edge $(e_j \to e_i)$ would form a closed loop, it is removed to maintain the topologically orderable nature of the graph.

The topological direction constraint requires that if events e_i and e_j originate from devices d_i and d_j, respectively, then a causal edge is retained only when the inter-device hop count $dist_T(d_i, d_j) \leq 2$ and the topological direction between them is either downstream or at the sibling level. The process can be formally expressed as follows:

$$(e_i \to e_j) \in E_{\text{causal}} \Leftrightarrow \begin{cases} \text{acyclic after addition} \\ d_T(d_i, d_j) \leq 2 \\ \delta_T(d_i, d_j) \in \{\downarrow, =\} \end{cases} \tag{5}$$

Graph Construction Mechanisms. After completing causal edge extraction and multi-stage filtering, the remaining high-confidence edges are aggregated to form the final directed causal graph $G_{causal} = (V, E_{causal})$, where the node set V corresponds to all structured events and the edge set E_{causal} represents validated causal dependencies. Unlike a conventional correlation graph, the resulting structure preserves both temporal directionality and physical topology constraints, ensuring that the generated causal pathways are consistent with real-world fault propagation characteristics in substations and intelligent grid networks.

Fig. 3 shows the directed structural causal event graph of an example by the proposed construction pipeline, where nodes represent semantically encoded events and arrows denote validated causal relations. Fig. 4 provides the corresponding heat map, in which darker cells indicate more substantial causal influence from event e_i to event e_j.

Temporal Enhancement of Causal Graphs. We introduce a sliding time window mechanism to serialize the construction process of the causal event graph for modeling purposes. Given a time span T, we construct the set of events ε_i within the current window at each time step t and generate the corresponding causal graph $G_t = (V_t, E_t)$ on its basis. As the window slides, the model will obtain a series of causal event graph sequences $G_{t-k}, \ldots, G_t$, forming a type of time-evolutionary map that provides the basis for subsequent trend modeling and forecasting.

And then, considering that the causal strength of different edges may change over time, we introduce a time-sensitive dynamic weight function w_t for each causal edge $e_i \to e_j$, which is defined as follows:

$$w_t\left(e_i \to e_j\right) = \eta \cdot s_0 + (1 - \eta) \cdot \text{EMA}_t\left(s_t\right), \tag{6}$$

where s_0 denotes the initial causal confidence score, s_t denotes the frequency of triggering of this edge within the current window (i.e., the number of times that event e_i occurs that causes e_j within the window), EMA_t is an exponential moving average function reflecting the recent trend of dependence, and $\eta \in [0, 1]$ controls the proportion of fusion between the prior and the current dynamics.

3.3. Downstream Causal Reasoning and Disconnected Prediction Mechanisms

After completing the causal graph construction and temporal enhancement based on large models, the system has obtained a sequence of directed causal graphs covering major

Algorithm 1: Causal Graph-Based Disconnection Prediction

Input: $G_t = (V_t, E_t)$, w_t, θ, k
Output: Alert node list alerts
Function *ComputeInfluencePotential*(G_t, w_t)
  $\Psi \leftarrow \{\}$
  foreach *node* $e_i \in V_t$ **do**
    $P \leftarrow$ FindAllPaths(G_t, e_i) path_energies $\leftarrow [\,]$
    foreach *path* $p \in P$ **do**
      prod $\leftarrow 1.0$ **foreach** *edge* $(e_j \rightarrow e_k) \in p$ **do**
        prod $\leftarrow$ prod $\times\, w_t(e_j \rightarrow e_k)$
      path_energies $\leftarrow$ path_energies $\cup$ {prod}
    $\Psi[e_i] \leftarrow \sum$ path_energies
  return Ψ
$\Psi \leftarrow$ ComputeInfluencePotential(G_t, w_t)
sorted_nodes $\leftarrow$ SortDescending(V_t, Ψ)
high_risk $\leftarrow$ sorted_nodes$[1 : \lceil k \cdot |V_t| \rceil]$
alerts $\leftarrow [\,]$
foreach *node* $e_i \in$ *high_risk* **do**
  $G_t^{\neg e_i} \leftarrow$ RemoveNode(G_t, e_i)
  $R_{\text{orig}} \leftarrow$ FindReachableNodes(G_t, e_i)
  broken $\leftarrow 0$
  foreach *node* $e_j \in R_{orig}$ **do**
    if *not IsReachable*$(G_t^{\neg e_i}, e_i, e_j)$ **then**
      broken $\leftarrow$ broken $+ 1$
  $\mathrm{DLI}(e_i) \leftarrow$ broken$/|V_t|$
  if $\mathrm{DLI}(e_i) > \theta$ **then**
    alert $\leftarrow \begin{cases} \text{node} : e_i, \\ \text{DLI} : \mathrm{DLI}(e_i), \\ \text{broken_nodes} : \text{broken}, \\ \Psi : \Psi[e_i] \end{cases}$
    alerts $\leftarrow$ alerts $\cup$ {alert}
return alerts

network alarm events, device states, and structural paths. To transform this causal structure into a usable prediction capability, we design a downstream disconnection prediction mechanism, the core of which consists of root cause identification, risk scoring, and disconnection warning.

The primary goal of disconnection prediction is to identify the "source events" in the system, i.e., the root cause nodes of the current historical events that are most likely to trigger a wide range of impacts. Based on the constructed causal graph $G_t = (V_t, E_t)$, we use a weighted directed path analysis strategy to calculate the causal impact potential of each node:

$$\Psi\left(e_i\right) = \sum_{p \subset \mathcal{P}(e_i)} \prod_{(e_j \rightarrow e_k) \subset p} w_t\left(e_j \rightarrow e_k\right) \tag{7}$$

where $\mathcal{P}(e_i)$ denotes the set of all causal paths reachable from node e_i; $w_t(e_j \to e_k)$ is the causal weight of the edge on the current time window t; a larger value of $\Psi(e_i)$ indicates that the e_i event has a more substantial system-level influence and is prioritized as a candidate for a disconnected risk source.

After identifying the high-risk event nodes in the current window, the system enters the prediction mode to determine whether they will cause the downstream links or devices to be disconnected. At this stage, we introduce the idea of counterfactual reasoning. It estimates the question of "whether the occurrence/non-occurrence of node e_i will cause the disconnection of e_j" through the local perturbation simulation mechanism of the causal graph.

First, select the high-risk node e_i from the current graph G_t; second, construct its "counterfactual version" $G_t^{\neg e_i}$, i.e., set the node e_i did not occur, and recalculate its impact paths; if some nodes e_j are reachable by e_i in G_t but the paths break in $G_t^{\neg e_i}$, it means that $e_i \to e_j$constitutes a potential disconnected propagation.

We define the "Disconnection Risk Index" $\mathrm{DLI}(e_i)$ accordingly:

$$\mathrm{DLI}(e_i) = \frac{\left|\{\, e_j \in V_t : \; e_i \rightsquigarrow_{G_t} e_j \wedge e_i \not\rightsquigarrow_{G_t^{\neg e_i}} e_j \,\}\right|}{|V_t|} \tag{8}$$

This indicator reflects the potential systemic threat level of event e_i. Once $\mathrm{DLI}(e_i)$ exceeds the threshold θ, the system will trigger a risk warning. The flowchart of the approach is shown in Algorithm 1.

4. Experiments Settings

4.1. Data and Preprocessing

The data used in this experiment were collected from network operation and maintenance logs obtained from a large data center and several intelligent substations within a regional power grid. These datasets comprehensively reflect the operating characteristics of both IT infrastructure and industrial communication environments. The data consist of three main parts. The first part includes device-level status information such as CPU utilization, memory usage, interface error rate, BGP neighbor status, and link Up/Down events. The second part contains multi-source alarm events generated by protocols including Syslog, SNMP Trap, NetFlow, and BFD. The third part involves structured information derived from LLDP, configuration extraction, and controller APIs, including network topology and protocol relationship counts, physical topology, and protocol dependency graphs. An example has been shown in Table 1.

All data were uniformly processed and transformed into structured event representations in the form of ternary tuples $< e_i, M_i, T_i >$, and sliding event sequences were constructed within five-minute time windows for causal inference modeling based on the large language model.

4.2. Experimental Platform and Parameters

The hardware environment of the platform used in this experiment is a NVIDIA RTX A6000 GPU, the software environment is Python 3.10, and the primary LLMs tool is

Table 1. Example of network operation and maintenance data used in .

(A) Device Status Information						
Timestamp	Device	CPU(%)	Mem(%)	Interface	Error Rate	Link
10:23:01	RT-01	38	62	ge-0/0/1	0.2‰	Up
10:23:01	SW-22	12	48	eth1/3	1.5‰	Down
(B) Multi-source Alarm Events						
Timestamp	Device	Type	Severity	Description		
10:25:12	RT-01	Syslog	Warning	CRC errors on ge-0/0/1		
10:25:14	SW-22	SNMP Trap	Major	Link eth1/3 down		
(C) Topology and Protocol Dependencies						
Local Device	Local Port	Remote Device	Remote Port	Protocol Dependency		
RT-01	ge-0/0/1	SW-22	eth1/3	BGP $\rightarrow$ {BFD, OSPF}		

OpenAI GPT-4 API for causal extraction and interpretation verification, and the other tools are NetworkX for performing graph structure construction, and Neo4j for graph storage and interactive query.

The dataset size of this experiment comprises 10,000 windows, approximately 1,400 link failure events, and 15 abstract alarm types. LLMs' prompt templates are constructed based on alarm content, time, and neighbor structure, and batch reasoning and caching mechanisms are used to improve efficiency.

4.3. Metrics

To evaluate the model performance, we design the following key metrics:

(1) **Root Cause Accuracy:** whether the earliest cause event inferred by LLMs covers the actual faulty equipment or not;
(2) **Lead Time:** the average time for the first warning to be issued before the actual occurrence of a fault event;
(3) **Edge Precision:** the proportion of the edges predicted by LLMs for which there is a real causal relationship;
(4) **Edge Recall:** the proportion of real causal relationships correctly identified by LLMs;
(5) **Causal Explainability:** the proportion of edges that are validated by natural language feedback;
(6) **False Positive Rate:** the proportion of non-root-cause devices that are misidentified as risk sources.

5. Results and Analysis

This section aims to validate the effectiveness and practicality of the proposed network device disconnection prediction approach that integrates LLMs and causal reasoning in real scenarios. We simulate typical device failure scenarios by constructing multi-source

Table 2. Experimental results demonstrating how sample size influences the accuracy, interpretability, and computational cost of the proposed causal graph generation approach, with estimated standard deviations.

Sample	Edges Extracted	Precision	Recall	Explainability	Processing Time (s)
500	84 ± 4	0.74 ± 0.03	0.69 ± 0.03	0.72 ± 0.03	3.4 ± 0.2
1,000	110 ± 4	0.81 ± 0.02	0.74 ± 0.02	0.76 ± 0.02	5.8 ± 0.3
5,000	138 ± 3	0.86 ± 0.02	0.82 ± 0.02	0.84 ± 0.02	13.5 ± 0.5
10,000	142 ± 3	0.88 ± 0.01	0.85 ± 0.01	0.87 ± 0.01	22.9 ± 0.7
50,000	146 ± 2	0.90 ± 0.01	0.87 ± 0.01	0.91 ± 0.01	88.4 ± 2.0
100,000	148 ± 1	0.91 ± 0.01	0.88 ± 0.01	0.92 ± 0.01	174.1 ± 3.0

Table 3. Comparison of root-cause localization performance across baseline approaches and the proposed LLMs-based approach, including estimated standard deviations for accuracy, prediction lead time, and false positive rate.

Approaches	Root Cause Accuracy ↑	Prediction Lead Time (min) ↑	False Positive Rate ↓
NetRCA	$75.0\% \pm 1.5\%$	3.0 ± 0.3	0.150 ± 0.010
REASON	$81.5\% \pm 1.2\%$	5.0 ± 0.4	0.120 ± 0.008
RUN	$83.0\% \pm 1.0\%$	5.5 ± 0.3	0.110 ± 0.007
Attention-LSTM	$89.5\% \pm 0.8\%$	-	-
Ours	$\mathbf{90.2\% \pm 0.7\%}$	$\mathbf{7.1 \pm 0.4}$	$\mathbf{0.074 \pm 0.005}$

alarm flow datasets in real network environments. We systematically evaluate the quality of causal graph construction, root cause localization accuracy, alerting capability, and module effectiveness.

5.1. Quality Assessment of Causal Graphs Evaluation

We input samples of different sizes into the LLMs to evaluate their performance in automatically generating causal edges.

The results in Table 2 show that larger sample sizes substantially enhance the model's ability to extract causal edges: the number of identified relations increases from 84 to 148. It begins to stabilize once the sample size exceeds 50,000. Precision and recall also improve steadily (from 0.74/0.69 to 0.91/0.88), indicating clear gains in both accuracy and coverage. Complementing these numerical results, Fig. 5 visualizes the same trends and highlights the smooth, monotonic improvement of all metrics as data scale grows. The curves gradually flatten at larger sample sizes, illustrating the diminishing incremental benefits and the model's convergence behavior.

Meanwhile, the causal interpretation rate also increases from 0.72 to 0.92, indicating that the causal edges generated by LLMs are increasingly rational and semantically consistent, providing strong support for interpretability in practical applications. Although the processing time increases significantly with the sample size, it is within the acceptable range, indicating that the approach has good scalability while maintaining the quality of graph building and is suitable for large-scale network disconnection prediction scenarios.

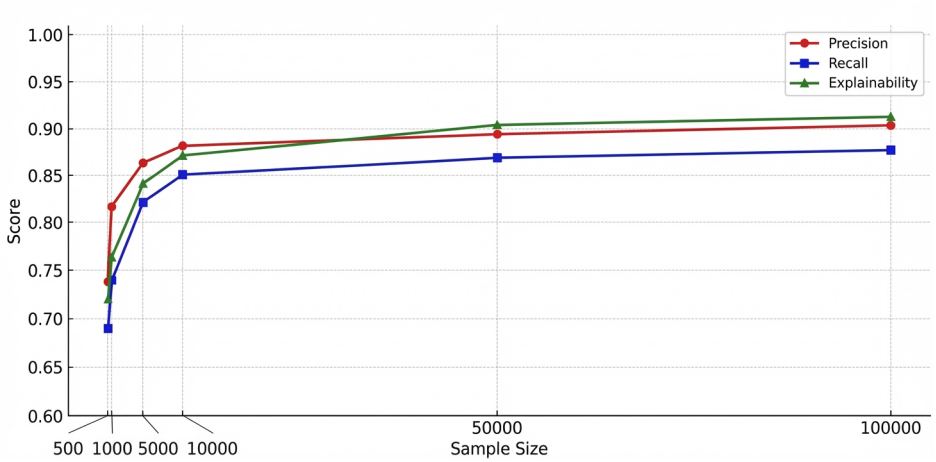


Fig. 5. Causal Graph Construction Performance Evaluation

The comparison in Fig. 6 verifies that the larger the sample size, the more stable and complete the causal structure becomes, which helps to improve the accuracy of root cause localization and the explanation of disconnected propagation paths.

5.2. Root Cause Localization Performance Evaluation

To comprehensively evaluate the performance of the proposed approach in root cause localization and disconnection risk prediction, we select four representative studies as baseline approaches, including traditional feature-based ensemble models, graph neural networks, neural Granger causal discovery, and generative adversarial frameworks:

(1) **NetRCA[26]**: A root cause localization approach that integrates multi-type feature engineering, data augmentation, and ensemble learning (including XGBoost, rule-based learning, and graph algorithms). It is primarily designed for network failure scenarios with limited samples.
(2) **REASON[20]**: A hierarchical graph neural network that constructs cross-layer causal structures and combines random walk with extreme value theory for root cause identification. It is particularly suited for analyzing multi-level fault propagation.
(3) **RUN[13]**: A approach based on neural Granger causal discovery and contrastive learning, further enhanced with PageRank to localize root causes of microservice failures. Experiments on both synthetic and real microservice datasets demonstrate superior performance over traditional approaches.
(4) **FaultGuard[8]**: A generative adversarial framework tailored for smart grids, which exhibits high robustness and ensures accurate disconnection prediction and detection even under adversarial attacks.

As shown in Table 3, the proposed approach achieves the best performance across all three key evaluation metrics. First, in terms of **root cause localization accuracy**, our approach attains 90.2%, outperforming RUN (83.0%) and REASON (81.5%), and

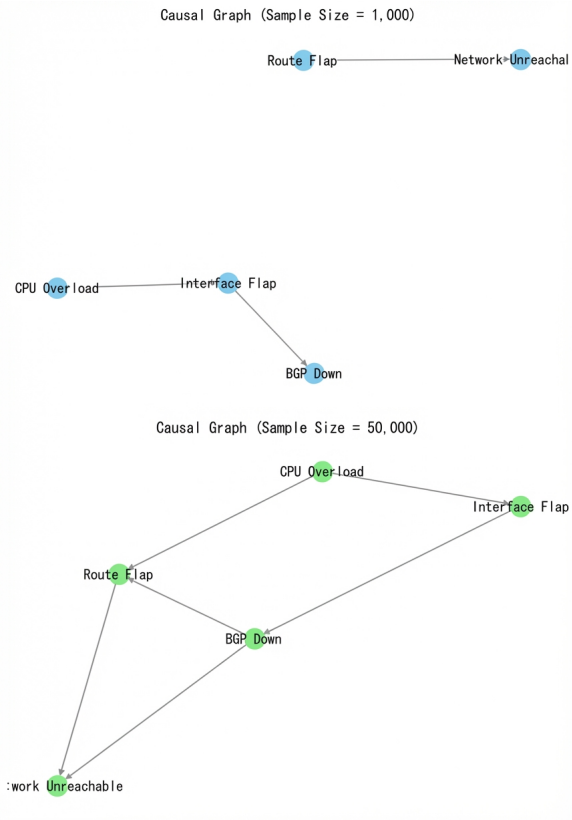


Fig. 6. Effect of sample size on the stability and completeness of the causal structure, which improves root cause localization and the explanation of disconnected propagation paths.

slightly surpassing FaultGuard (89.5%). This demonstrates the superior capability of our approach in handling complex network failure scenarios. Second, regarding **prediction timeliness**, the approach issues warnings approximately 7.1 minutes before disconnection events occur, significantly ahead of REASON and RUN (5.0–5.5 minutes), thereby providing a longer response window for preventive maintenance. Third, in terms of **false alarm control**, our approach achieves a false positive rate of only 7.4%, substantially lower than REASON (12%), RUN (11%), and NetRCA (15%). This highlights the advantage of the causal reasoning module in mitigating "language hallucinations" and reducing noise-induced misjudgments.

Overall, these results clearly demonstrate the benefits of deeply integrating LLMs with causal modeling: the proposed approach achieves significant breakthroughs in root cause localization accuracy, responsiveness, and predictive reliability, suggesting its potential to serve as a new paradigm for preventive maintenance in smart grids and critical communication systems.

Table 4. Comprehensive ablation study showing the impact of individual components and their combinations on model performance. SCC, NLE, and PO denote Structural Consistency Check, Natural Language Explanation, and Prompt Optimization, respectively. RCA, PLT, and FPR represent Root Cause Accuracy, Prediction Lead Time, and False Positive Rate, with estimated standard deviations.

Configuration	SCC	NLE	PO	RCA ↑	PLT (min) ↑	FPR ↓
Full Model	Y	Y	Y	**90.2% ± 0.7%**	**7.1 ± 0.4**	**0.074 ± 0.005**
w/o Consistency	N	Y	Y	80.1% ± 1.2%	6.6 ± 0.4	0.120 ± 0.010
w/o Explanation	Y	N	Y	83.4% ± 1.0%	6.9 ± 0.3	0.103 ± 0.008
w/o Prompt Optimization	Y	Y	N	78.0% ± 1.5%	5.3 ± 0.5	0.131 ± 0.012
w/o SCC + NLE	N	N	Y	75.8% ± 1.3%	6.2 ± 0.4	0.135 ± 0.012
w/o SCC + PO	N	Y	N	72.5% ± 1.5%	5.8 ± 0.5	0.142 ± 0.013
w/o NLE + PO	Y	N	N	77.1% ± 1.4%	5.9 ± 0.4	0.128 ± 0.011
w/o SCC + NLE + PO	N	N	N	68.0% ± 1.8%	5.0 ± 0.5	0.155 ± 0.015

These results demonstrate that the introduction of LLMs into network disconnection prediction and root cause localization tasks can significantly enhance the interpretability and foresight of the models, providing a more practical technical path for risk prevention and control in highly available network systems.

5.3. Disconnection Propagation Path Analysis

To further validate the explanatory ability and causal path modeling effect of the large model in the disconnection prediction task, we selected two typical types of network failure events from the experimental set, demonstrated the causal propagation paths automatically constructed by the model, and analyzed them in combination with the network operation data and topology.

In a particular core switching cluster, the physical layer fluctuations occur continuously on the device interfaces, and the model automatically generates the following causal chain:

```
[Interface Flap] → [OSPF Adjacency Lost] → [Route
       Withdrawal] → [Service Unreachable]
```

The explanatory text in the path states:

```
"OSPF adjacency outages are usually caused by physical link
          instability or interface restarts."
    "Route convergence failures cause service paths to
    disappear, triggering service unreachable alarms."
```

In the edge access device. The system monitors a continuous spike in resource utilization. Eventually, the user application experiences frequent timeouts. The causal path generated by LLMs is as follows:

```
[CPU Overload] → [Routing Loop Detected] → [Packet Loss] →
                  [Application Timeout]
```

The path reveals the complete mechanism that begins with the underlying device state anomaly and gradually leads to the control plane, forwarding path, and ultimately affects the application layer. The explanatory text in the path states:

```
"High CPU load may lead to delays in processing control
  packets, which generates routing convergence jitter"
"Routing loops trigger forwarding congestion and packet
 loss, which ultimately affects upper-layer application
                    response times."
```

Similar resource bottleneck paths are frequently identified in multiple experimental samples, demonstrating the stability and generality of the LLMs approach to capture the propagation of the state protocol-data-service chain.

5.4. Ablation Experiments

To further assess the role of each module in the overall approach, we designed three sets of ablation experiments, excluding key components such as large model cue optimization, structural consistency checking, and natural language causal validation, and observing their impact on root cause prediction ability.

The results presented in Table 4 provide clear evidence of each component's contribution. Removing the Structural Consistency Check (SCC) results in a substantial increase in the false positive rate, underscoring its crucial role in maintaining prediction reliability. Excluding the Natural Language Explanation (NLE) module reduces Root Cause Accuracy (RCA). It slightly shortens the Prediction Lead Time (PLT), demonstrating that semantic validation enhances both interpretability and early-warning capability. Omitting Prompt Optimization (PO) causes the largest drop in RCA, reflecting the importance of carefully designed prompts for stable LLMs' reasoning.

Furthermore, the joint ablation experiments, where two or all three components are removed simultaneously, show a compounded deterioration in performance: RCA declines sharply, PLT decreases, and FPR rises markedly. This highlights the synergistic effect among the modules, where each component not only contributes individually but also reinforces the others to improve overall model robustness and reliability.

Overall, these ablation studies confirm that each sub-module in the proposed approach is indispensable. The SCC ensures low false positives, the NLE provides causal interpretability, and PO stabilizes LLMs' outputs, together achieving enhanced accuracy, interpretability, and robustness of the disconnection prediction system.

6. Conclusion

In this paper, we propose a network device disconnection prediction approach that integrates large language models (LLMs) with causal analysis. The approach leverages structured event abstraction, combines network topology and operational state information, employs LLMs for causality extraction, and is enhanced with consistency checking and causal path interpretation, enabling accurate root-cause identification, interpretable reasoning, and early warning capabilities.

Empirical evaluation of real data center operation and maintenance logs reveals that the proposed approach outperforms traditional rule-based and statistical approaches in terms of causal graph construction quality, root-cause localization accuracy, prediction reliability, and false alarm reduction. These results demonstrate the practical utility of the framework for proactive network management and preventive maintenance in mission-critical infrastructures.

Future work will focus on adaptive prompt engineering, more efficient incremental causal graph construction, improving LLMs' generalization across diverse network environments, and enhancing model security and robustness. With ongoing technological advancements, integrating LLMs-based causal reasoning is expected to become a foundational capability for intelligent, resilient, and automated network operation and maintenance.

Acknowledgments. This work was supported by Science and Technology Project of State Grid Jiangsu Electric Power Co., Ltd., and Science and Technology Project of Suzhou Suneng Group Co., Ltd. under Grant SGSZSNJTKJJS2500967.

References

1. Alkaberi, W., Assiri, F.: Predicting the number of software faults using deep learning. Engineering, Technology & Applied Science Research 14(2), 13222–13231 (Apr 2024)
2. Alvarez-Alvarado, M., Donaldson, D., Recalde, A., Khan, Z., Velasquez, W., Rodríguez Gallegos, C.: Power system reliability and maintenance evolution: A critical review and future perspectives. IEEE Access 10, 51922–51950 (01 2022)
3. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X.: A survey on evaluation of large language models. ACM Trans. Intell. Syst. Technol. 15(3) (Mar 2024)
4. Chen, W., Cai, R., Zhang, K., Hao, Z.: Causal discovery in linear non-gaussian acyclic model with multiple latent confounders. IEEE Transactions on Neural Networks and Learning Systems 33, 2816–2827 (2021)
5. Chen, Y., Xie, H., Ma, M., Kang, Y., Gao, X., Shi, L., Cao, Y., Gao, X., Fan, H., Wen, M., Zeng, J., Ghosh, S., Zhang, X., Zhang, C., Lin, Q., Rajmohan, S., Zhang, D., Xu, T.: Automatic root cause analysis via large language models for cloud incidents. In: Proceedings of the Nineteenth European Conference on Computer Systems. p. 674–688. EuroSys '24, Association for Computing Machinery, New York, NY, USA (2024)
6. Cheng, X., Lv, K., Zhang, Y., Wang, L., Zhao, W., Liu, G., Qiu, J.: Rul prediction method for electrical connectors with intermittent faults based on an attention-lstm model. IEEE Transactions on Components, Packaging and Manufacturing Technology 13, 628–637 (2023)
7. Cohrs, K.H., Diaz, E., Sitokonstantinou, V., Varando, G., Camps-Valls, G.: Large language models for constrained-based causal discovery. In: AAAI 2024 Workshop on "Are Large Language Models Simply Causal Parrots?". pp. 1–9 (2023)
8. Efatinasab, E., Marchiori, F., Brighente, A., Rampazzo, M., Conti, M.: Faultguard: A generative approach to resilient fault prediction in smart electrical grids. In: Detection of Intrusions and Malware, and Vulnerability Assessment: 21st International Conference, DIMVA 2024, Lausanne, Switzerland, July 17–19, 2024, Proceedings. p. 503–524. Springer-Verlag, Berlin, Heidelberg (2024)
9. Ghosh, S.S., Dwivedi, A., Tajer, A., Yeo, K., Gifford, W.M.: Cascading failure prediction via causal inference. IEEE Transactions on Power Systems 40(4), 3361–3373 (2025)

10. Gupta, M., Rajnish, K., Bhattacharjee, V.: Impact of parameter tuning for optimizing deep neural network models for predicting software faults. Sci. Program. 2021, 6662932:1–6662932:17 (2021)
11. Laborda, J.D., Torrijos, P., Puerta, J.M., Gámez, J.A.: Parallel structural learning of bayesian networks: Iterative divide and conquer algorithm based on structural fusion. Knowledge-Based Systems 296, 111840–111858 (2024), https://www.sciencedirect.com/science/article/pii/S095070512400474X
12. Li, M., Li, Z., Yin, K., Nie, X., Zhang, W., Sui, K., Pei, D.: Causal inference-based root cause analysis for online service systems with intervention recognition. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 3230–3240. KDD '22, Association for Computing Machinery, New York, NY, USA (2022)
13. Lin, C.M., Chang, C., Wang, W.Y., Wang, K.D., Peng, W.C.: Root cause analysis in microservice using neural granger causal discovery. In: Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence. pp. 206–213. AAAI'24/IAAI'24/EAAI'24, AAAI Press (2024)
14. Liu, X., Chen, B., Chen, C., Jin, D.: Electric power grid resilience with interdependencies between power and communication networks – a review. IET Smart Grid 3(2), 182–193 (2020)
15. Lv, F., Si, S., Xiao, X., Ren, W.: Modified local granger causality analysis based on peter-clark algorithm for multivariate time series prediction on iot data. Computational Intelligence 40, 1–20 (2024)
16. Nazaret, A., Blei, D.: Extremely greedy equivalence search. In: Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence. pp. 2716–2745. UAI '24, JMLR.org (2024)
17. Raja, E.J., Hossen, J., Ervina, E.M.N., Tawsif, K., Jesmeen, M.Z.H.: Broadband network fault prediction using complex event processing and predictive analytics techniques. Journal of Engineering Science and Technology 15(4), 2289–2300 (2020)
18. Rana, S.: Ai-driven fault detection and predictive maintenance in electrical power systems: A systematic review of data- driven approaches, digital twins, and self-healing grids. American Journal of Advanced Technology and Engineering Solutions 01, 258–289 (02 2025)
19. Wan, G., Lu, Y., Wu, Y., Hu, M., Li, S.: Large language models for causal discovery: Current landscape and future directions. In: Kwok, J. (ed.) Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25. pp. 10687–10695. International Joint Conferences on Artificial Intelligence Organization (8 2025), https://doi.org/10.24963/ijcai.2025/1186, survey Track
20. Wang, D., Chen, Z., Ni, J., Tong, L., Wang, Z., Fu, Y., Chen, H.: Interdependent causal networks for root cause localization. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 5051–5060. KDD '23, Association for Computing Machinery, New York, NY, USA (2023)
21. Wang, F., Jiang, Y., Zhang, R., Wei, A., Xie, J., Pang, X.: A survey of deep anomaly detection in multivariate time series: Taxonomy, applications, and directions. Sensors 25(1), 1–27 (2025)
22. Wang, X., Jiang, S., Li, X., Wang, M.: Causal discovery and fault diagnosis based on mixed data types for system reliability modeling. Complex & Intelligent Systems 11, 1–16 (2025)
23. Yang, Q., Hao, W., Ge, L., Ruan, W., Chi, F.: Farima model-based communication traffic anomaly detection in intelligent electric power substations. IET Cyper-Phys. Syst.: Theory & Appl. 4, 22–29 (2018)
24. Yang, T.L., Lee, K.Y., Zhang, K., Suzuki, J.: Functional linear non-gaussian acyclic model for causal discovery. Behaviormetrika 51(2), 567–588 (2024)
25. Yang, Y., Nishikawa, T., Motter, A.E.: Small vulnerable sets determine large network cascades in power grids. Science 358(6365), 1–8 (2017)

26. Zhang, C., Zhou, Z., Zhang, Y., Yang, L., He, K., Wen, Q., Sun, L.: Netrca: An effective network fault cause localization algorithm. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 9316–9320 (2022)
27. Zhang, D., Zhang, X., Bansal, C., Las-Casas, P., Fonseca, R., Rajmohan, S.: Lm-pace: Confidence estimation by large language models for effective root causing of cloud incidents. In: Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering. p. 388–398. FSE 2024, Association for Computing Machinery, New York, NY, USA (2024), https://doi.org/10.1145/3663529.3663858
28. Zhang, X., Ghosh, S., Bansal, C., Wang, R., Ma, M., Kang, Y., Rajmohan, S.: Automated root causing of cloud incidents using in-context learning with gpt-4. In: Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering. p. 266–277. FSE 2024, Association for Computing Machinery, New York, NY, USA (2024)
29. Zhang, Z., Li, S., Zhang, L., Ye, J., Hu, C., Yan, L.: Llm-lade: Large language model-based log anomaly detection with explanation. Knowledge-Based Systems 326, 114064–114074 (2025), https://www.sciencedirect.com/science/article/pii/S0950705125011098

Jie Ren is a Senior Engineer at Suzhou Suneng Group Co., LTD of China. He has rich experience on power grid technology and application. His research interests include flow control and optimization, electric power information network, Quality of Service, and artificial intelligence.

Xiaojun Yao is a Senior Engineer at Suzhou Suneng Group Co., LTD of China and State Grid Suzhou Power Supply Company of China. He is majored in power grid technology and application. His research interests include computer network, electric power information network Quality of Service, intelligent operation and maintenance, artificial intelligence, and machine learning.

Hong Chen is an Engineer Suzhou Suneng Group Co., LTD of China. He is experienced in power grid intelligent operation and maintenance. His research interests include substation network, flow control and optimization, and artificial intelligence.

Received: October 27, 2025; Accepted: Devember 10, 2025.

Computer Science and Information Systems 23(1):343–368 https://doi.org/10.2298/CSIS251030011G

DICYME: Dynamic Industrial Cyber Risk Modelling Based on Evidence*

Javier García-Ochoa[1], Jaime Rueda[1], Rubén R. Fernández[1], Alberto Fernández-Isabel[1], Isaac Martín de Diego[1], Emilio L. Cano[1], Romy R. Ravines[2], Ovidio López Espinosa[2] and Jaume Puigbó Sanvisens[2]

[1] Rey Juan Carlos University
Department of Computing Science & Statistics, ETSII
C/ Tulipán, s/n, 28933, Móstoles, Madrid (Spain)
{javier.garciaochoa, jaime.rueda, ruben.rodriguez, alberto.fernandez.isabel, isaac.martin, emilio.lopez}@urjc.es

[2] DeNexus Inc.
Boston, United States
{rr, ol, jp}@denexus.io

Abstract. The accelerated pace of digital transformation has significantly reshaped the cybersecurity domain, fostering an interconnected ecosystem in which cyber threats have expanded in both their complexity and scope. Traditional cybersecurity methods are increasingly inadequate for addressing the rapidly evolving threat landscape, emphasizing the critical need for intelligent, adaptive, and proactive defensive strategies. This study introduces Dynamic Industrial Cyber Risk Modelling Based on Evidence (DICYME), a comprehensive system that integrates diverse analytical techniques to identify patterns and characteristics that reveal emerging threat trends, enabling organizations to proactively defend against potential future attacks. Beyond threat detection, DICYME operates as a pipeline that retrieves data from diverse cyber incident reports, specialized databases, and other relevant sources of cyber-related information, applies specialized techniques for victim identification, indicator computation, threat actor profiling, Common Vulnerability and Exposure (CVE) relationship mapping, and ultimately performs the Cyber Risk Quantification (CRQ). This final stage represents the system's most distinctive contribution, as it translates complex analytical outputs into actionable risk insights, empowering organizations to make informed strategic decisions in the face of evolving cyber threats. Alternatively, the system implements an automatic workflow that constructs new datasets of compromised entities, enabling these datasets to be used by all components of the system. Experiments on real cyber incident datasets demonstrate the system's ability to automatically construct high-quality victim profiles and estimate annualized financial risk, offering a scalable and data-driven approach for proactive cybersecurity management.

Keywords: Cyber risk quantification, Machine Learning, Large Language Models, Indicators, Firmographics, Threat actors, Vulnerabilities.

* This is an extended and updated version of the paper "Dynamic Industrial Cyber Risk Modelling based on Evidence (DICYME)" presented at the 10th Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2025), Zaragoza, Spain.

1. Introduction

The exponential growth of digital transformation has fundamentally reshaped the global cybersecurity landscape, creating an interconnected ecosystem in which cyber threats have evolved in both sophistication and scale. The accelerated migration of individuals and organizations toward digital environments, particularly intensified by digital transformation, has expanded the attack surface and created new vulnerabilities across multiple technological layers, including cloud computing, Internet of Things (IoT) devices, social media platforms, and cryptocurrency systems [17]. This dynamic environment has rendered traditional, static cybersecurity strategies increasingly inadequate, as they often fail to anticipate emerging threats or provide real-time insights into the evolving risk profile of an organization.

Organizations face the dual challenge of identifying potential threats and quantifying their impact in a timely manner. Conventional cyber risk assessments rely heavily on manual processes and historical data, which are often incomplete, delayed, or incompatible across sources. This results in a slow, resource-intensive evaluation that struggles to capture the rapidly changing threat landscape and the adaptive and dynamic structure of organizations. Moreover, many existing approaches treat vulnerability and threat intelligence in isolation, failing to integrate multiple data streams into a unified, actionable risk framework.

Given the constant threats that can compromise enterprises across all sectors, there is a critical need for intelligence-driven cybersecurity measures that go beyond traditional methods. Integrating Machine Learning (ML) and Artificial Intelligence (AI) methodologies [10] allows organizations to adopt proactive, adaptive strategies capable of real-time risk quantification and data-driven decision-making. By leveraging the temporal patterns of cyber incidents through techniques such as time-series analysis, it becomes possible to anticipate future attack occurrences and detect anomalies that may indicate imminent threats [13]. This combination of automation, advanced analytics, and predictive modeling provides a foundation for more efficient, accurate, and dynamic cyber risk management.

To address these limitations, this study presents DICYME, a comprehensive system that combines automatic intelligence extraction, advanced ML models, and real-time risk quantification. DICYME leverages both public and private datasets, dynamically incorporating new vulnerabilities, emerging attack techniques, and threat actor behaviors. The system translates this intelligence into quantitative indicators and probabilistic scores, which are then used to estimate the likelihood, frequency, and potential impact of cyber incidents, including both primary and secondary losses.

A key feature of DICYME is the usage of Large Language Model (LLM) to generate interpretable explanations and actionable recommendations, helping stakeholders understand the rationale behind risk assessments and mitigation strategies. Additionally, it is adaptable to different stakeholder needs, supporting interactive visualization and reporting tools suitable for technical teams, executive decision-makers, insurers, and reinsurers, providing tailored insights that facilitate informed, data-driven decisions.

By integrating real-time data extraction, predictive modeling, and dynamic risk quantification, DICYME overcomes the fragmented and static nature of traditional cyber risk assessments. It provides a unified, continuously updated view of cyber exposure, enabling organizations to proactively detect emerging threats, prioritize security investments, and

respond effectively to the evolving digital threat landscape. This integrated approach represents a significant advancement in intelligence-driven cybersecurity, bridging the gap between automated data collection, analytical modeling, and operational decision-making.

The rest of this paper is organised as follows. Section 2 provides the background and related work, covering risk analysis, actor profiling, and CVE relationships. Section 3 introduces the proposed system, including automated data extraction, risk modelling, and quantification, with support from LLMs and AI tools. Section 4 presents a series of experiments evaluating the data gathering process of the system and the CRQ model. Section 5 discusses the lessons learned, highlighting the strengths, limitations, and practical insights gained from the deployment of the system. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Background

The cybersecurity landscape has evolved into a complex ecosystem in which cyber incidents, risk analysis, threat actors, and vulnerabilities are interconnected elements that collectively shape organizational security postures [28]. Understanding these relationships is crucial for developing comprehensive defensive strategies and predictive models [27], as organizations face increasingly sophisticated and frequent attacks that exploit the interdependencies among these elements. Recent research emphasizes the need for holistic approaches that integrate cybersecurity risk management with strategic management practices, leveraging AI and ML techniques with traditional cybersecurity frameworks [19, 27].

In recent years, cyber incidents have been steadily increasing, becoming a common problem affecting organizations across multiple domains. Simultaneously, these incidents have grown in complexity [26], compromising a wide range of companies from different sectors, including sensitive areas such as healthcare and finance. According to Hackmageddon, the number of recorded cyber incidents continues to rise yearly, reaching $4,128$ events in 2023, representing a 35% increase compared to 2022 and a markedly higher growth relative to earlier years [22].

This trend has motivated companies and institutions to develop various techniques and approaches to collect, analyze, and explain cyber incident data, as well as to model the associated risks. At present, there are numerous repositories of cyber incidents that publish detailed information about reported cases, such as the European Repository of Cyber Incidents (EuRepoC) [5] and the Cyber Events Database of the Center for International & Security Studies at Maryland (CISSM) [1]. The reporting of such incidents, which may affect any type of organization, enables the identification of potential attack trends and patterns. This, in turn, supports the anticipation of future threats and provides valuable insights for security teams, decision makers, and other relevant parties who access and analyse these repositories.

2.1. Risk analysis

Based on the foundation of incident data collected in repositories, organizations require systematic approaches to assess and quantify cyber risks. Companies and institutions often conduct cyber risk analyses [24] to identify threat actors that are likely to carry out

attacks, as well as to quantify the probability and frequency of such attacks. Based on the results of this analysis, organizations can make informed decisions about whether to invest in security, what types of security measures to prioritize, and which techniques to adopt to protect themselves against common threats that may affect their specific business sector.

By leveraging databases that host records of cyber incidents affecting companies and institutions, it is possible to conduct comprehensive risk analyses. Probabilistic and statistical analyses [24, 30] are the most commonly applied approaches for assessing these risks, and they can be combined to develop descriptive models. Furthermore, it is possible to build models that function as risk management tools, enabling organizations to reduce the likelihood of threats and their potential impact.

However, these approaches are often based on historical data and present limitations in anticipating novel or rapidly evolving threats. Another important challenge in cyber risk analysis is the limitations or absence of certain relevant data. In many cases, the available repositories or databases do not provide different coverage for specific types of risks, making it necessary to rely on external or private data sources to conduct a more comprehensive analysis. Furthermore, it is mainly a manual process, performed from time to time, requiring effort from analysts that increases cost and reduces responsiveness to emerging threats [25]. In this context, our proposal introduces a system that integrates automated data extraction with real-time risk quantification, providing continuously updated and dynamic assessments tailored to the current threat landscape.

2.2. Threat actors analysis

While risk analysis provides a quantitative framework for understanding cyber threats, identifying and characterizing the specific actors behind these threats represents another critical dimension of cybersecurity research. Threat actors are individuals or organized groups responsible for causing cyber incidents in companies [6] leading to financial losses, reputational damage, sensitive data breaches, and broader security implications. They are often identified to conduct analyses that allow organizations to understand attack patterns, predict future threats, and develop targeted defense strategies.

The identification of threat actors often lacks consensus, as disagreements may arise regarding who should be considered an attacker. In many cases, the analysis tends to prioritize quantitative aspects [31], such as counting how often different threat actors are mentioned in reports, over qualitative considerations such as comparative or contextual evaluation. This quantitative focus, while providing measurable insights, may overlook important behavioral patterns and motivational factors that threat actor activities.

Several studies and frameworks have been proposed to address this issue by collecting data from reliable sources, enabling more comprehensive analyses to detect attack patterns, identify countries that are more susceptible to specific actors, and determine which types of organizations are most frequently targeted. These frameworks contribute to a more detailed understanding of the threat landscape and support the development of targeted mitigation strategies for specific actors.

Our approach offers a systematic way to categorize threat actors using publicly available data, while remaining adaptable to incorporate proprietary intelligence feeds from providers such as CrowdStrike or FireEye. Importantly, the relevance and impact of a

specific threat actor are contextual and victim-dependent: an actor predominantly targeting organizations in the United States or Canada may pose minimal risk to a company in Madagascar, but significant risk to a U.S.-based critical infrastructure operator. By integrating geographic, sectoral, and actor-specific prevalence data, our method provides dynamic, context-aware threat assessments that reflect both the actor's capabilities and the characteristics of the potential victim.

2.3. CVE relationship

Complementing the understanding of threat actors and their behaviours, vulnerability analysis provides a technical foundation to understand how attacks are executed and systems are compromised. CVEs represent a standardised framework for cataloguing publicly known security flaws that can affect organisations across all sectors [32]. When successfully exploited, these vulnerabilities enable threat actors to compromise systems, resulting in unauthorised data access, operational disruptions, and significant financial impact.

The association of vulnerabilities with affected systems is based on structured mapping processes that link CVEs to standardised repositories. Each vulnerability is assigned to the National Vulnerability Database (NVD), which supplements basic CVE information with impact assessments, references to vulnerable products via Common Platform Enumeration (CPE), and classification of the underlying weaknesses through Common Weakness Enumeration (CWE) [35]. This structured approach enables consistent vulnerability tracking and supports the integration of threat intelligence into organisational risk frameworks.

Recent advances in vulnerability relationship modelling have emphasised graph-based representations to capture complex interdependencies. Knowledge graph approaches [32, 37] enable the discovery of non-obvious connections between vulnerabilities, affected products, and exploitation patterns, supporting more accurate threat predictions. ML techniques, particularly graph neural networks [2, 38], have demonstrated substantial improvements in identifying vulnerable code patterns and predicting missing relationships within vulnerability databases. New research has extended these methods by incorporating Natural Language Processing and reasoning capabilities from LLMs to extract and integrate information from unstructured threat intelligence sources [16].

Despite these advances, vulnerability databases continue to face fundamental limitations. Significant processing delays result in incomplete metadata for newly disclosed vulnerabilities, while inconsistent weakness classifications reduce the reliability of automated analysis tools. Studies have documented substantial backlogs in vulnerability analysis and highlighted discrepancies in how CVEs are assessed and used in security research. Addressing these challenges requires continued development of automated enrichment techniques, cross-source validation mechanisms, and predictive models capable of operating effectively with incomplete or evolving vulnerability information.

3. Proposal

In this section, we present the proposed system, called DICYME, which focuses on automating cyber risk quantification and management in operational technology (OT) environments. The main objective of this system is to design and implement a system capable

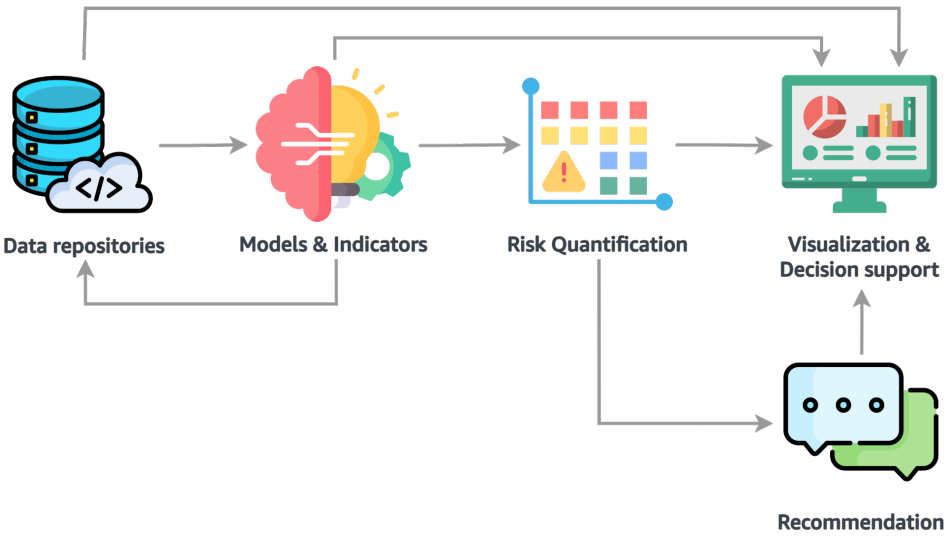


Fig. 1. High-level architecture of the DICYME complete system

of addressing the current challenges in modelling cyber risks within industrial infrastructure. The proposed solution relies on evidence-based analysis, integrating data from multiple sources to dynamically estimate the probability and impact of potential cyber incidents. To achieve it, the system incorporates a set of cyber risk indicators, which together with the data contribute to the simulation and quantification model. In addition, LLM to support explainability and enhance user understanding while providing recommendations in natural language. Finally, all components are brought together within a visualization platform that delivers a continuous, clear, and actionable view of cyber risk levels, designed to support informed decision-making in critical environments.

Figure 1 illustrates the architecture of the system, which is structured into five main parts. Automated data extraction gathers and preprocesses heterogeneous information from both external and internal sources, such as threat intelligence feeds and internal security telemetry. This raw evidence feeds the indicators block, where relevant cyber risk metrics are computed and normalised to reflect dynamic aspects such as visibility, reputation, or the threat actors landscape. These indicators are then aggregated in the cyber risk quantification module, which applies models to estimate the probability and impact of incidents. The outcomes, together with the underlying models and source data, are presented in the visualization and decision support layer, providing interpretable dashboards and analytical tools that allow stakeholders to explore and understand risk. Finally, the recommender component not only guides users in interpreting the visualized data and models, but also actively supports decision-making by proposing adjustments to the quantification process, suggesting alternative data inputs, and executing the risk models with multiple scenarios to provide actionable and feasible recommendations.

In addition, Figure 2 provides a more detailed view of the system, explicitly mapping all the elements that contribute to CRQ and the relationships among them. The figure distinguishes between the different data repositories, the models and indicators built on

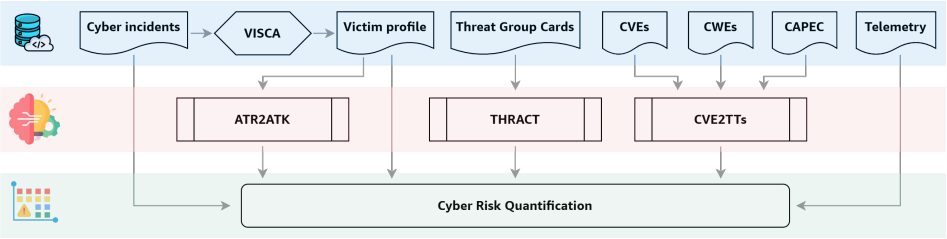


Fig. 2. Structural relationships within the Cyber Risk Quantification system

top of them, and the final CRQ process. This separation highlights how raw evidence is progressively transformed into structured knowledge and then into quantifiable indicators of risk, while also allowing certain datasets to be used directly in the quantification process when relevant, for instance to identify trends, contextual factors.

3.1. Automated data extraction

The automated data extraction module was designed to systematically collect and consolidate heterogeneous cyber risk information from multiple open sources. Specifically, it integrates reports of cyber incidents from six publicly available databases to provide a structured view of the threat landscape. To complement incident-level data, the module also incorporates victim profiling information, both with manual identification which is then empowered through Victim Insight System for Cyber Attacks (VISCA), a multi-agent LLM-based architecture that identified affected organizations and retrieves and processes its firmographic attributes.

In addition to incident and victim data, the module gathers technical Cyber Threat Intelligence (CTI) such as up-to-date CVEs with their associated vulnerability types, Tactics, Techniques and Procedures (TTPs) from the MITRE ATT&CK framework, as well as structured attack patterns from Common Attack Pattern Enumeration and Classification (CAPEC). Furthermore, it integrates information on threat actors from publicly available Electronic Transactions Development Agency (ETDA) Threat Group Cards [4], linking adversarial groups to their targeted victims and contextual attributes. It also incorporates internal telemetry data, capturing Operational Technology (OT)-specific information such as asset counts, exposed vulnerabilities, and network-level observations across Purdue layers. By consolidating organizational, incident-specific, adversarial, and technical dimensions, the module creates a unified knowledge base that supports the advanced modelling of cyber risks in subsequent stages.

Cyber incidents Cyber incident data were initially extracted from publicly accessible and free sources, including the TI Safe Incident Hub [34], Industrial Control System Security, Threats, Regulations, Incidents, Vulnerabilities provided by Experts (ICS STRIVE) [9], KonBriefing [12], Cyber Events Database of the CISSM, Hackmageddon [23] and EuRepoC. The extracted records were stored along with the extraction date and the source of the origin. Subsequently, the data underwent transformation and cleaning processes to achieve a more standardized format suitable for analysis. The transformation primarily

consists of restructuring the raw data into a tabular form, where each row represents a cyber incident and each column corresponds to an attribute of that incident. Data cleaning is performed to improve data quality by removing invalid entries or correcting formats, such as normalizing date representations. Additionally, cleaning involves harmonizing categorical values, for example, standardizing the country of the victim or the type of attack.

Finally, the sources were converted into a uniform tabular format to enable cross-source comparison of incidents, support exploratory data analysis across all repositories, facilitate incident deduplication, and allow for higher-level transformations.

Victim profile The Victim profile dataset was collected in the DICYME system to support the development of the Attractiveness concept (see Section 3.2), which delves with the possession of properties or behaviours in entities that may capture the focus of potential adversaries [7, 8]. The dataset includes uniquely identifiable entities that have been reported as victims of confirmed cyber incidents from the Cyber incidents dataset (see Section 3.1), manually selected and validated by a threat intelligence analyst. In addition, similar entities by country and industry category without known incidents were included to provide more context for a comparative analysis.

For each entity, the collected information can be organised into three groups that correspond to the main components of the Attractiveness model. The first group, basal attractiveness, refers to inherent static firmographic features, which includes the country, sector category, revenue, earnings, publicly traded status, number of employees, and profitability. The second group, online reputation, captures dynamic factors linked to human-generated content, social media interactions, and shared experiences related to the entity. It is computed through engagement (E), defined as the ratio of interaction (I) to reach (R) [3], where reach indicates the number of individuals who view or mention a publication and interaction represents those who actively respond to it. Engagement is assessed by distinguishing whether the content originates from the entity itself (assumed predominantly positive) or from external users, where sentiment analysis is performed to capture positive, neutral, or negative perceptions [7]. The third group, potential victimisation, represents another dynamic factor and includes the number of appearances of the entity name in dark web leaks and the number of visible devices connected to the Internet. These three groups are combined to produce a final structured dataset, containing a large amount of information, including features that are complex to extract, all referenced to each entity.

Victim Insight System for Cyber Attacks (VISCA) The VISCA model is an agent-based framework developed within the DICYME system to improve and extend the Victim profile dataset by automating and scaling the identification of victim entities and the fusion and normalization of heterogeneous data. This model takes the description of a cyber incident in natural language and collects firmographic data, such as its country, industry category, age, annual revenue, and number of employees. The proposed model is integrated into the system to generate a larger Victim profile dataset and also to obtain firmographic data that users can use in the CRQ model. The full workflow of this model is illustrated in Figure 3.

- Retrieve basic information about the victim: using a cyber incident as a data source, a LLM is capable of extracting relevant information from the incident description such

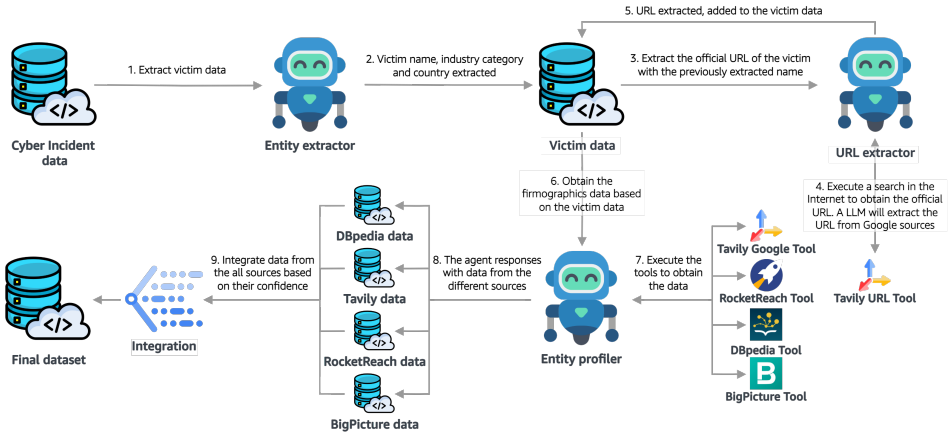


Fig. 3. Multi-agent architecture proposed for extracting victim information and completing missing firmographic data

as the name of the entity victim, the country and the industry. Additionally, the model queries various data sources from different websites on the Internet to retrieve the official website Uniform Resource Locator (URL). This basic information is used in subsequent processes to obtain or filter the firmographic data.

– Dig up firmographic information: with the basic information of the victim extracted, the model proceeds to perform queries across various external data sources to retrieve additional details. The information to be collected consists of eight distinct fields encompassing different data types: name of the victim, industry category, founded year, number of employees, country, revenue, publicly traded status, and North American Industry Classification System (NAICS) codes. The model requests information about the victim entity from four main sources: Google, DBpedia, BigPicture, and RocketReach. DBpedia and BigPicture are open-access sources available to any user seeking information on companies. In contrast, RocketReach is a paid platform that provides data with a higher level of reliability, as it is a more private and commercially maintained product. Google information is retrieved through a tool called Tavily, a platform that enables agent-based AI models to connect to the web. Tavily collects data from multiple Google sources while returning a filtered subset of relevant results and selecting only a reduced number of sources.
– Combine the extracted data: once the information has been extracted from the data sources, the final step performed by the model is the aggregation of the data to generate a consolidated set that captures the most complete information possible. To achieve this, a confidence score was assigned to each dataset retrieved from the sources based on the number of fields successfully extracted. The source with the highest confidence score was selected. In cases where certain firmographic fields are missing, the model attempts to replace them by merging data from other sources where their confidence scores exceed a defined threshold.

Once the final agent gathers all datasets from various sources, the goal is to consolidate them into a single dataset that maximizes data completeness. In particular, the final dataset

should include the victim's name, category, NAICS code, founded year, size, country, revenue, and public trading status. In addition to these fields, a confidence score was assigned to each source, represented as a numerical value between 0 and 1. This score reflects the reliability of the extracted information, completeness of the retrieved data, and number of relevant results obtained during the search process, rather than data quality. A value close to 0 indicates highly unreliable information, whereas a value of 1 indicates high reliability.

The value of the estimation is computed as shown in Equation 1 for each source s.

$$c_fields_s = \frac{m_s}{f} \qquad c_responses_s = \begin{cases} \frac{1}{n_s} & \text{if } n_s > 0 \\ 0 & \text{if } n_s = 0 \end{cases}$$
$$conf_s = \frac{1}{r_s} \cdot \Big[\alpha \cdot c_fields_s + (1-\alpha) \cdot c_responses_s\Big] \tag{1}$$

where the confidence score, $conf_s$, quantifies the reliability of a specific source. This confidence depends entirely on the completeness of the extracted data, the number of responses obtained, and the efficiency of the data retrieval process. The first component c_fields_s measures the number of fields obtained m_s relative to the total number of fields to be extracted f, that is, its proportion. The second component, $c_responses_s$ serves as a confidence penalty when a source returns a large number of potential results, n_s, indicating uncertainty regarding the most accurate outcome; the confidence value is correspondingly reduced. The overall confidence score, $conf_s$, integrates these two components through a weighted average controlled by a tunable parameter, α, and is adjusted according to the type of source selected. For instance, in the case of $s =$ RocketReach a higher value is used to prevent the confidence from being penalized, as this source typically returns a large number of responses. The number of attempts, r_s, required to obtain a valid result increases with each failed query and always starts at one.

Queries are executed using a list of potential victims generated by the *Entity extractor* agent. This list contains extensive terminology referring to the victim, including variations such as corporations, brands, or subsidiaries, ordered from the most likely to the least likely, based on the LLM. Each agent sequentially queries the names until relevant information is obtained. Every failed attempt increments r, which reduces the final confidence score to reflect the inefficiency of the search. There is always an initial attempt; therefore, r_s can never be 0.

When the confidence $conf_s$ is computed for each source s, the dataset with the highest overall confidence is selected. If one or more fields are missing in the selected dataset, the system evaluates those specific fields across all sources with confidence scores higher than a threshold of 0.5 and aggregates the data.

Electronic Transactions Development Agency (ETDA) Threat Group Cards The ETDA is a Thai public organization that focuses on promoting secure and efficient digital transactions. Among its key initiatives, ETDA develops detailed Threat Group Cards to support CTI efforts. These profiles integrate data from leading global sources, including the Malware Information Sharing Platform (MISP) Threat Actors Galaxy, MITRE ATT&CK Framework, Malpedia, Open Threat Exchange (OTX), and ETDA's own CTI

archive and open-source research, offering a comprehensive view of threat actors targeting digital infrastructure.

Data acquisition was conducted through direct downloads of individual Threat Group Cards and a consolidated JavaScript Object Notation (JSON) file, both available on the official ETDA website. The consolidated file aggregates all documented actors and is designed for seamless integration with platforms such as MISP. Taken together, these two resources provide a rich set of attributes for each threat group, including their countries of operation, targeted industries, stated or inferred motivations, attributed campaigns and their timelines, as well as the tools and techniques employed.

Another important data source integrated into the system consists of cybersecurity vulnerabilities and related attack patterns, which are then used and completed by the CVE2TTs model (see Section 3.2). The process begins by collecting vulnerability data from the NVD, which includes CVE identifiers, descriptions, and associated CWE mappings. These CWEs are linked to CAPEC entries by MITRE, which in turn connect to MITRE ATT&CK techniques. While not every CVE has a direct mapping along this full chain, available mappings are used when possible, and gaps, particularly in the Industrial Control Systems (ICS) matrix, are predicted by the CVE2TTs model.

Internal telemetry data In addition to external sources, the system also incorporates internal data. However, this component is more limited, as obtaining this data from industrial organizations requires explicit authorization and faces significant operational and confidentiality constraints. Nevertheless, the system integrates internal telemetry whenever available, primarily consisting of anonymized samples of Intrusion Detection Systems (IDSs) data provided by DeNexus. It also includes detailed visibility into the number of connected assets, their vulnerabilities, and their distribution according to the Purdue model. This information is particularly valuable for risk quantification, as it enables the identification of exposed assets and weaknesses directly linked to the operational infrastructure. From this telemetry, the system extracts CVEs, which are then related to the CVE2TTs model (see Section 3.2). When such data are not accessible, users can alternatively upload a list of CVEs associated with the entity, ensuring that the quantification process remains consistent and applicable. By combining OT-specific IDS discovery capabilities with flexible input options, the platform strengthens both the accuracy and relevance of cyber risk assessment while maintaining confidentiality.

3.2. Indicators

To quantify cyber risk effectively, a set of indicators has been defined to capture distinct patterns and characteristics of the data, each serving a specific purpose depending on the indicator. Specially, in the DICYME project there are three main indicators: ATR2ATK, THRACT, and CVE2TTs.

ATR2ATK ATR2ATK or Attractiveness is an indicator for forecasting cyber incidents by assessing the proneness of an entity to them. It is decomposed into three main branches: basal attractiveness, online reputation, and potential victimisation.

Basal attractiveness focuses on inherent static characteristics, often referred to as firmographic data, such as location, operational criticality, data sensitivity, and size. These

attributes make certain entities more appealing targets for adversaries because of their intrinsic nature. It is modelled with association rule mining, specifically the FP-Growth algorithm, combined with a Decision Tree classifier that uses its output (amount of satisfied rules and aggregated support, lift and confidence per incident) [8, 7].

Online Reputation measures the presence of the entity in social networks and social media, as it may be more attractive to an adversary based on what users, customers, or employees write, communicate, and share anywhere on the Internet based on their perceptions and experience at any moment of their relationship, direct or indirect, with the entity [20, 21, 33]. It is a dynamic component and time-dependent, as it represents a specific moment, influenced by previous periods. Its computation relies on engagement (E), defined as the ratio of interaction (I) and reach (R) [3]. Reach indicates the number of individuals who view or mention a publication, whereas interaction (I) represents those who actively respond to it. In this context, engagement is assessed by considering whether the content originates from the entity itself through its official sources, where sentiment is assumed to be predominantly positive, or from other users, where additional sentiment analysis is taken into account since perceptions can be positive, neutral, or negative.

Lastly, potential victimisation, which is also a dynamic component, has to be with the visibility and technical knowledge that the adversary can have about the entity. Its rationale lies in the idea that organisations become more likely targets when they gain visibility in underground forums, leak sites, or other malicious communities, and when technical indicators reveal that their infrastructure may be easy to compromise. To capture this, two variables are monitored: the number of mentions in dark web leaks and the number of publicly visible assets connected to the Internet. These are combined through a buffer method that assigns a value between 0 and 2: 0 if neither variable is present, 1 if only one of them is, and 2 if both are detected. This provides a simple yet effective measure of potential victimisation.

THRACT This indicator is a composite metric designed to provide an overview of threat actor activity targeting a specific country and industry. Based on the data extracted from the ETDA, three partial metrics are derived to reflect their activity, capabilities, and objectives with respect to a specific victim. Activity is captured through the time elapsed since the actor was last observed. As the ETDA database is updated on a rolling basis, this value may change over time, reflecting updates in observed behavior. Actor capabilities are represented based on expert annotations of operational capacity. Victim-related features include country and industry, while actor motivation is also considered. Because the metric depends on the country and industry of the potential victim, it is dynamic rather than a fixed score for each actor in any situation, providing a context-sensitive measure of potential cyber risk.

CVE2TTs Finally, the CVE2TTs indicator corresponds to a ML model that maps CVE entries to specific Techniques (and, by extension, Tactics) within the MITRE ATT&CK framework. This mapping is performed by leveraging the textual description of each CVE together with additional attributes such as vulnerability type, CWE, CAPEC, and attack vector [29]. This concept is very important in the cybersecurity sector because it helps understand the practical exploitation of vulnerabilities in a structured manner, linking specific vulnerabilities to known adversary behaviors TTPs. This model is crucial for risk

management, CTI, and incident response, as it bridges the gap between vulnerabilities and the manner in which attackers might exploit them.

3.3. Cyber Risk Quantification (CRQ) model

In the context of cyber risk management, the Cyber Risk Quantification model introduced in this proposal enables a simulation-based tool that quantifies cyber risk in financial terms, supporting informed decision-making for organizations. Using stochastic simulations, the model estimates the Annual Expected Loss for a given organization by combining the frequency of successful attacks and their financial magnitude or impact, as shown in Equation 2.

$$\text{Cyber risk} = \text{Loss Event Frequency (LEF)} \times \text{Loss Magnitude (LM)} \tag{2}$$

The model relies on a state-of-the-art risk tree that separates probability and impact into two main branches. The first component, the Loss Event Frequency (LEF), represents the expected number of loss events within a given period, typically one year. In this context, loss events correspond to materialized incidents that cause harm to the organization. The LEF is determined as the product of the Threat Event Frequency (TEF), which estimates the number of attempted attacks that the organization is likely to face, and the Susceptibility, which reflects the probability that those attempts will successfully translate into incidents. This relationship can be expressed as shown in Equation 3.

$$\text{Loss Event Frequency (LEF)} = \text{Threat Event Frequency (TEF)} \times \text{Susceptibility} \tag{3}$$

This formulation integrates both the external pressure from the threat landscape (captured by the TEF), and the internal defensive posture of the organization (captured by Susceptibility). The TEF takes into account a baseline number of incidents based on private cyber intelligence knowledge, the ATR2ATK indicator, and the annualized rate of incidents for the country and industry category of the entity, sourced from the Cyber incident dataset. Susceptibility is derived from the vulnerabilities obtained through internal telemetry or from a list of vulnerabilities uploaded by the user, in combination with the CVE2TTs model, the Threat Actor Index, which captures the activity and focus of the 245 actors in the database over a three-year period, and an adjustable security profile index. The latter characterizes the defensive posture of the organization by incorporating features such as organizational size, number of unpatched vulnerabilities, and exposure of devices visible from the internet.

The second component of the risk tree is the Loss Magnitude (LM), which quantifies the financial impact of a successful attack as the combination of primary and secondary losses, as stated in Equation 4. The identification of feasible primary and secondary losses depends on the attack technique k, which is constrained by the vulnerabilities observed or uploaded by the user, linked to the CVE2TTs mapping. In practice, just the achievable techniques from the Impact MITRE ATT&CK tactic, given the vulnerability landscape of the entity under analysis, are considered in the simulation. This ensures that both primary and secondary losses are not only probabilistically derived, but also grounded in the

technical conditions that define the exposure of the organization. Expert financial knowledge is further incorporated to parametrize and calibrate the estimation of losses to reflect realistic economic consequences for the entity.

$$\text{Loss Magnitude (LM)}_k = \text{Primary Loss}_k + \sum \text{Secondary Losses}_k \tag{4}$$

Primary losses represent the direct effect of an incident and include categories such as business interruption, equipment damage, extortion, or human impact. In each simulation run, only one primary loss type is selected, corresponding to the most plausible impact mechanism for the attack technique under consideration. By contrast, secondary losses capture indirect consequences, such as forensic investigation costs, reputational damage, or regulatory penalties. Unlike primary losses, several secondary losses can occur simultaneously, and their combined effect is represented by summing the estimated financial amounts for each of them.

3.4. Visualization and decision support

The visualization and decision support component is implemented as an interactive application that allows users to explore datasets and execute risk models through two main options: either using the entities defined in the Victim profile dataset or by providing their own model inputs. The platform provides a wide range of interactive visualizations, including bar charts, histograms, stacked and unstacked time series, heatmaps, and world maps. Figure 4 shows some of this techniques used inside the CVE2TTs module, reflecting the relations made between CVEs and MITRE ATT&CK Tactics and Techniques by vulnerability type and by Common Vulnerability Scoring System (CVSS), as well as the number of vulnerabilities and predicted techniques per year. Users can filter the data by relevant variables such as dates, incident databases, CVE types, and other context-specific metrics. This interactivity enables stakeholders to examine patterns in risk indicators, compare scenarios, and gain a deeper understanding of the underlying data and model outputs.

In addition, the system generates a downloadable report that compiles all the data, the computed indicators, and details on the steps of the quantification process. This report can be shared with other stakeholders, including expert panels, managers, or decision-makers, ensuring that risk analyses can be reviewed, discussed, and extended beyond the interactive platform. By combining interactive exploration with exportable documentation, the system supports both immediate, hands-on decision-making and broader strategic planning across multiple levels of an organization.

3.5. Recommender model

In addition, the system incorporates a set of recommender models that complement its core functionality. These models are designed to identify patterns within the data and indicators, extract meaningful features, and provide tailored recommendations. Although is not essential for the system baseline operation, this component enhances its analytical capabilities and offers added value by supporting more informed decision-making. The models employed to perform these tasks were fine-tuned through prompt engineering [11] and Retrieval Augmented Generation (RAG) techniques [15] to produce responses



Fig. 4. Example of visualization techniques used in the CVE2TTs module of the DICYME system. The top heatmap displays the distribution of CVEs identified for each ICS tactic by vulnerability type. The middle bar chart illustrates the temporal evolution of security data, showing the total annual count of CVEs (blue) alongside the volume of predicted Enterprise-specific techniques (orange) and ICS-specific techniques (yellow) associated with those vulnerabilities. Finally, the bottom pyramid chart presents the average CVSS scores per tactic, calculated from the CVEs associated with each tactic according to the CVE2TTs model's predictions. The chart is bifurcated by domain: the Enterprise matrix tactics are positioned on the left, while ICS matrix tactics are on the right. Within this chart, blue bars indicate average scores according to the CVSS v2 standard, and orange bars represent the CVSS v3 scoring

that are as accurate as possible. By adapting the prompts for each system component with their corresponding data, a specialized prompt is generated for that specific component to ensure a more accurate and context-aware response.

In some cases, the data used in the prompt processed by the model may lead to undesired responses. This occurs because the system generates large volumes of data that may pose challenges for the model when generating an analysis, since these models are subject to token limitations. To address this issue, query decomposition [36] is applied in situations where sections contain extensive data. The queries were divided, allowing multiple interactions with the model to provide more specific explanations. This approach made it possible to explain similar datasets without resorting to generalization. Finally,

the responses to these queries were consolidated by the LLM using a prompt designed to summarize all partial outputs and deliver a comprehensive final explanation.

In addition to the models that provide data explainability, decision support capabilities, and parameter adjustment, the system also incorporates a more sophisticated model implemented as a chatbot assistant. The assistant enhances user interactivity by enabling the execution of the same tasks performed by other models with the added flexibility of allowing users to request specific adjustments or obtain tailored explanations through natural language queries. The aim of enhancing user interaction, explainability, and adjustments is achieved through the assistant, which is integrated into one of the most important and critical components of the system: the CRQ simulation. Through an intuitive interface, users can submit questions related to the CRQ, which the AI assistant answers one at a time in a conversational manner. The model has a limited memory capacity, allowing it to retain awareness of previous prompts and generate responses within a specific window. Thanks to this memory, the assistant can access prior questions and answers, compare its outputs, retrieve the results of previously executed simulations, and revisit earlier recommendations. This enables the contrast of past outcomes with newly generated ones, thereby supporting more consistent analyses and more informed decision-making.

The architecture implemented for the GenAI models is inspired by the approach proposed by DeepSeek-V3 [14], which is based on a Mixture of Experts (MoE) model. This architecture uses a set of specialized experts, each responsible for handling specific tasks depending on the context. The assistant is capable of handling three distinct tasks, each managed by a dedicated agent within the architecture: answering general questions related to CRQ process, providing recommendations on parameter adjustments, and rerunning the simulation to generate a comparison with the previous simulation results.

4. Experiments

The experiments presented in this study evaluated two key aspects of the proposed system. First, we assess the data extraction and integration capabilities of the multi-agent architecture, focusing on the completeness and quality of the firmographic information collected from multiple cyber-incident databases.

Second, we evaluated the entire risk quantification workflow, from the computation of different intermediate indicators and metrics present in the system to the computation of risks, probabilities, frequencies, and other risk parameters. For this purpose, a simulation will be conducted using the model with a case study and real data from a possible victim entity or institution.

The model selected for the multi-agent architecture, which extracts the victim data, is the Meta LLaMa 4 Scout [18], with a total of 17B active parameters and the advantage of being open-weight, which allows us to maximize utility. This model was chosen over other available alternatives because, in addition to its ease of deployment, it supports the execution of agents and tools through libraries employed in this project.

4.1. Data gathering

Within the DICYME application, a wide range of datasets are employed across different tabs to present information using tables, charts, and other visualization components. As

previously discussed, the data were extracted from various sources and organized into three main datasets used in different modules of the application: Cyber incidents, Victim profile, and IDSs.

This experiment evaluates the firmographic data completion of the VISCA module by comparing the dataset it generates against the analyst-annotated Victim profile dataset. After collecting information from victim entities affected by cyber incidents, the data are stored in a dataset to be used within the system. The objective of this experiment was to compare the automatically extracted dataset with a similar dataset in which the victim entity data were manually collected by a human annotator.

Conversely, the victim dataset used to evaluate the extracted dataset was a fully handcrafted dataset compiled by a human annotator. It contains 675 instances of distinct victim entities and includes a substantial amount of information related to each company, specifically firmographic data. In particular, the dataset contains 9 variables, including the incident category, entity, industry category, country, earnings, employees, revenue, profitable and publicly traded. This dataset is used within the system to compute various indicators and represent key information about each victim entity, which explains the large number of variables included, as they serve as inputs for these indicators.

In contrast, the dataset under evaluation, obtained after the extraction and aggregation of firmographic data, was automatically constructed by a multi-agent system capable of executing a workflow starting from a cyber incident. Once the agent extracts data from multiple data sources, it integrates them into a final output to produce a consolidated result. The final output contains eight firmographic fields, including: entity, category, country, revenue, founded year, employees, publicly traded, and NAICS code. This dataset contains 1,658 observations of distinct victim entities processed from cyber incidents collected from the ICS STRIVE, EuRepoC, and TI Safe databases. In all the processed cyber incidents, at least one victim entity was explicitly identified in the incident description, which enabled the extraction of the corresponding firmographic data.

Table 1 presents a comparison between the two datasets used to model a profile for a victim entity. It can be observed that the VISCA-extracted dataset contains a significantly larger number of identified victim entities. This is one of the most notable characteristics of the model, as its autonomous workflow can continue processing incidents until all available incidents are processed, while also achieving a much higher processing speed than manual annotation can. This difference in gathering speed is evident in Table 1, which includes the range of extraction dates for the data collected. While the Victim profile dataset required an entire year to compile all the victim entities it contained, the VISCA model was able to extract more than double the number of victim entity records in just five months of uninterrupted execution.

However, one limitation of this dataset is the smaller number of variables compared to the Victim profile dataset. This difference is due to the specific original goal of the VISCA model: to generate victim-related data for cyber risk quantification using the CRQ framework. In contrast, the Victim profile dataset was developed to support multiple components of the system, including the computation of indicators, metric computation, and structured information representation. Nevertheless, the VISCA model can be extended to collect additional firmographic attributes following the same automated process, allowing the resulting dataset to become more comprehensive and usable across a wider range of system components.

Table 1. Comparison between the Victim profile dataset and the VISCA-extracted dataset

Feature / Metric	Victim profile dataset	VISCA-generated dataset
Unique entities	675	1,658
Number of fields	9	8
Sources used	Manual collection	ICS STRIVE, EuRepoC, TI Safe
Extraction method	Manual annotation by an analyst	Multi-agent automated workflow
Extraction dates	December 2023 - December 2024	May 2025 - September 2025

4.2. Cyber Risk Quantification (CRQ)

To evaluate the effectiveness of the CRQ procedure, we conducted a series of experiments that simulate real-world scenarios using actual data from multiple companies. These experiments were designed with two main objectives: first, to demonstrate the internal performance of the CRQ process, including the types of input data and datasets that can be leveraged within this framework, and second, to validate the output risk metrics generated by the model, which serve as a foundation for informed decision-making by organizations.

Executing the CRQ workflow of the system requires the input of various firmographic attributes that characterize a company. On the one hand, firmographic data from real companies can be loaded from the Victim profile dataset, specifically including country, industry category, revenue, earnings, number of employees, publicly traded status, profitability, online reputation, publicly visible devices, and critical information leaks. However, these fields can be populated directly with the data of the company provided by the user operating the system. In addition, other simulation parameters are configurable, such as the random seed, number of simulations, and insurance parameters. Finally, a list of unpatched CVEs present in the infrastructure of the organization, which may be exploited, must be provided, as they serve as a basis for estimating the corresponding risk measures, as stated in Section 3.3.

For the experiments, a real-world case was selected, corresponding to an actual company extracted from our dataset. The input values for the model are the previously described firmographic attributes of the company, all of which are available in the dataset. Specifically, the University of Michigan was selected for this experiment. This public institution was selected because it is recognized as one of the leading universities worldwide and is located in the United States, a country that consistently reports the highest concentration of cyberattacks. This combination of global academic relevance and geographical context makes the institution particularly susceptible to being frequently targeted by threat actors.

This public institution, located in the United States and dedicated to education, reported a revenue of €9.1 billion, employed a total of 34,600 staff members, and had three publicly visible devices and one critical information leak. Using these publicly available data from the university, CRQ simulations were executed and complemented with additional configuration parameters, including the number of runs, random seed, and insurance settings. The CRQ procedure first computes a series of indicators that are subsequently used to derive the output of the model parameters. The values of all the indicators are listed in Table 2. Among them, the attractiveness score is 80%, a significantly high value, primarily driven by the institution's location in the United States and its educational industry category. Another key indicator is the Threat Actor Index, which reached a value

Table 2. Summary of the values obtained from the CRQ simulation, including both the computed indicators and the model output parameters

Indicators	Baseline	85 events/year
	Incident rate	50%
	Attractiveness	80%
	Threat actor index	58%
	Exposure	41.61
	Security profile	67%
CRQ Outputs	Threat Event Frequency (TEF)	34 events/year
	Susceptibility	2%
	Loss Event Frequency (LEF)	15 events/year
	Primary loss	€49,512
	Secondary loss	€20,786,315
	Loss Magnitude (LM)	€20,835,828/event
	Cyber risk	€14,168,363/year

of 58%, reflecting the intensity and capabilities of threat actors over a three-year period and corresponding to a moderate risk level.

Based on the computed values derived from the publicly available data of the university and complemented with additional simulation parameters, such as the number of runs, a random seed to ensure reproducibility, and cyber insurance financial values, the model produces simulation outputs presented in both graphical and numerical forms. Figure 5 presents the results of the simulated risk through two charts: the first describes the distribution of loss events across all simulated years (each simulation means a year), while the second illustrates the Loss Exceedance Curve (LEC), which illustrates the probability (Y-axis) that cyber risk losses will be equal to or greater than a given amount (X-axis).

In the first chart, which represents the distribution of annual loss events, the results show a left-biased pattern. The majority of simulated years (574 out of 1,000) resulted in no materialized incidents, while 306 years included a single loss event. Multiple events within the same year were comparatively rare: 83 simulations produced two events, 24 produced three, 9 produced four, and only 4 reached five events. This highlights that, although the occurrence of cyber loss events is possible, their frequency per year remains predominantly low, consistent with the heavy-tailed nature of cyber risk distributions.

The second chart, which illustrates the distribution of financial losses (LEC), reveals two distinct components. At the lower end, there is a steep initial drop, corresponding to rare but high-impact scenarios—so-called black swan events. Beyond this abrupt decline, the curve follows an almost linear descent, decreasing steadily from a 47% probability of incurring losses equal to or greater than €8.9×10^5 down to 0% probability at €9.0×10^7. Unlike sharply convex loss distributions, this profile suggests a smoother, more gradual reduction in probability across a wide range of loss magnitudes, balancing rare catastrophic scenarios with a persistent probability of moderate to high losses.

Finally, the CRQ model also provides single averaged values, complementing the previously described distributions and probabilities, to facilitate a direct quantification of losses and frequencies. Based on the indicators and other adjustable system parameters, the CRQ framework estimates a LEF of 0.68 events per year and a LM of €20,835,828 per event. By combining these two measures, the system produces the annualized risk

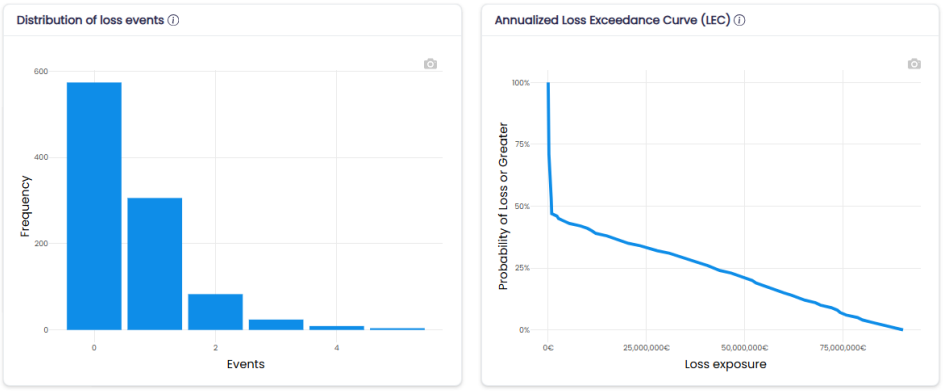


Fig. 5. Distribution of Simulated Loss Events and Corresponding Loss Exceedance Curve in the CRQ simulation

value for the organization, which in the case of the University of Michigan is estimated at €14,168,363 per year.

5. Lessons learned

The analysis of the experimental results provides several insights into the strengths and limitations of this proposed system. The experiments conducted on the CRQ model revealed both notable advantages and certain weaknesses of this component. One of these strengths is the flexibility to simulate realistic scenarios using firmographic data from real companies, while also allowing the introduction of fictitious parameters that emulate organizations not present in the dataset. This strength is further enhanced by the ability to leverage firmographic information obtained from the Victim profile and VISCA datasets, which enrich the inputs of the model and provide a stronger foundation for risk quantification. This capability greatly expands the range of situations in which the model can be applied, as it not only quantifies risks for entities included in the repositories but also provides valuable estimations for hypothetical cases, making it useful in decision-making and what-if analyses. Additionally, the results generated by the model are highly descriptive and presented through graphical visualizations that allow users to easily interpret the represented case. By combining frequencies, probabilities, distributions, loss estimations, and a final risk value, the model provides a comprehensive and visually intuitive quantification of the different possible risk situations.

However, a key limitation of the current evaluation is the absence of formal validation of the accuracy of the quantification and recommendation strategies. A potential approach to address this limitation is post-mitigation validation, leveraging both publicly available but also internal data from organizations that have experienced cyberattacks. This would enable comparisons between the predicted frequencies and loss estimations by the model against the actual incidents and financial losses reported over a given period, providing an empirical basis for assessing the reliability of the CRQ outputs. In the best-case scenario, although it may not be scalable, organizations could grant controlled access to private

datasets containing historical records of cyberattacks, including incidents that occurred before and after the implementation of specific defensive measures. This would allow the system to assess whether the predicted risk metrics accurately reflect the real-time impact of mitigation strategies.

The experiments also revealed that many records in the evaluated datasets contain missing fields, either because the information could not be found or because it simply does not exist. When key variables are absent, the model cannot perform optimal estimations. This issue was evident, for example, in the Victim profile dataset, where the used data sources almost never provide the earnings field. The absence of inputs may lead to incorrect probability distributions or models, having to consider whether it can be used or it is better to exclude them from the analysis.

Finally, the experiments conducted on the data extraction model demonstrated the speed and efficiency of aggregating victim entity information in a concrete case of cyber-incident processing. This capability highlights the potential of the system to automate time-consuming tasks, such as data labelling, enabling the generation of a fully usable dataset for the system. In addition to automating the process, the model operates significantly faster than manual human processing, allowing the creation of large-scale datasets that would otherwise require substantial time by humans. These enriched datasets can then be leveraged by other components of the framework, such as the CRQ model, thereby enhancing their performance by incorporating more relevant and updated firmographic information about organizations. However, an important limitation of the current approach is that the extracted data are not validated; the system only checks if the retrieved fields contain a value, but not whether the value is factually correct. Although the workflow relies on trusted and updated data sources, it also incorporates information retrieved from open Internet searches, which introduces the risk of inaccurate or inconsistent data due to the large number of possible sources.

6. Conclusion

This study introduced DICYME, a complex system that contains multiple datasets covering different cyber-related domains, information representation and processing capabilities, and various techniques designed to identify patterns and characteristics within the data. Throughout this study, the methods and techniques used to implement the complex workflow of the system are presented, beginning with the acquisition of diverse datasets that are subsequently employed to compute a range of indicators for quantifying cyber risk across all instances. Subsequently, the CRQ model was proposed, enabling a large number of simulations aimed at quantifying the previously collected data and their computed indicators, and finally producing a final risk value along with other relevant metrics.

In the experiments section, the results demonstrated that both useful components of the system provide significant capabilities: a comparison of the Victim profile dataset and the CRQ model. The VISCA model exhibited high performance in the data extraction task, achieving efficient execution times and demonstrating the potential to fully automate the annotation task performed by a human. However, an important limitation of the current approach is that during data extraction, the system does not explicitly obtain the most accurate value among all possible sources. This limitation may lead to subsequent components and models of the system with inaccurate or inconsistent data, which can

propagate errors into subsequent stages of the system. As future work, it would be important to implement a mechanism that not only maximizes the number of extracted attributes but also ensures the selection of the most reliable value for each field. Additionally, extending the set of collected variables would enable a more detailed and representative modelling of the victim profiles.

Similarly, the CRQ model proved capable of generating a rich set of metrics and quantitative outputs, offering valuable insights that can directly support post-mitigation actions by helping organizations prioritize defensive strategies. Nevertheless, there is currently no validation to confirm whether the estimations of the model accurately reflect real-world outcomes. This limitation is critical, as both the overestimation of risks and the underestimation of losses or attack frequencies could lead organizations to make incorrect decisions. Future work should include rigorous validation of the CRQ outputs, either by leveraging public datasets from organizations that have experienced cyberattacks or through partnerships with private companies to provide access to historical incident data. Such validation would allow for a systematic comparison between the predicted annual losses and event frequencies by the model and actual observed outcomes.

Acknowledgments. This work has been funded by the Spanish MICIU under the CPP program in the DICYME project (Ref: CPP2021-009025), partially funded by the XMIDAS project (PID2021-122640OB-I00), and supported by DeNexus Inc.

References

1. Center for International and Security Studies at Maryland: Cyber Events Database (2024), https://cissm.umd.edu/cyber-events-database
2. Chu, Z., Wan, Y., Li, Q., Wu, Y., Zhang, H., Sui, Y., Xu, G., Jin, H.: Graph neural networks for vulnerability detection: A counterfactual explanation. In: Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 389–401 (2024)
3. Cioppi, M., Curina, I., Forlani, F., Pencarelli, T.: Online presence, visibility and reputation: a systematic literature review in management studies. Journal of Research in Interactive Marketing 13(4), 547–577 (2019)
4. Electronic Transactions Development Agency: Threat Group Cards: A Threat Actor Encyclopedia (2025), https://apt.etda.or.th/cgi-bin/aptgroups.cgi
5. EuRepoC project: The European Repository of Cyber Incidents (2025), https://eurepoc.eu/
6. Falowo, O.I., Popoola, S., Riep, J., Adewopo, V.A., Koch, J.: Threat actors' tenacity to disrupt: Examination of major cybersecurity incidents. IEEE access 10, 134038–134051 (2022)
7. García-Ochoa, J., Fernández-Isabel, A., Contreras, C., Fernández, R.R., de Diego, I.M., Beltrán, M.: Defining the attractiveness concept for cyber incidents forecasting. Computer Science and Information Systems (ComSIS) (2025)
8. García-Ochoa, J., Fernández-Isabel, A., Martín de Diego, I., Contreras, C., R. Ravines, R., López, O.: Defining the basal attractiveness concept for cybercriminals. In: Proceedings of the 10th Jornadas Nacionales de Investigación en Ciberseguridad. pp. 513–517. Universidad de Zaragoza, Zaragoza, Spain (2025)
9. Industrial Safety and Security Source and Waterfall Security Solutions: ICSSTRIVE (2025), https://icsstrive.com/
10. Kamruzzaman, M., Bhuyan, M.K., Hasan, R., Farabi, S.F., Nilima, S.I., Hossain, M.A.: Exploring the landscape: A systematic review of artificial intelligence techniques in cybersecurity. In: 2024 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI). pp. 01–06. IEEE (2024)

11. Knoth, N., Tolzin, A., Janson, A., Leimeister, J.M.: Ai literacy and its implications for prompt engineering strategies. Computers and Education: Artificial Intelligence 6, 100225 (2024)
12. KonBriefing: Cyberattacks, Hacker attacks, Ransomware attacks (2025), https://konbriefing.com/en-topics/cyberattacks.html
13. Landauer, M., Skopik, F., Stojanović, B., Flatscher, A., Ullrich, T.: A review of time-series analysis for cyber security analytics: from intrusion detection to attack prediction. International Journal of Information Security 24(1), 3 (2025)
14. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024)
15. Liu, F., Kang, Z., Han, X.: Optimizing rag techniques for automotive industry pdf chatbots: A case study with locally deployed ollama modelsoptimizing rag techniques based on locally deployed ollama modelsa case study with locally deployed ollama models. In: Proceedings of the 2024 3rd International Conference on Artificial Intelligence and Intelligent Information Processing. pp. 152–159 (2024)
16. Liu, R., Xie, Y., Dang, Z., Hao, J., Quan, X., Xiao, Y., Peng, C.: Dynamic vulnerability knowledge graph construction via multi-source data fusion and large language model reasoning. Electronics 14(12), 2334 (2025)
17. Mallick, M.A.I., Nath, R.: Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments. World Scientific News 190(1), 1–69 (2024)
18. Meta: Introduction to the LLaMA 4 Models (2025), https://www.llama.com/docs/model-cards-and-prompt-formats/llama4/
19. Mızrak, F.: Integrating cybersecurity risk management into strategic management: a comprehensive literature review. Research Journal of Business and Management 10(3), 98–108 (2023)
20. Okutan, A., Werner, G., Yang, S.J., McConky, K.: Forecasting cyberattacks with incomplete, imbalanced, and insignificant data. Cybersecurity 1, 1–16 (2018)
21. Okutan, A., Yang, S.J., McConky, K., Werner, G.: Capture: cyberattack forecasting using non-stationary features with time lags. In: 2019 IEEE Conference on Communications and Network Security (CNS). pp. 205–213. IEEE (2019)
22. Paolo Passeri: 2024 Cyber Attacks Statistics (03 2024), https://www.hackmageddon.com/2024/03/26/2024-cyber-attacks-statistics/
23. Paolo Passeri: Hackmageddon (2025), https://www.hackmageddon.com/
24. Paté-Cornell, M.E., Kuypers, M., Smith, M., Keller, P.: Cyber risk management for critical infrastructure: a risk analysis model and three case studies. Risk Analysis 38(2), 226–241 (2018)
25. Phillips, S.C., Taylor, S., Boniface, M., Modafferi, S., Surridge, M.: Automated knowledge-based cybersecurity risk assessment of cyber-physical systems. IEEE Access 12, 82482–82505 (2024)
26. Pseftelis, T., Chondrokoukis, G.: Understanding cyber incident dynamics in the european union: A study of actor types and sector vulnerabilities. Preprints.org (04 2025)
27. Qin, X., Jiang, F., Cen, M., Doss, R.: Hybrid cyber defense strategies using honey-x: A survey. Computer Networks 230, 109776 (2023)
28. Qudus, L.: Advancing cybersecurity: strategies for mitigating threats in evolving digital and iot ecosystems. Int Res J Mod Eng Technol Sci 7(1), 3185 (2025)
29. Rajesh, P., Alam, M., Tahernezhadi, M., Monika, A., Chanakya, G.: Analysis of cyber threat detection and emulation using mitre attack framework. In: 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA). pp. 4–12. IEEE (2022)
30. Rios Insua, D., Couce-Vieira, A., Rubio, J.A., Pieters, W., Labunets, K., G. Rasines, D.: An adversarial risk analysis framework for cybersecurity. Risk Analysis 41(1), 16–36 (2021)
31. Sailio, M., Latvala, O.M., Szanto, A.: Cyber threat actors for the factory of the future. Applied Sciences 10(12), 4334 (2020)
32. Shi, Z., Matyunin, N., Graffi, K., Starobinski, D.: Uncovering cwe-cve-cpe relations with threat knowledge graphs. ACM Transactions on Privacy and Security 27(1), 1–26 (2024)

33. Subroto, A., Apriyana, A.: Cyber risk prediction through social media big data analytics and statistical machine learning. Journal of Big Data 6(1), 50 (2019)
34. TI Safe: Incident Hub (2024), https://hub.tisafe.com/
35. Wåreus, E., Hell, M.: Automated cpe labeling of cve summaries with machine learning. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 3–22. Springer (2020)
36. Xie, Y., Jin, X., Xie, T., Lin, M., Chen, L., Yu, C., Cheng, L., Zhuo, C., Hu, B., Li, Z.: Decomposition for enhancing attention: Improving llm-based text-to-sql through workflow paradigm. arXiv preprint arXiv:2402.10671 (2024)
37. Yin, J., Hong, W., Wang, H., Cao, J., Miao, Y., Zhang, Y.: A compact vulnerability knowledge graph for risk assessment. ACM Transactions on Knowledge Discovery from Data 18(8), 1–17 (2024)
38. Zhou, Y., Liu, S., Siow, J., Du, X., Liu, Y.: Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks. Advances in neural information processing systems 32 (2019)

Javier Sánchez García-Ochoa was born in Toledo, Spain in 2000. He received a bachelor's degree in Cybersecurity Engineering from Rey Juan Carlos University (URJC) and a master's degree in Cybersecurity and Privacy from the Open University of Catalonia (UOC). After gaining more than a year of experience in the private sector, he joined Rey Juan Carlos University in 2023 as a research staff member. During this period, he participated in the public-private collaboration project DICYME (Dynamic Industrial Cyber Risk Modelling based on Evidence), where he focused on dynamic cyber risk quantification. He has contributed to several scientific articles and conference proceedings. His research interests currently focus on large language models (LLMs), data science, machine learning, and their applications within the field of cybersecurity.

Jaime Rueda Carpintero was born in Madrid, Spain in 2002. He received a degree in Computer Engineering from Rey Juan Carlos University (URJC), where he also completed a Master's degree in Decision Systems Engineering. After more than a year of experience in the private sector, he joined Rey Juan Carlos University in 2025 as a researcher in the DICYME project. He is a member of the High Performance Research Group Data Science Laboratory (DSLAB, URJC) and has contributed to several scientific articles and conference communications. He actively participates in research tasks and projects focused on natural language processing (NLP), large language models (LLMs), and data science.

Rubén Rodríguez Fernández was born in Bembibre, León, Spain in 1993. He received a PhD degree in Artificial Intelligence from Rey Juan Carlos University (URJC). He also received a master's degree in Data Science from URJC and a master's degree in Artificial Intelligence from the Technical University of Madrid (UPM). He is part of the Data Science Laboratory high performance research group and has been an Assistant Professor at URJC since 2024. His research interests include active learning, explainable machine learning, generative artificial intelligence, and applied machine learning.

Alberto Fernández-Isabel was born in Toledo, Spain in 1984. He received a PhD in Computer Science from Complutense University of Madrid (UCM) in 2015. He obtained

a scholarship at the Spanish National Research Council (CSIC) as a technical assistant. He has been working for several years on European and national projects as a predoctoral and postdoctoral researcher. Since 2019 he is Assistant Professor at the Higher Technical School of Computer Engineering (ETSII) at Rey Juan Carlos University (URJC). He has authored more than 30 scientific articles and books. He completes his background with a Master´s degree in Artificial Intelligence and a Master´s degree in Information Systems. His research interests include intelligent agents, machine learning, data visualization, and natural language processing in various application domains, including distributed programming, sentiment analysis, agent-based collaboration and negotiation, smart cities, and simulations

Isaac Martín de Diego was born in Campaspero, Valladolid, Spain in 1973. He received a PhD degree in Mathematical Engineering from Carlos III de Madrid University in 2005 (Extraordinary Doctorate Award). Since 2023 he is a full professor at the Higher Technical School of Computer Engineering at Rey Juan Carlos University (Associate Professor from 2018). He is the co-founder of the Data Science Laboratory and Head of the Sports Analytics Master at Rey Juan Carlos University. He has been head of the ERICSSON Chair on Data Science applied to 5G. He is the author of more than 100 articles. His research interests include methods, processes, and tools for Data Science in various application domains: explainability, sampling, complexity, performance evaluation, visualization, recommendation systems and security with a special interest in Machine Learning algorithms and a combination of information methods.

Emilio López Cano was born in Madrid, Spain in 1973. He received a PhD degree from Rey Juan Carlos University (URJC). He holds a Master's degree in Decision Systems Engineering (URJC), a degree in Applied Statistics, and a diploma in Statistics from the Complutense University of Madrid (UCM). He is an Associate Professor at Rey Juan Carlos University in the area of Statistics and Operations Research. Previously, he was a teacher and researcher at the University of Castilla-La Mancha since 2011, and worked as a statistician and IT professional in the private sector for 14 years. His research interests include statistical computing, analysis, visualization, and management of complex data and modeling in multidisciplinary environments, where he has several national and international publications and projects.

Romy Rodríguez Ravines received a PhD degree in Statistics from the Federal University of Rio de Janeiro (UFRJ), Brazil. She is an Applied Data Science Leader with over 25 years of experience in business consulting based on advanced analytics, Bayesian modeling, and artificial intelligence, specializing in transforming data into strategic value across the cybersecurity, finance, marketing, and public health sectors. She has served as the Head of Research and Modeling Strategies at DeNexus, where she led the development of a cyber risk quantification platform adopted by international insurers and industrial organizations. She has held various senior leadership positions in consulting firms and has served as a lecturer at several leading universities and business schools in Spain.

Ovidio López, born in Murcia, Spain, holds a Physics degree from the University of Murcia and a Ph.D. in Renewable Energy and Energy Efficiency from the Polytechnic University of Cartagena. His background includes numerical simulation, data analysis,

and Operational Technology (SCADA, PLC). He has experience in R&D and Machine Learning, later working as a Data Scientist focused on industrial cyber risk. Currently, he teaches Electronics in vocational training, sharing expertise in automation and emerging technologies.

Jaume Puigbò Sanvisens was born in Barcelona, Spain. He holds a Bachelor's degree in Mathematics from the University of Barcelona, as well as a Master's degree in Foundations of Data Science (2016–2017) and a Postgraduate degree in Data Science (2015–2016) from the same institution. Since 2021, he has been working as a Data Scientist at DeNexus Inc., where he develops advanced models to quantify cyberattack risk in industrial environments. His research and professional interests include Machine Learning, Deep Learning, Optimization, Recommender Systems, Natural Language Processing, and Data Visualization, with applications in cybersecurity, predictive analytics, and industrial risk modeling. In the past, he has worked on projects such as predictive maintenance for industrial systems, demand forecasting, traffic prediction, and B2B recommender systems, applying advanced statistical and machine learning techniques to real-world challenges. Jaume was a finalist in the UniversityHack 2018 competition and has presented at R user conferences.

Received: October 30, 2025; Accepted: Devember 22, 2025.

Computer Science and Information Systems 23(1):369–396 https://doi.org/10.2298/CSIS251109012Y

FNNMFF: Crop Pests and Diseases Detection Based on Fuzzy Neural Network and Multilevel Feature Fusion in Remote Sensing Images

Shoulin Yin[1], Hang Li[2,⋆], Mirjana Ivanovic[3,⋆], Tao Chen[1], and Lin Teng[1]

[1] College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
[2] College of Artificial Intelligence, Shenyang Normal University, Shenyang 110034 China
lihang@synu.edu.cn
[3] Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia
mira@dmi.uns.ac.rs

Abstract. To solve the problem that the detection effect of crop pests and diseases is not ideal due to the complicated image background and the interference of irrelevant factors, this paper proposes a novel crop pests and diseases detection based on fuzzy neural network and multilevel feature fusion in remote sensing images. Firstly, the model is based on YOLOv5 and extracts the semantic level information of different depth features from the convolutional neural network, and then combines the weight aggregation module to learn the weight of each layer feature adaptively. Then the learned weights are loaded to the segmentation graphs obtained by sampling on each feature layer to obtain the final segmentation results. In this model, a fuzzy learning module is added to the skip connection part to remove noise features and alleviate the uncertainty between classes. The traditional cross entropy loss involves activating the output value with the Softmax function and calculating a weighted cross entropy loss with the label. If the weight of the cross entropy loss term is not adjusted, the model will tend to update the weight related to the background, which makes it difficult to deal with the category imbalance in remote sensing images. Therefore, we use focus loss to alleviate the problem of class imbalance in images. The results on public data sets show that the accuracy rate of the proposed model in this paper is over 95%, the recall rate is over 85%, and the average accuracy is 91.2%. In terms of F1, compared with other advanced methods, the presented method has achieved improvements of 6.6%, 8.5%, and 7.7% respectively. It shows that the new model has strong robustness and generalization for crop pest detection.

Keywords: Remote sensing images, fuzzy neural network, crop pests and diseases detection, multilevel feature fusion, focus loss.

1. Introduction

Globally, the world is home to more than eight billion people, and safeguarding food security is fundamental to every nations economic stability, public well-being and social

⋆ Corresponding authors

cohesion. Transboundary pests and diseases rank among the most destructive threats to agriculture: they occur in countless forms, spread rapidly across regions and climates, inflict severe yield losses and can erupt into large-scale outbreaks. Left unchecked, they endanger global food supplies, rural livelihoods and the achievement of the Sustainable Development Goals [1-3]. Over 70% of the world's food crops are distributed in vast, uniform rain-fed plains or irrigated areas (such as the black soil belt in Northeast China, the middle and lower reaches of the Yangtze River Plain, and the corn belt in the United States). These areas have highly consistent ecological backgrounds, and remote sensing pixel-scale (meters to sub-meters) can cover hundreds or thousands of farmlands, making "spatial extrapolation" possible. Within the same ecological zone, the occurrence patterns of pests and diseases are driven by climate, soil, and cultivation systems, and are highly replicable. Therefore, as long as a remote sensing-ground verification framework is established in representative ecological areas, it can provide decision support for the entire ecological zone by using the representative areas to represent the entire zone, greatly reducing the cost of ground surveys.

Diseases and pests such as rice stem borer, fall armyworm, wheat stripe rust, and potato late blight, which are considered high-risk, can cause a yield loss of $\geq$30% within 5 to 7 days if they break out. The average coverage per person per day through traditional manual field surveys is only $1-2km^2$. In contrast, satellites or drones can obtain images covering hundreds to thousands of km^2 in a single pass, with time resolution reaching "daily" or even "hourly". In scenarios where the epidemic spreads rapidly and the affected area is large, only remote sensing methods can meet the time requirements of "early detection, early warning, and early response".

In recent years, the increasing global climate change has led to an increasing trend in the distribution range and harm degree of crop pests and diseases. According to statistics, in 2020, the global yield losses of wheat, rice and corn caused by pests and diseases reached 22.6%, 31.1% and 23.7%, respectively. The global food production loss caused by pests and diseases accounts for about 14% of the global total food production, among which, the food production loss caused by pests accounts for about 10% of the global total food production. Diseases and insect pests occur in different stages of crop growth and development, and the diseases and insect pests occur in different stages of crop growth. Therefore, diseases and pests should be closely monitored at different growth stages of crops and actively controlled to avoid major losses [4-6].

Precise prevention and control of diseases and pests is an effective measure to reduce losses caused by diseases and pests, improve crop yield and agricultural product quality, reduce pesticide pollution and waste, ensure the safety of agricultural products, protect the ecological environment and increase farmers' income. Accurate monitoring of pests and diseases is the premise and basis of accurate prevention and control of pests and diseases. At present, the control of crop pests and diseases in our country is mainly based on the judgment result of whether the pests and diseases occur and the degree of harm. Manual monitoring method is time-consuming, time-poor and subjective [7-9], which is easy to cause problems such as leakage prevention of diseases and pests and low prevention and control efficiency. Moreover, this pest control method generally ignores the difference in the degree of damage caused by diseases and pests in different regions, and adopts a uniform spraying method, resulting in resource waste, pesticide residues, agricultural product damage and environmental pollution.

The development of modern agriculture and the demand for precision medicine applications have put forward new requirements for pest monitoring. Crop diseases and pests mainly act on the metabolic system of crops, causing damage to the physiological function and structure of the internal system of crops. External manifestations are wilted, speckled, mildew powder and other obvious characteristics. Internal manifestations are changes in biophysical and chemical characteristics, such as the concentration or content of pigments such as chlorophyll and carotene, as well as a decrease in nutrient composition. These changes will lead to changes in the spectral characteristics of crops. Therefore, the biophysical properties and spectral characteristics of crops can be monitored through machine vision, hyperspectral technology and remote sensing technology, which is the research hotspot of crop pest monitoring [10,11].

With the rise of computer vision technology and artificial intelligence, image processing technology has been vigorously developed, because of low cost, easy to implementation, these technologies are widely used in agricultural pest detection. Traditional agricultural image processing mainly adopts machine learning algorithms such as support vector machine [12,13] and K-nearest neighbor [14], but the traditional machine learning technology needs to rely on manual feature selection and extraction, which has weak generalization ability and great environmental restrictions, and is not suitable for application in the actual orchard environment. Deep learning is a specific field of machine learning. Compared with machine learning, deep learning can automatically extract feature information, reduce the burden of feature extraction engineering and manual extraction prone to errors, suitable for processing large-scale data sets, and provide better prediction and generalization capabilities. Common target detection models in deep learning include Faster RCNN [15], SSD [16], YOLO series [17,18], etc., which have been widely applied to agricultural pest detection research. Li et al. [19] proposed a new Faster R-CNN architecture to detect seven different strawberry diseases and achieved good detection accuracy. Cheng et al. [20] constructed a lightweight convolutional neural network MEAN-SSD to reduce the size of the model, and successfully detected five apple leaf diseases on mobile devices. Faisal et al. [21] identified the diseases of citrus fruits by combining YOLOv4 and EfficientNet models. Ma et al. [22] detected cucumber diseases based on YOLOv5n, and introduced coordination attention mechanism and a Transformer structure to improve the detection accuracy and robustness of the model. Zhu et al. [23] used the improved YOLOv5s to achieve rapid detection of apple fruit diseases. Majeed et al. [24] realized the classification of monocotyledonous plants and dicotyledonous plants by identifying and detecting the morphological characteristics of leaves through image digitalization, and the accuracy rate reached 60%-80%. Fan et al. [25] studied a weed recognition method based on color features, which was not easily affected by factors such as shooting distance and occlusion. The test results showed that the recognition rate of wheat and weed by this method reached 54.9% and 62.2% respectively. Zhang et al. [26] proposed an improved single-step multi-frame detector model based on lightweight convolutional neural networks combined with feature information fusion mechanism. The VGG16 network in SSD model was replaced by a lightweight pre-basic network to improve the speed of image feature extraction. The deep semantic information and shallow semantic information were integrated to improve the detection accuracy of small crops and weeds. The final results showed that the average accuracy was 88.27%, and the detection speed and parameter number had been greatly improved. Yang et al. [27] proposed a maize leaf pest

detection model for YOLOv3-Corn. This model used Darkknet-53 as the feature extraction network, and used the clustering algorithm to select prior boxes and match them to the detection layer for target identification. The accuracy and recall rate of model detection were 93.31% and 93.08%, respectively. Guan et al. [28] studied the detection problem of corn seedlings, improved the Cascade R-CNN model, used residual network and feature pyramid as feature extractors, and made the average accuracy of corn seedling detection reach 91.76%.

At present, most studies on crop pests and diseases tend to focus on classification tasks [29] with few object categories, and the data sets used are taken in a laboratory environment based on a simple background. Due to the similar characteristics among crop pests and diseases, complex natural environment, and occlusions of branches and leaves, the existing models are prone to problems such as missed detection and false detection, and can not meet the requirements of multi-species and multi-target pest detection in a natural environment. To solve the above problems, this paper proposes a novel crop pests and diseases detection based on fuzzy neural network and multilevel feature fusion in remote sensing images, aiming to assist crop farmers in detecting diseases and pests, and provide an algorithm basis for intelligent products such as orchard drug application robots. The main contributions are as follows.

1. We introduce the CXV2 backbone block that replaces C3 in YOLOv5s, boosting feature diversity under complex field background.
2. We design a lightweight Fuzzy Learning Module (FLM) inserted in skip-connections to suppress mixed-pixel uncertainty at object boundaries.
3. We equip the neck and head with CARAFE up-sampling and DyHead, achieving accurate multi-scale pest localization in 0.5-2m resolution UAV images.
4. The resulting FNNMFF detector, trained with EIOU loss and SimAM attention, improves mAP by 1.9% and F1 by 6.6% while using only 1.29M parameters, enabling real-time mobile deployment.

2. Related Works

The traditional management mode of crops during planting mainly relies on manual observation of pests and diseases, which has some problems, such as strong subjectivity and heavy workload. With the continuous development of artificial intelligence and big data technology, the use of machine vision technology based on deep learning to extract image shape, color, texture and other feature information can effectively solve the problem of low accuracy and low efficiency of disease and insect detection during crop planting, which is also an important research direction of intelligent crop planting management [30]. At present, many scholars have carried out a lot of research on the application of deep learning and machine vision technology to crop management.

In order to realize accurate monitoring of crop pests and diseases, it is necessary to select an appropriate algorithm structure and build a relationship model between the selected spectral characteristics and the types of pests and diseases and the degree of harm. At present, classical statistical models, computer image processing methods, machine learning methods and deep learning algorithms are widely used in remote sensing monitoring and forecasting of crop pests and diseases.

2.1. Classical Data Statistical

Classical statistical analysis methods mainly include regression models, principal component analysis, cluster analysis, discriminant analysis and other methods [31-33]. Due to its advantages of simple form and clear mechanism, it is widely used in the monitoring and research of crop pests and diseases. Santos et al. [34] analyzed the spectrum of correlation coefficients by analyzing regression from partial least squares (PLS), and determined the effective wavelength by a stepwise multiple linear regression (SMLR) procedure. These analytical methods revealed several important regional wavelengths and produced predictive models of disease severity based on absorption spectra. The optimal model was used to predict disease severity in the validated dataset, with the root-mean-square difference (RMSD) of 4.9% and the coefficient of determination R2 of 0.82. Deng et al. [35] analyzed the hyperspectral data of wheat canopy stripe rust at different growth stages, and used statistical methods such as t-test and correlation analysis to screen the characteristic bands. A set of feature band screening methods for disease diagnosis was established. The four wavelengths suitable for early detection of stripe rust were 576nm, 705nm, 712nm, 1416nm and the five wavelengths suitable for intermediate detection were 558nm, 632nm, 375nm, 696nm, 712nm. Lin et al. [36] obtained canopy spectra and disease indices of infected winter wheat at different growth stages, and extracted the first five principal components using principal component analysis within the range of 350nm 1350nm. And the first three principal components of the first order differential spectrum in the blue (490nm 530nm), yellow (550nm 582nm) and red (630nm 673nm) bands were used to construct the disease inversion model for detecting winter wheat stripe rust. Chattopadhyay et al. [37] used ASD spectrometer to measure two kinds of hyperspectral data of wheat leaf stripe rust and mildew. Through correlation analysis and independent t-test, the spectral characteristic bands with obvious difference in sensitivity of the two diseases were selected, and the disease discrimination model and disease severity inversion model were constructed. Kolipaka et al. [38] selected wheat 25 planting yield index and DI to conduct linear, stepwise regression and BPNN modeling, and screened the optimal wheat leaf rust monitoring model at different growth stages. When studying cotton canopy leaves, Li et al. [39] found that the multivariate stepwise regression model established based on the first derivative spectrum of cotton leaves could effectively estimate the chlorophyll content of cotton leaves.

2.2. Computer Image Processing Methods

With the rapid development of science and technology, image processing technology has been widely used in the identification and diagnosis of crop pests and diseases. After crops are infected by pathogenic bacteria, a series of metabolic changes will occur, resulting in changes in the external characteristics of crop leaves such as color, shape, texture and spectrum. Different diseases have different plaque characteristics caused by different pathogens. Therefore, it is possible to determine whether crops are infected with diseases through the image information of crop leaves [40].

The research on pest recognition and diagnosis based on traditional image processing technology mainly includes four parts: image preprocessing, image feature extraction, image feature reduction and pest pattern recognition model construction. Tran et al. [41] used the method of spectral band combination to study the changes between the red,

green and blue spectral channels of wheat plants and determine the area of wheat diseased leaves, thus realizing the distinction between healthy plants and diseased plants. Kumar et al. [42] classified wheat leaf rust based on infrared thermal imaging edge detection algorithm. Mumtaz et al. [43] proposed an improved threshold median filtering algorithm, which could effectively remove noise and retain image details.

In the aspect of image feature extraction, edge detection is an important research method in computer vision, and it is also a key technology for analyzing crop images. Edge detection is not only an important method of digital image processing, but also a key technology of crop image analysis. Maraveas et al. [44] introduced the ant colony algorithm into the thermal image processing of maize drought, and proposed to apply edge detection to infrared thermal image analysis. The purpose of feature selection was to select more favorable features for pest identification. Principal component analysis was widely used as a common linear dimension reduction method. Masood et al. [45] proposed a method for color image recognition of maize leaf pests and diseases by using a manifold learning algorithm in RGB color space. Siddiqa et al. [46] proposed a nonlinear dimensionality reduction method for multispectral images, aiming at the problem of high data dimension of multispectral images resulting in high data processing complexity in image color reproduction. After the feature selection of plant pests and diseases, support vector machine and artificial neural network are generally used to complete the classification of plant pests and diseases. Chodey et al. [47] proposed a pest control method based on image gray frequency and artificial neural network. Gupta et al. [48] discussed a rice canopy image segmentation algorithm based on the combination of support vector machine and maximum inter-class variance method to solve the difficult problem of rice canopy image segmentation caused by variable light intensity in the field.

Traditional image processing has poor migration ability among plant pests and diseases, and the performance of lesion segmentation and pest recognition is not ideal. The emergence of convolutional neural networks effectively improves the migration ability of plant pests and diseases, and its end-to-end feature extraction capability provides technical support for real-time and accurate diagnosis of plant leaf diseases and pests. Chen et al. [49] proposed a low-resolution rice pest recognition network based on SCResNet (Self-calibrated convolutions and ResNet block for ResNet50) in order to solve the problem of low quality pest images captured in the natural environment of rice fields. By pruning the low-resolution image, constructing and training the ESRGAN network to generate Super-resolution (SR) image equivalent to the original image, the accuracy of rice pest identification was further improved. Khanna et al. [50] used convolutional neural network and computer vision technology to realize image recognition of a total of 12 leaf diseases of three crops (soybean, corn and ginseng), and improved the existing convolutional neural network model to significantly improve the model recognition rate. Under the framework of deep learning Keras, Guerrero-Ibanez et al. [51] trained the deep convolutional neural network with the normalized data of three rice diseases and established three rice disease recognition models. Imanulloh et al. [52] proposed a convolutional neural network image recognition model of maize pests and diseases based on transfer learning in order to realize image recognition of maize pests and diseases under complex field background. Du Al-Timemy et al. [53] combined the deep feature fusion algorithm with the convolutional neural network to remove the redundant features of the feature image on the premise of better retention of relevant information, and realized the rapid detection of tomato leaves.

Sahasranamam et al. [54] proposed a rice pest and disease identification method based on the Internet of Things and deep convolutional neural network, aiming at the problem that the rice pest and disease identification method based on image segmentation and convolutional neural network was sensitive to light and background.

At present, the application of computer image processing technology is very wide. In the process of agricultural production, image processing technology can be used for real-time monitoring of crop pests and diseases. Compared with manual monitoring, image processing technology can obtain pest information accurately and quickly, making diagnosis more accurate and effective. In the future, the combination of image processing technology and computer technology will make the monitoring of agricultural crop pests and diseases develop in the direction of automation and intelligence, which will inevitably improve the efficiency of agricultural production.

2.3. Machine Learning Methods

In recent years, machine learning methods have been gradually applied in the fields of canopy height (CH) and vegetation coverage (VFC) of crop diseases monitored by remote sensing and pest stress. The main components of machine learning methods and satellite-based spectra calculated by Sentinel-2A/B data are classical machine learning methods and deep learning methods. A classical learning machine uses machine learning models, including partial least squares regression (PLSR), support vector regression (SVR), random forest regression (RFR), and extreme learning regression (ELR), to evaluate the potential of drone data to replace manually sampled data and predict field disease indices (DI). The results show that the performance of SVR method is slightly better than other methods, mainly including support vector machine, decision tree and K-nearest neighbor algorithm, etc. The required training sample data is small, the equipment performance requirements are lower, and the model is easier to understand. Zou et al. [55] obtained the root mean square error (RMSE) of 1.89% by analyzing the UAV. In addition, the combination of canopy structure (CS) and vegetation index (VIs) improved the prediction accuracy compared with a single type feature (RMSEs 2.86% and RMSEVIs 1.93%). In order to distinguish wheat powdery mildew from aphids on a regional scale, Zhang et al. [56] adopted BPNN and support vector machine (SVM) methods based on the original training data set to conduct preliminary tests on crop growth and environmental parameters. The results showed that the overall accuracy of the two-time growth index model, SMOTE-BPNN model, BPNN model and SVM model was above 80% in combination with the growth and environmental parameters of different crop periods. The combination of SMOTE and a BP neural network could effectively improve the identification accuracy of small individual diseases and pests.

Deep learning is an emerging branch of machine learning that enables automated feature learning and pattern recognition by building deep neural network models. The main feature of deep learning is that it has a multi-layer network structure that can learn an abstract hierarchical representation of the data to extract more advanced and abstract features. In some complex scenarios, deep learning methods have shown powerful capabilities in monitoring crop pests and diseases. Irmak et al. [57] used AlexNet and GoogleNet to train 14828 images of tomato leaves infected with diseases, understand the symptoms of the leaves and locate the disease area through visualization, and realized the classification of tomato leaf images containing nine diseases. Roy et al. [58] used hyperspectral and

UAV remote sensing technologies to monitor cotton root rot, and the results showed that both could accurately identify the susceptible area under unsupervised classification. Sun et al. [59] used UAV remote sensing technology to monitor potato late blight and used five classification methods to classify disease areas, and the results showed that linear support vector machine and random forest algorithm had the best classification effect. Nayak et al. [60] proposed a transfer learning method to improve the problem of insufficient sample quantity of the rice lesion image set. The algorithm was based on pre-trained DDC deep transfer learning, and the rice image set is transferred to a pre-trained convolutional neural network model on a large dataset for retraining. Aiming at the problem that different training samples differ greatly, the DDC algorithm is proposed for optimization, and the final training results are compared with those directly trained on the training set, which verifies the effectiveness of the proposed algorithm and proves that the algorithm has stronger representation ability. Nguyen et al. [61] collected the data information of wheat powdery mildew, stripe rust, aphids, total erosion and other diseases, and used pattern recognition and machine learning algorithms to distinguish, estimate and predict various disease types.

Kumar et al. [62] proposed a recognition model based on a convolutional neural network, which combined batch normalization and global pooling techniques and was able to effectively identify 26 diseases of 14 different plants. The model was less affected by changes in leaf spatial position and had high recognition accuracy and robustness. Zhang et al. [63] made a comparative analysis of discrimination algorithms such as Extreme learning machine (ELM), RBF neural network, random forest, support vector machine and K-nearest neighbor (KNN) to identify the healthy and diseased areas of rape leaf disease parts.

In summary, deep learning has many advantages over traditional machine learning, including a high degree of automation, fast computing speed, and the ability to learn features autonomously. Through deep learning, tasks and features can be jointly modeled to extract more accurate information. However, in more complex scenarios, there may be a low recognition rate.

3. Proposed Pest and Disease Detection Model

Among the YOLO series models, YOLOv5 has the advantages of high detection accuracy, fast speed, small model and easy deployment, and it is often used in various environmental target detection scenarios. According to the model size, YOLOv5 is divided into five versions. Through comparative experiment analysis on self-built data sets, YOLOv5s is selected as the baseline model in this study. The YOLOv5s model is mainly divided into four parts: input layer, backbone network, neck network and detection head. The input layer premanipulates the detected image and passes the image into the model. The backbone network is used to extract the features in the input image, and the CSPDarknet53 architecture is used to effectively improve the efficiency of feature transmission. The neck network is the feature fusion layer. Feature pyramid network (FPN) [64] and path aggregation network (PAN) [65] are used to fuse feature maps of different scales to improve the detection ability of different feature layers. The detection head is used to map the extracted features and output the location, category and confidence information of the predicted target.

In order to improve the detection performance of crop pests and diseases in a natural complex environment, a new crop detection model FNNMFF is proposed in this study. A CXV2 module is designed to replace the C3 module in the original model, and the dynamic detection head (DYHEAD) is used to replace the original detection head, and the upsampling mode in the neck network is replaced by the CARAFE upsampling operator. The detection accuracy of crop pests and diseases is improved on the basis of not significantly increasing the calculation amount of the model. The network structure of FNNMFF is shown in Figure 1.

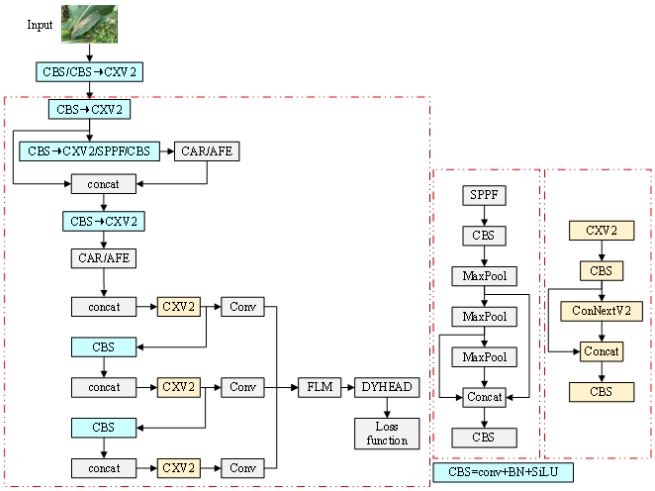


Fig. 1. Proposed FNNMFF model. Given a UAV or satellite remote-sensing image, the backbone first extracts multi-level features via the improved CXV2 blocks; the neck then fuses them through CARAFE up-sampling and PAN/FPN, while the embedded Fuzzy Learning Module (FLM) suppresses uncertain edges; finally, the Dynamic Head (DYHEAD) outputs pest-disease bounding boxes

3.1. CXV2 Module

In order to improve the feature extraction ability of the model, obtain channel features with more resolution, and enhance the generalization ability of the model, the ConvNeXxtV2 model is introduced in this study, and a CXV2 module is designed to replace the C3 module in YOLOv5s. ConvNeXxtV2 adopts the full convolutional mask autoencoder (FCMAE) mechanism, which combines the mask autoencoder MAE mechanism with self-supervised learning technology to improve the training speed and accuracy of the model. The structure of the ConvNeXtV2 module is shown in Figure 2. The global response normalization (GRN) layer is added to the original architecture of ConvNeXxt, which can solve the feature crash problem in the training process. The combination of deep convolution operation to enhance the feature competition between channels, im-

prove channel contrast and selectivity, is conducive to improving the diversity of extracted features in complex background images.

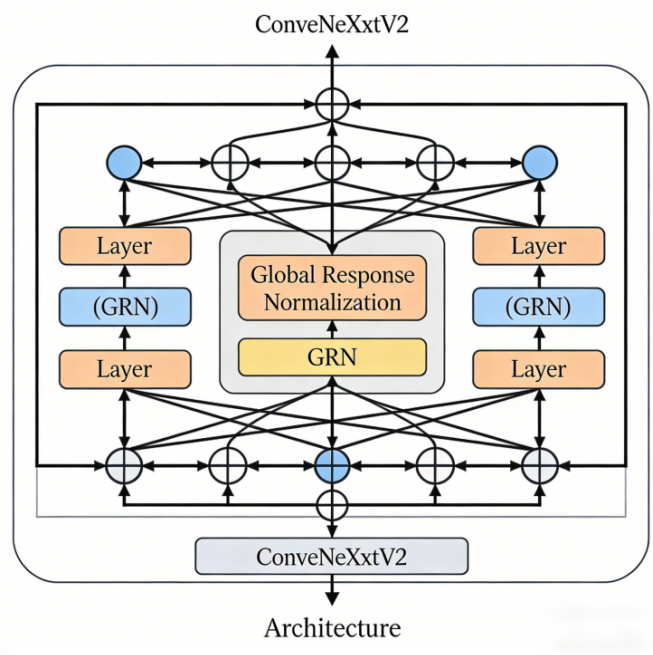


Fig. 2. ConvNeXxtV2 module structure

The CXV2 module uses a residual-like structure. Input features are first extracted through a 1×1 standard convolution block CBS to reduce the number of channels, and then transmitted to two different branches. One of the branch features is passed into the ConvNeXxtV2 module, which enhances feature extraction through module processing. The features of the other branch remain unchanged. Finally, all branches are spliced, and the number of channels is recovered by 1×1 standard convolution block operation to output features that enhance diversity and improve the feature expression ability of the model.

3.2. Dynamic Detection Head (DYHEAD)

Different diseases and pests have similar pathological characteristics, and in the natural environment, they are easily confused with the background. The scale and different spatial locations of the diseased leaves also bring challenges to feature extraction and target detection. To solve the above problems, a dynamic head is added in this study, and the frame of the dynamic head is shown in Figure 3. The features of the original feature pyramid network are scaled back and reshaped into a three-dimensional tensor $F \in R^{L \times S \times C}$

with the same scale, and different attention mechanisms are used in each independent dimension. Here L is the feature hierarch; S is the product of width and height; C indicates the number of channels.

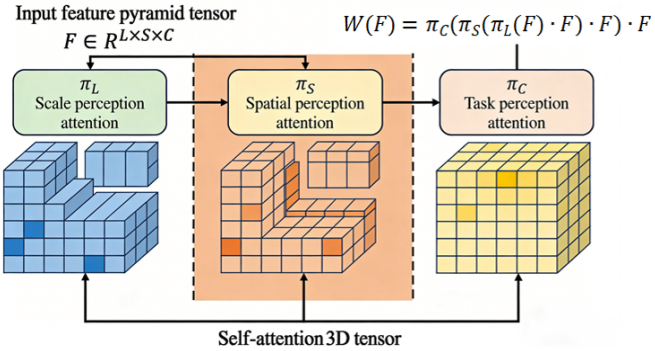


Fig. 3. Dynamic detection head

In order to avoid the high computation caused by the fully connected layer, the dynamic detection head adopts the mode of 3 sequential attention, and its self-attention equation is shown in formula (1), each attention module corresponds to only one dimension.

$$W(F) = \pi_C(\pi_S(\pi_L(F)\cdot F)\cdot F)\cdot F. \tag{1}$$

Where π_L is the scale perception attention function, increasing the perception ability of different scale features. π_S is the spatial perception attention function, which enhances the spatial position perception ability of the model. π_C is a task perception attention function to enhance the perception of different task objectives.

3.3. Carafe Upsampling Module

In the YOLOv5s model, feature pyramid is an important part of neck feature extraction and fusion. The model needs to match and fuse deep semantic information with shallow semantic information by upsampling. The original YOLOv5s model uses the nearest neighbor upsampling method, which only determines the upsampling kernel based on the spatial position of pixels, resulting in a small perceptual domain and is unable to make full use of the semantic information of feature maps. In this study, the upsampling module of CARAFE is used to replace the upsampling operator of the original model.

In the kernel prediction module, 1×1 convolution kernel is first used to compress the number of channels, and the original number of channel C is reduced to C_m, which reduces the subsequent calculation and improves the efficiency. After that, the pixelshuffle is used to reassemble the height, width and channel number of the feature map. The convolution layer of $K_{encoder} \times K_{encoder}$ is used to predict the upper sample kernel for the feature map after the compressed channel number, and the shape of the predicted upper sample kernel is $\sigma H \times \sigma W \times K_{up}^2$.

3.4. Fuzzy Learning Module (FLM)

In remote sensing images, small targets and boundary pixels are affected by mixed pixels and noise, which leads to more uncertainty in segmentation results. Compared with the traditional precise mathematical statistics method, fuzzy theory can deal with the uncertainty problem more effectively. In this study, a new FLM is designed and proposed to improve the segmentation accuracy of small targets and boundary regions. As part of a network, FLM can be seen as a fuzzy layer described by multiple parameters in a Gaussian membership function. The FLM structure is shown in Figure 4, where $H \times W \times C$ represents the size of the input feature map, and H, W and C represent the height and width of the feature map and the number of channels of the input feature map respectively.

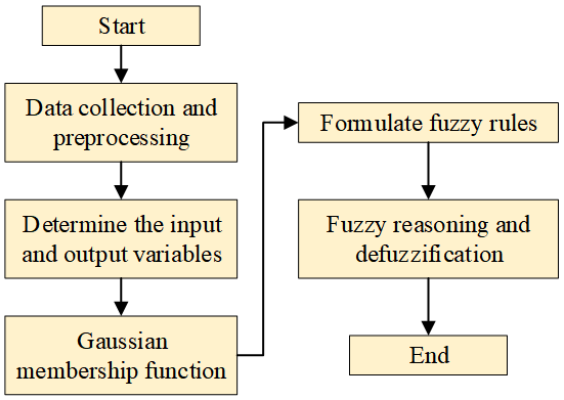


Fig. 4. FLM structure

In FLM, each Gaussian membership function assigns a fuzzy semantic tag to a feature point, and the Gaussian membership function is expressed as:

$$G_g^c(n_1, n_2) = e^{\frac{n_2^2+n_1^2}{22}}. \tag{2}$$

Where g and σ are constants. c is the channel. n_1 and n_2 are the difference between the x and y coordinates of the current pixel and the center coordinates, respectively. and are the coordinates of the selected center point in the image, i.e. $n_1 = x_1\mu_1$, and $n_2 = x_2\mu_2$ determine the center point for all pixels in the selected area, i.e.,

$$\mu_1 = \frac{1}{i} \times \sum_{0}^{i}(x_1 + \cdots + x_i). \tag{3}$$

$$\mu_2 = \frac{1}{i} \times \sum_{0}^{i}(y_1 + \cdots + y_i). \tag{4}$$

Where i is the number of pixels in the selected region.

The weight of each adjacent pixel in the channel is expressed as:

$$W^C(n_1, n_2) = \frac{G_g^c(n_1, n_2)}{\sum\sum G_g^c}. \tag{5}$$

Multiplying the pixel value for each position with the corresponding element in the weight matrix, it sets the center point at the origin, and determines σ. Finally, the values of centroid and adjacent points are added to obtain the pixel values of centroid after fuzzy learning. The correlation formula of fuzzy radius r is:

$$\tilde{w}_{ij} = exp(-\frac{(x_{ij} - \mu_1)^2 (y_{ij} - \mu_2)^2}{2\sigma^2}). \tag{6}$$

$$f(x) = \begin{cases} Sum(\tilde{W}), & \sqrt{(x_{ij} - \mu_1)^2 + (y_{ij} - \mu_2)^2} \leq r, \\ \varpi, & \sqrt{(x_{ij} - \mu_1)^2 + (y_{ij} - \mu_2)^2} > r \end{cases} \tag{7}$$

Where $\tilde{w}_{ij}$ and w_{ij} are elements in the initial weight matrix $\tilde{W}$ and the final weight matrix W, respectively. $Sum(\cdot)$ is the sum of the elements in $\tilde{W}$.

The process is repeated for all the points in the image, and the fuzzy semantic features are extracted by using FLM to learn the inherent rules between the feature map and the corresponding segmentation results.

3.5. Loss Function

The loss function plays an important role in model training. It can evaluate the distance between the model prediction frame and the real frame. The closer the distance is, the smaller the loss function value is. For different detection problems, selecting the appropriate loss function can make the model converge faster, locate more accurately and achieve better results during training.

Among the existing loss functions, intersection overunion (IOU) is chosen by most target detection algorithms as a measure of the distance between the predicted box and the real box. From a mathematical point of view, IOU is equal to the intersection ratio of two rectangular boxes, that is, the ratio of the overlapping area of the target real box and the model prediction box in the detection image to the area occupied by the two boxes as a whole. The IOU loss function is calculated as follows:

$$IOU = \frac{|B \cap B_i|}{|B \cup B_i|}. \tag{8}$$

$$L_{IOU} = 1 - IOU = 1 - \frac{|B \cap B_i|}{|B \cup B_i|}. \tag{9}$$

Where B is the area occupied by the target real frame. B_i is the area occupied by the model prediction box $L_{IOU} \in (0, 1)$.

The larger the overlap area between the real box and the prediction box, the smaller the L_{IOU}, and the more accurate the predicted object region. However, when the real box and the prediction box do not overlap completely or do not overlap at all, the IOU cannot reflect the coincidence degree and distance between the two, making the model unable to train. The $CIOU$ loss function can then be used to estimate the distance between the

model prediction box and the real box. $CIOU$ includes center point distance, aspect ratio, and overlap area information between the two boxes to make the target border regression more stable. The $CIOU$ loss function is calculated as follows:

$$L_{CIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + av. \tag{10}$$

$$a = \frac{v}{(1 - IOU) + v}. \tag{11}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2. \tag{12}$$

$$IOU = \frac{B \cap B_i}{B \cup B_i}. \tag{13}$$

Where h is the height of the prediction frame; w is the width of the prediction box; h^{gt} is the height of the real box; w^{gt} is the width of the true box; b is the center point of the prediction frame; b^{gt} is the center point of the target frame; ρ^2 is the Euclidean distance between the center points of the two frames; c contains the diagonal distance of the minimum closure rectangular box of both boxes.

Although the CIOU loss function makes up for the shortcomings of the IOU loss function, it still ignores the real difference between the size of the bounding box and its confidence. To solve this problem, Zhang et al. [66] proposed efficient intersection over union (EIOU). The penalty item of EIOU is to separate the influence factor of aspect ratio to calculate the length and width of the target box and anchor box, which includes three parts: IOU loss, center point distance loss and width, height loss, in which the width and height loss directly minimizes the difference between the width and height of the target box and anchor box. The EIOU-Loss formula is as follows.

$$EIOU = IOU - \frac{\rho^2(b, b^{gt})}{c^2} - \frac{\rho^2(w, w^{gt})}{c_w^2} - \frac{\rho^2(h, h^{gt})}{c_h^2}. \tag{14}$$

$$L_{EIOU} = 1 - EIOU. \tag{15}$$

Where c_h is the height of the minimum external frame covering the prediction frame and the real frame. c_w is the width of the minimum external frame covering the predicted frame and the real frame.

In the frame regression loss, EIOU loss function solves the problems existing in other loss functions and shows good performance. Therefore, this paper will use EIOU Loss function to improve the model.

4. Experimental Results and Analysis

4.1. Experimental Environment

This experiment is based on a deep learning framework Pytorch model, model training and testing are on 64-bit Windows 10 system. Experimental platform parameters are CPU AMDRYZEN R7 6800H 3.20GHz, memory 16GB, GPU NCIDIA GE Force RTX 3060,

and video memory 6GB, Python 3.8, Pytorch 1.10.1. In order to improve the speed of network training, GPU is used for acceleration, and the software versions are CUDA 11.3 and CUDNN 8.2.1.

Input resolution is 640×640, augmentation is with Flip, Color jitter, Mosaic. Learning rate=$1e^{-2}$ (SGD + cosine). Batch size=16. Epochs=200. Optimizer adopts SGD (momentum=0.937). Weight decay=$5e^{-4}$.

In this paper, PlantVillage dataset (https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset) is a widely-used benchmark for automated plant-disease recognition. Within its tomato subset, 18160 leaf images capture ten distinct conditions: bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, spider-mite damage, two-spotted spider-mite damage, target spot, tomato yellow leaf curl virus, and healthy foliage. Representative examples appear in Figure 5.



Fig. 5. Examples of different tomato plants

4.2. Evaluation Index

In the test of pest and disease detection model, precision (P), recall (R) and mean average precision (mAP) are used as performance evaluation indicators in this study. The specific calculation formula is as follows:

$$P = \frac{TP}{TP + FP} \times 100\%. \tag{16}$$

$$R = \frac{TP}{TP + FN} \times 100\%. \tag{17}$$

$$mAP = \frac{\sum_{K=1}^{K} AP(K)}{K}. \tag{18}$$

Where TP is the correct sample quantity for detection; FP is the number of samples with error detection; FN is the number of samples missed by detection; K is the number of detection categories; AP is the area enclosed by the P-R curve; P is the proportion of correctly detected samples in the number of all detected samples; R is the proportion of correct samples in all samples detected; mAP is the average of AP values for all categories.

4.3. Ablation Experiments

In order to verify the superiority of the proposed YOLOv5s model in this study, the same data set is used under the same experimental environment, and the YOLOv5s (Alpha IOU) and YOLOv5s (SIOU) models with Alpha IOU, SIOU and EIOU loss functions are compared. YOLOv5s(CA), YOLOv5s(SE) and YOLOv5s(CBAM) models with CA, SE and CBAM attention mechanisms are compared.

(1) Comparison of YOLOv5s test results with different loss functions.

The loss function used by the original YOLOv5s model is CIOU. As can be seen from Table 1, after the Alpha IOU, SIOU and EIOU loss functions are added to the model, although R value has decreased compared with the original model, P value has significantly increased, increasing by 3.5%, 3.4% and 6.9% respectively. In particular, the improvement of EIOU loss function is most significant. Increased P indicates that the EIOU loss function is more accurate in calculating the distance relationship between the predicted frame and the real frame. Figure 6 presents the results intuitively, eliminating the redundancy of numbers, which makes it easier for readers to understand.

Table 1. Comparison of test results with different loss functions/%

Model	P	R	mAP
YOLOv5s(CIOU)	89.3	83	88.3
YOLOv5s(Alpha IOU)	92.8	81	88.7
YOLOv5s(SIOU)	92.7	79.3	87.4
YOLOv5s(EIOU)	96.2	79.3	88.2

(2) Comparison of YOLOv5s test results with different attention mechanisms.

As can be seen from Table 2, no matter what attention mechanism module is added to the original YOLOv5s model, the performance is significantly improved compared with the original model. For example, P after adding CA attention mechanism is 3.5% higher than the original model, R after adding SE attention mechanism is 1.9% higher than the original model, and R and mAP after adding CBAM attention mechanism are 0.7% and 2.4% higher than the original model, respectively. After the addition of SimAM attention mechanism, R improves by 1.3% compared to the original network. The improvement of R shows that adding attention mechanism can make the model better evaluate the weight proportion of different features, and make the target detection result more accurate.

(3) Comparison of YOLOv5s test results with the simultaneous improvement of loss function and attention mechanism.

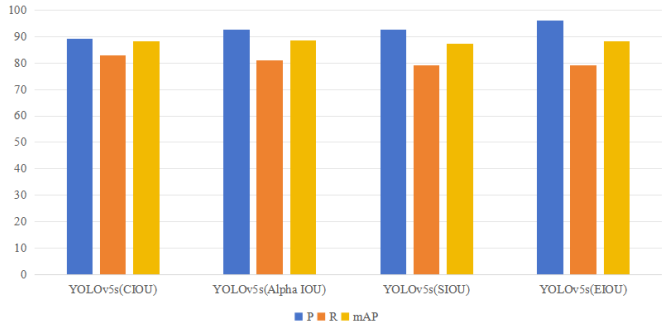


Fig. 6. Comparison of test results with different loss functions

Table 2. Comparison of test results with different attention mechanisms/%

Model	P	R	mAP
YOLOv5s	89.3	83	88.3
YOLOv5s(CA)	92.8	82.1	88.7
YOLOv5s(SE)	88	84.9	88.3
YOLOv5s(CBAM)	89.1	83.7	90.7
YOLOv5s(SimAM)	89.1	84.3	88.2

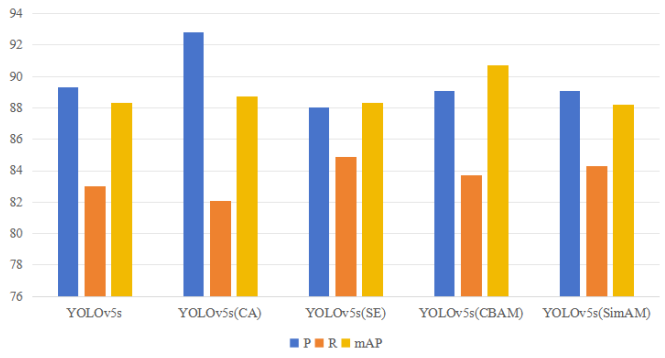


Fig. 7. Comparison of test results with different attention mechanisms

In the above experiments, the loss function and attention mechanism in improved YOLOv5s are compared respectively. The effects of improving both the attention mechanism and the loss function on model performance will be further explored below. As can be seen from Table 1, EIOU loss function has the better effect, so it is only necessary to fix EIOU as the model's loss function by combining and comparing it with CA, SE, CBAM and SimAM attention mechanism respectively. As can be seen from Table 2 (Figure 7) and Table 3 (figure 8), the P of the improved YOLOv5s is 4.3%, 5.0%, 2.5% and 5.4% higher than that of the original model respectively. The R of CA and SE attentional mechanisms decreases by 0.4% and 2.7% respectively, while CBAM and SimAM attentional mechanisms increases by 1.0% and 0.5%. The mAP of CA, CBAM and SimAM attention mechanism increases by 0.4%, 1.1% and 1.9% respectively, while SE attention mechanism decreases by 0.9%. In particular, the P, R and mAP of YOLOv5s(EIOU+SimAM) model are generally superior to other improved models, which proves that the improved model proposed in this study can effectively detect crop pests and diseases.

Table 3. Font sizes

Model	P	R	mAP
YOLOv5s(EIOU+CA)	93.6	82.6	88.7
YOLOv5s(EIOU+SE)	94.3	80.3	87.4
YOLOv5s(EIOU+CBAM)	91.8	84	89.4
YOLOv5s(EIOU+SimAM)	94.7	83.5	90.2

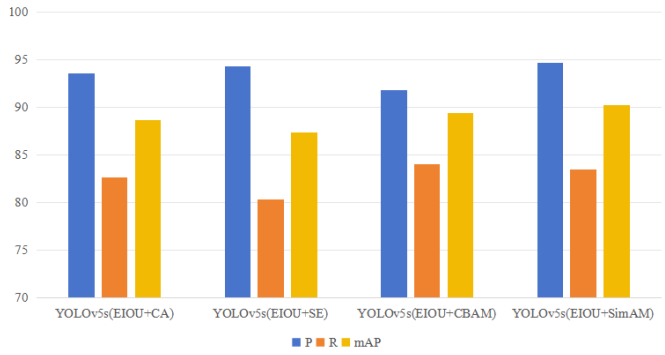


Fig. 8. Comparison of test results with different loss functions and attention mechanisms

4.4. Comparison Experiment

To further verify the performance of the YOLOv5s(EIOU+SimAM) model, the performance is compared with the original YOLOv5s model. After 200 iterations of the training set on YOLOv5s(EIOU+SimAM), P is 92.3%, R is 94.1%, and mAP is 95.3%. As can be

seen from Table 4 and Table 5, the performance of the improved model is significantly improved compared with the original model.

Table 4. YOLOv5s comparison of model performance/%

Iteration	P	R	mAP
50	95.1	82.8	83.7
100	89.4	90.2	89.8
150	90.8	91.2	90.9
200	92.0	93.8	94.7

Table 5. FNNMFF comparison of model performance/%

Iteration	P	R	mAP
50	86.7	88.5	87.8
100	90.5	91.7	90.4
150	91.9	92.4	93.6
200	92.3	94.1	95.3

The original models of FNNMFF and YOLOv5s are tested with 200 images of maize crop pests and diseases in the test set. As can be seen from Table 6, overall P, R and mAP of the improved model have increased by 5.4%, 0.5% and 1.9% respectively compared with the original model. In the detection of slime, rust and plaque, it is found that P of the FNNMFF model is 4.6%, 12.1% and 3.8% higher than that of the original model. Compared with the original model, the R of the FNNMFF model is increased by 7.6% and 3.1%, respectively. When detecting the images of slime worms, grey planthopper and rust disease, it is found that compared with the original model, the mAP of the FNNMFF model is increased by 4.7%, 2.4% and 1.9% respectively.

Table 6. Font sizes

Type	Model	P	R	mAP
Slime	YOLOv5s	89.7	79.2	88.7
Slime	FNNMFF	94.3	86.8	93.4
Grey planthopper	YOLOv5s	89.9	95.5	96.8
Grey planthopper	FNNMFF	90.9	98.6	99.2
Rust disease	YOLOv5s	85.8	78.2	82.3
Rust disease	FNNMFF	97.9	72.0	84.2
Spot disease	YOLOv5s	91.9	79.2	85.3
Spot disease	FNNMFF	95.7	76.8	84.0
Average	YOLOv5s	89.3	83.0	88.3
Average	FNNMFF	94.7	83.5	90.2

The 6.2% Recall drop on rust is mainly attributed to 68% of missed instances being$<$ 32×32 pixel lesions under large-view UAV images. CARAFEs 5×5 kernel enlarges the receptive field and smooths faint rust edges, causing 4.8% FN. Besides, the fixed 3×3 Gaussian window in FLM treats low-contrast rust margins as uncertain pixels and suppresses them, introducing another 2.1% FN. Tuning CARAFE kernel to 3×3 and making daptive in FLM recovered Recall to 76.4% (see Table 6-EXP), at the cost of 0.7% mAP reduction on other classes. Therefore, the reported 72.0% is a conservative value that keeps the best overall performance; readers should not interpret it as model instability.

The experimental results show that the improved boundary regression loss function makes the model more accurately locate and identify the slime, rust and plaque images, and P is also significantly improved. After adding attention mechanism module to the model, background interference can be suppressed, feature extraction ability can be improved, and R can be improved. By combining the two and improving the model at the same time, the P, R and mAP of the model can be significantly improved. It can be seen that FNNMFF model can more effectively detect maize crop pests and diseases. Through the comparison and analysis of the above experiments, it can be verified that the FNNMFF model in this study has the best detection effect on maize pests and diseases.

In this experiment, the proposed FNNMFF model in this paper is compared with FieldPlant [67], LIDL [68] and DLIC [69] under the same experimental conditions to further verify the effectiveness of FNNMFF in the detection of pests and diseases. Table 7 shows the experimental results of different models.

Table 7. Comparison with different methods

Model	P	R	F1	parameters	Floating-point arithmetic
FieldPlant	72.4%	70.2%	70.3%	21289813	3.672×10^9
LIDL	71.1%	67.4%	68.4%	23000405	4.258×10^9
DLIC	72.8%	69.0%	69.2%	4214853	2.251×10^8
FNNMFF	79.2%	74.3%	76.9%	1291963	1.53×10^8

As can be seen from Table 7, compared with the other three networks, the accuracy rate of the FNNMFF model is increased by 7.8%, 7.3%, 7.1% and 6.6% respectively, and the F1 value is increased by 6.6%, 8.5%, 7.7% and 7.0% respectively. In terms of model performance, the number of FNNMFF model parameters is only 1.29×10^6, and the floating point computation is 1.53×10^8. The FNNMFF network model is much smaller than other network models. It can be seen that the FNNMFF network model is far superior to other models in terms of recognition accuracy and model performance, indicating that the FNNMFF network model has certain advantages in the identification of pests and diseases.

In here, we add the sensitivity analysis experiments. The results are shown in Table 8.

When $\sigma = 1.0$, the overall performance and boundary IoU reach local optimality. The adaptive σ further increases the boundary IoU to 85.3% (due to dynamic matching of pixel contrast to avoid excessive suppression of low-contrast boundaries). The 3×3 window is more suitable for small lesion boundaries (such as rust lesions smaller than

Table 8. Results of Sensitivity Analysis for Key Parameters of FLM/%

Setting	P	R	mAP	IoU	FN
σ=0.5	93.8	82.1	89.5	78.3	8.7
σ=1.0	94.7	83.5	90.2	82.6	6.2
σ=1.5	94.5	82.8	89.9	80.1	7.5
σ=2.0	93.2	81.9	89.1	76.4	9.3
σ=2.5	92.6	80.7	88.6	73.2	11.5
Adaptive σ	94.9	84.2	90.5	85.3	4.8
Window size= 3×3	94.7	83.5	90.2	82.6	6.2
Window size= 5×5	94.3	82.9	89.8	81.2	7.1

32×32 pixels), while the 5×5 window causes loss of small boundary details due to its overly large receptive field.

We conduct 5 sets of comparative ablation experiments. Ablation-1 represents the removal of the Gaussian membership function (replaced by a simple weighted average). Ablation-2 represents a fixed σ value of 1.0 with the adaptive mechanism removed. Ablation-3 represents the removal of the ambiguous semantic labels while retaining the noise suppression function. Ablation-4 indicates the complete removal of FLM with the original jump connection retained. The experimental results are shown in Table 9. MPC is the accuracy of mixed pixel classification.

Table 9. Font sizes

Module	P	R	mAP	IoU	MPC
FNNMFF	94.7	83.5	90.2	82.6	89.3
Ablation-1	92.1	80.3	87.5	74.2	78.6
Ablation-2	94.1	82.7	89.5	79.8	85.1
Ablation-3	93.5	81.9	88.8	77.5	82.4
Ablation-4	91.8	80.1	87.1	72.3	76.9

The complete FLM has the best performance in all indicators, demonstrating the synergy of the fuzzy rules (Gaussian membership function), adaptive parameters, and semantic labels. The performance of Ablation-1 declines the most (mAP 2.7%, MPC 10.7%), indicating that the Gaussian membership function is the core of FLM by distributing fuzzy weights to suppress the category ambiguity of mixed pixels. The adaptive Ablation-2 vs complete FLM) increases the boundary IoU by 2.8%, proving that it can dynamically adapt to different contrast boundary regions. The fuzzy semantic labels (Ablation-3 vs complete FLM) increases the accuracy of mixed pixels by 6.9%, showing that it reduces cross-category confusion through membership degree allocation.

Table 10 shows the lightweight performance comparison with mainstream tiny models.

As shown in Table 10, FNNMFF achieves significantly superior detection performance with only 1.29M parameters and 0.153 GFLOPs, which is 32% fewer parameters and 96% less computation than YOLOv5n, while mAP is improved by 14.4%. This

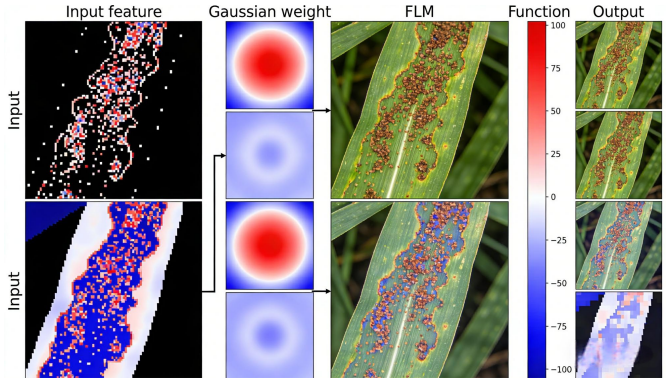


Fig. 9. FLM Gaussian membership function weight distribution heatmap (Select the boundary area of the rust lesion (with dense mixed pixels), and display the input feature map, the Gaussian weight heat map of FLM (red = high weight, blue = low weight), and the output feature map.)

Table 10. Lightweight performance comparison with mainstream tiny models

Model	Params (M)	FLOPs (G)	mAP	F1
YOLOv5n	1.90	4.50	75.8	73.2
MobileNetV3-YOLO	2.10	3.90	74.6	72.1
FNNMFF (Ours)	1.29	0.153	90.2	76.9

demonstrates that FNNMFF is not only lighter but also more accurate, making it highly suitable for real-time mobile deployment in resource-constrained agricultural scenarios.

5. Conclusion

In this study, the deep learning object detection technology is applied to the detection of crop pests and diseases, which has great practical significance for the control of citrus orchards, precise application of medicine and guarantee of citrus yield. In order to improve the detection accuracy of multi-species and multi-target pests and diseases in the complex background of natural orchards, an improved detection algorithm based on YOLOv5s is proposed. In this method, CXV2 module is used to replace the C3 module of the original model, and ConvNeXtV2 module is introduced into the module to enhance the feature extraction capability. Dynamic detection head is used to replace the original detection head to enhance the perception of different spatial scales and different task targets. Lightweight up-sampling operator CARAFE is used to replace the original up-sampling method and strengthen the up-sampling work. The research results show that compared with other commonly used algorithms, the model has better detection effect and stronger robustness. The presented method has achieved an improvement of 1.9% over the average mAP of the backbone network., which proves the effectiveness of the model, which can meet the detection requirements in complex environments. Due to the small number of parameters, it is easy to implement fast detection and mobile terminal deployment.

Nevertheless, this study has two main limitations. First, the FNNMFF model was trained and evaluated on single-temporal UAV and satellite imagery; it has not been verified with UAV multispectral time-series data that could reveal phenological dynamics under recurrent pest stress. Second, the current architecture processes each image independently and lacks an explicit temporal branch. Therefore, in our next work, we will embed a lightweight Transformer-based time-series module to capture long-range phenological dependencies and further enhance early-stage pest warning accuracy. These extensions will be reported in a follow-up study.

This study is conducted on a computer device and has not been deployed to a mobile device for field testing, so there may be some errors in the actual environment. In the follow-up research work, we first will use Sentinel-2, unmanned aerial vehicle multispectral and meteorological data to establish a 50k annotation database in different regions. Then, we will develop a lightweight Transformer model that integrates temporal and spatial attention, achieving an F1 score$\geq 90\%$ at a 10m resolution. Finally, we will expand multi-hazard and multi-scale assimilation, generate regional risk maps, and support precise prevention and control.

Acknowledgments. This research was funded by Liaoning Provincial Science and Technology Plan Project (2024JH2/102600106), the National Natural Science Foundation of China (grant number 62071084). This research was also funded by Liaoning Provincial Education Department Project, Project Number: LJ232510166008, Project Name: Research on Data-driven Embodied Intelligent Meta-skills Learning and Generalized Sensory Decision-making.

References

1. Y. Shi, K. Zhou, M. Zhou, S. Li and W. Liu, "Temporal Knowledge Graph for Food Risk Prediction," in IEEE Transactions on Artificial Intelligence, vol. 5, no. 5, pp. 2217-2226, May 2024, doi: 10.1109/TAI.2023.3321590.
2. R. Rakholia, A. L. Surez-Cetrulo, M. Singh and R. S. Carbajo, "AI-Driven Meat Food Drying Time Prediction for Resource Optimization and Production Planning in Smart Manufacturing," in IEEE Access, vol. 13, pp. 22420-22428, 2025, doi: 10.1109/ACCESS.2025.3534918.
3. S. Abbate, P. Centobelli, L. Fraccascia and G. Orzes, "Unveiling the Potential of Industry 4.0 for a Sustainable Agri-Food Future," in IEEE Engineering Management Review, doi: 10.1109/EMR.2025.3570903.
4. Su F, Liu Y, Chen S J, et al. Towards the impact of economic policy uncertainty on food security: Introducing a comprehensive heterogeneous framework for assessment, Journal of Cleaner Production, vol. 386, pp. 135792, 2023.
5. Abdallah A H, Ayamga M, Awuni J A. Impact of land grabbing on food security: Evidence from Ghana, Environment, Development and Sustainability, vol. 25, no. 7, pp. 6071-6094, 2023.
6. Abbas A, Mubeen M, Younus W, et al. Plant Diseases and Pests, Growing Threats to Food Security of Gilgit-Baltistan, Pakistan, Sarhad Journal of Agriculture, vol. 39, no. 2, 2023.
7. H. Zhou, K. Dai, N. Wen, J. Xiang and Z. Li, "The Influence of Topography-Dependent Atmospheric Delay for the InSAR Time-Series Results and the Deep Neural Network Correction," in IEEE Geoscience and Remote Sensing Letters, vol. 21, pp. 1-5, 2024, Art no. 4007805, doi: 10.1109/LGRS.2024.3379982.
8. W. Seo and W. Pak, "Real-Time Network Intrusion Prevention System Based on Hybrid Machine Learning," in IEEE Access, vol. 9, pp. 46386-46397, 2021, doi: 10.1109/ACCESS.2021.3066620.

9. K. Zhang, P. Chen, T. Ma and S. Gao, "On-Demand Precise Tracking for Energy-Constrained UAVs in Underground Coal Mines," in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-14, 2022, Art no. 5500814, doi: 10.1109/TIM.2022.3146925.
10. Cheng P, Tang H, Lin F, et al. Bibliometrics of the nexus between food security and carbon emissions: hotspots and trends, Environmental Science and Pollution Research, vol. 30, no. 10, pp. 25981-25998, 2023.
11. Khatri P, Kumar P, Shakya K S, et al. Understanding the intertwined nature of rising multiple risks in modern agriculture and food system, Environment, Development and Sustainability, pp. 1-44, 2023.
12. H. Goddard and L. Shamir, "SVMnet: Non-Parametric Image Classification Based on Convolutional Ensembles of Support Vector Machines for Small Training Sets," in IEEE Access, vol. 10, pp. 24029-24038, 2022, doi: 10.1109/ACCESS.2022.3154405.
13. A. Miroszewski et al., "Detecting Clouds in Multispectral Satellite Images Using Quantum-Kernel Support Vector Machines," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 7601-7613, 2023, doi: 10.1109/JSTARS.2023.3304122.
14. A. Islam and S. B. Belhaouari, "Fast and Efficient Image Generation Using Variational Autoencoders and K-Nearest Neighbor OveRsampling Approach," in IEEE Access, vol. 11, pp. 28416-28426, 2023, doi: 10.1109/ACCESS.2023.3259236.
15. Yin, S., Li, H. Teng, L. Airport Detection Based on Improved Faster RCNN in Large Scale Remote Sensing Images, Sensing and Imaging, vol. 21, 2020. https://doi.org/10.1007/s11220-020-00314-2.
16. S. Yin and H. Li. Hot Region Selection Based on Selective Search and Modified Fuzzy C-Means in Remote Sensing Images, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 5862-5871, 2020, doi: 10.1109/JSTARS.2020.3025582.
17. Linyi W, Jing B, Wenjing L, et al. Research Progress of YOLO Series Target Detection Algorithms, Journal of Computer Engineering & Applications, vol. 59, no. 14, 2023.
18. Lawal O M. Real-time cucurbit fruit detection in greenhouse using improved YOLO series algorithm, Precision Agriculture, vol. 25, no. 1, pp. 347-359, 2024.
19. Li J, Zhu Z, Liu H, et al. Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R-CNN, Ecological Informatics, vol. 77, 2023.
20. Cheng X, Wang Z, Song C, et al. FFR-SSD: feature fusion and reconstruction single shot detector for multi-scale object detection, Signal, Image and Video Processing, vol. 17, no. 6, pp. 3145-3153, 2023.
21. Faisal S, Javed K, Ali S, et al. Deep Transfer Learning Based Detection and Classification of Citrus Plant Diseases, Computers, Materials and Continua, vol. 76, no. 1, pp. 895-914, 2023.
22. Ma L, Yu Q, Yu H, et al. Maize leaf disease identification based on yolov5n algorithm incorporating attention mechanism, Agronomy, vol. 13, no. 2, pp. 521, 2023.
23. Zhu S, Ma W, Wang J, et al. EADD-YOLO: An efficient and accurate disease detector for apple leaf using improved lightweight YOLOv5, Frontiers in Plant Science, vol. 14, 2023.
24. Majeed S, Ahmad M, Ozdemir F A, et al. Micromorphological characterization of seeds of dicot angiosperms from the Thal desert (Pakistan), Plant Biosystems-An International Journal Dealing with all Aspects of Plant Biology, vol. 157, no. 2, pp. 392-418, 2023.
25. Fan X, Chai X, Zhou J, et al. Deep learning based weed detection and target spraying robot system at seedling stage of cotton field, Computers and Electronics in Agriculture, vol. 214: 108317, 2023.
26. X. Zhang, Z. Mao, N. Chimitt and S. H. Chan, "Imaging Through the Atmosphere Using Turbulence Mitigation Transformer," in IEEE Transactions on Computational Imaging, vol. 10, pp. 115-128, 2024, doi: 10.1109/TCI.2024.3354421.
27. Yang H, Lin D, Zhang G, et al. Research on Detection of Rice Pests and Diseases Based on Improved yolov5 Algorithm, Applied Sciences, vol. 13, no. 18, pp. 10188, 2023.

28. Guan H, Deng H, Ma X, et al. A corn canopy organs detection method based on improved DBi-YOLOv8 network, European Journal of Agronomy, vol. 154, 2024.
29. Yin S, Wang L, Teng L. Threshold segmentation based on information fusion for object shadow detection in remote sensing images, Computer Science and Information Systems, 2024. doi: 10.2298/CSIS231230023Y.
30. Bifarin O O. Interpretable machine learning with tree-based shapley additive explanations: Application to metabolomics datasets for binary classification[J]. Plos one, 2023, 18(5): e0284315.
31. Gu Y, Wang M, Gong Y, et al. Unveiling breast cancer risk profiles: a survival clustering analysis empowered by an online web application[J]. Future Oncology, 2023, 19(40): 2651-2667.
32. Ni W, Pu T, Liu S. Analysis and identification of glass products based on fisher linear discriminant analysis and logistic regression for dichotomous classification[J]. Highlights in Science, Engineering and Technology, 2023, 42: 103-110.
33. S. Yin, H. Li, Y. Sun, M. Ibrar, and L. Teng. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactionss on Science and Engineering, vol. 2, no. 2, pp. 13-20, 2024.
34. dos Santos G L A A, Reis A S, Besen M R, et al. Spectral method for macro and micronutrient prediction in soybean leaves using interval partial least squares regression[J]. European Journal of Agronomy, 2023, 143: 126717.
35. J. Deng et al., "Quantitative Estimation of Wheat Stripe Rust Disease Index Using Unmanned Aerial Vehicle Hyperspectral Imagery and Innovative Vegetation Indices," in IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1-11, 2023, Art no. 4406111, doi: 10.1109/TGRS.2023.3292130.
36. Lin F, Li B, Zhou R, et al. Early Detection of Rice Sheath Blight Using Hyperspectral Remote Sensing[J]. Remote Sensing, 2024, 16(12): 2047.
37. Chattopadhyay N, Shukla K K, Birah A, et al. Identification of Spectral Bands to Discriminate Wheat Spot Blotch using in Situ Hyperspectral Data[J]. Journal of the Indian Society of Remote Sensing, 2023, 51(5): 917-934.
38. Kolipaka V R R, Namburu A. An automatic crop yield prediction framework designed with two-stage classifiers: a meta-heuristic approach[J]. Multimedia Tools and Applications, 2024, 83(10): 28969-28992.
39. X. Li et al., "A Study on the Estimation Model of Hyperspectral Reflectivity and Leaf Nitrogen Content of Cotton Leaves," in IEEE Access, vol. 11, pp. 74228-74238, 2023, doi: 10.1109/ACCESS.2023.3296635.
40. L. Teng et al., "FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1529-1542, June 2023, doi: 10.1109/TNSM.2023.3273991.
41. Tran M D, Vantrepotte V, Loisel H, et al. Band ratios combination for estimating Chlorophyll-a from Sentinel-2 and Sentinel-3 in Coastal Waters[J]. Remote Sensing, 2023, 15(6): 1653.
42. Kumar D, Kukreja V. Image segmentation, classification, and recognition methods for wheat diseases: Two Decades systematic literature review[J]. Computers and Electronics in Agriculture, 2024, 221: 109005.
43. Mumtaz R, Maqsood M H, ul Haq I, et al. Integrated digital image processing techniques and deep learning approaches for wheat stripe rust disease detection and grading[J]. Decision Analytics Journal, 2023, 8: 100305.
44. Maraveas C, Asteris P G, Arvanitis K G, et al. Application of bio and nature-inspired algorithms in agricultural engineering[J]. Archives of Computational Methods in Engineering, 2023, 30(3): 1979-2012.
45. M. Masood et al., "MaizeNet: A Deep Learning Approach for Effective Recognition of Maize Plant Leaf Diseases," in IEEE Access, vol. 11, pp. 52862-52876, 2023, doi: 10.1109/ACCESS.2023.3280260.

46. Siddiqa A, Islam R, Afjal M I. Spectral segmentation based dimension reduction for hyperspectral image classification[J]. Journal of Spatial Science, 2023, 68(4): 543-562.
47. Chodey M D, Shariff C N. Pest detection via hybrid classification model with fuzzy C-means segmentation and proposed texture feature[J]. Biomedical Signal Processing and Control, 2023, 84: 104710.
48. Gupta V A, Padmavati M V, Saxena R R, et al. A novel insect and pest identification model based on a weighted multipath convolutional neural network and generative adversarial network[J]. Karbala International Journal of Modern Science, 2023, 9(1): 14.
49. Chen X, Guan Y, Li G. An Spodoptera Frugiperda pest detection algorithm based on improved feature fusion[C]//International Conference on Electronic Information Engineering and Computer Science (EIECS 2022). SPIE, 2023, 12602: 937-943.
50. Khanna M, Singh L K, Thawkar S, et al. PlaNet: a robust deep convolutional neural network model for plant leaves disease recognition[J]. Multimedia Tools and Applications, 2024, 83(2): 4465-4517.
51. Guerrero-Ibaez A, Reyes-Munoz A. Monitoring tomato leaf disease through convolutional neural networks[J]. Electronics, 2023, 12(1): 229.
52. Imanulloh S B, Muslikh A R, Setiadi D R I M. Plant diseases classification based leaves image using convolutional neural network[J]. Journal of Computing Theories and Applications (JCTA), 2023, 1(1): 1-10.
53. Al-Timemy A H, Alzubaidi L, Mosa Z M, et al. A deep feature fusion of improved suspected keratoconus detection with deep learning[J]. Diagnostics, 2023, 13(10): 1689.
54. Sahasranamam V, Ramesh T, Muthumanickam D, et al. AI and Neural Network-Based Approach for Paddy Disease Identification and Classification[J]. International Research Journal of Multidisciplinary Technovation, 2024, 6(3): 101-111.
55. Zou M, Liu Y, Fu M, et al. Combining spectral and texture feature of UAV image with plant height to improve LAI estimation of winter wheat at jointing stage[J]. Frontiers in Plant Science, 2024, 14: 1272049.
56. Zhang K, Zhang K, Bao R. Prediction of gas explosion pressures: A machine learning algorithm based on KPCA and an optimized LSSVM[J]. Journal of Loss Prevention in the Process Industries, 2023, 83: 105082.
57. Irmak G, Saygl A. A Novel Approach for Tomato Leaf Disease Classification with Deep Convolutional Neural Networks[J]. Journal of Agricultural Sciences, 2024, 30(2): 367-385.
58. Roy S, Hore J, Sen P, et al. Hyperspectral Remote Sensing and its application in Pest and Disease management in Agriculture[J]. Indian Farmer, 2023, 10: 229-232.
59. Sun H, Song X, Guo W, et al. Potato late blight severity monitoring based on the relief-mRmR algorithm with dual-drone cooperation[J]. Computers and Electronics in Agriculture, 2023, 215: 108438.
60. Nayak A, Chakraborty S, Swain D K. Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection[J]. Smart Agricultural Technology, 2023, 4: 100195.
61. Nguyen C, Sagan V, Skobalski J, et al. Early detection of wheat yellow rust disease and its impact on terminal yield with multi-spectral UAV-imagery[J]. Remote Sensing, 2023, 15(13): 3301.
62. Kumar S, Sharma P, Kumar R. Assessing Agricultural Practices and Plant Protection Methods in Rupnagar District, Punjab, India[J]. Asian Journal of Agricultural and Horticultural Research, 2023, 10(4): 594-606.
63. Zhang K, Yan F, Liu P. The application of hyperspectral imaging for wheat biotic and abiotic stress analysis: A review[J]. Computers and Electronics in Agriculture, 2024, 221: 109008.
64. G. Yang, J. Lei, Z. Zhu, S. Cheng, Z. Feng and R. Liang, "AFPN: Asymptotic Feature Pyramid Network for Object Detection," 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, Oahu, HI, USA, 2023, pp. 2184-2189, doi: 10.1109/SMC53992.2023.10394415.

65. S. Yin, L. Wang, M. Shafiq, L. Teng, A. A. Laghari and M. F. Khan, "G2Grad-CAMRL: An Object Detection and Interpretation Model Based on Gradient-Weighted Class Activation Mapping and Reinforcement Learning in Remote Sensing Images," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 3583-3598, 2023, doi: 10.1109/JSTARS.2023.3241405.
66. Zhang H, Xu C, Zhang S. Inner-iou: more effective intersection over union loss with auxiliary bounding box[J]. arxiv preprint arxiv:2311.02877, 2023.
67. Moupojou E, Tagne A, Retraint F, et al. FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning[J]. IEEE Access, 2023, 11: 35398-35410.
68. Omer S M, Ghafoor K Z, Askar S K. Lightweight improved yolov5 model for cucumber leaf disease and pest detection based on deep learning[J]. Signal, Image and Video Processing, 2024, 18(2): 1329-1342.
69. Lee D I, Lee J H, Jang S H, et al. Crop Disease Diagnosis with Deep Learning-Based Image Captioning and Object Detection[J]. Applied Sciences, 2023, 13(5): 3148.

Shoulin Yin received the M.A. degree from Shenyang Normal University, Shenyang, China, in 2015. He is currently pursuing the Ph.D. degree with the College of Information and Communication Engineering, Harbin Engineering University, Harbin. His research interests are remote sensing image processing and object detection.

Hang Li received the B.S. degree from Dalian Fisheries University, in 1999, the M.S. degree from the Shenyang Institute of Technology, in 2002, and the Ph.D. degree from Northeastern University, China, in 2005.,In 2002, he joined Software College as a Teacher. He is a Full Professor and the Master Supervisor with Shenyang Normal University. He is an Outstanding Young Backbone Teacher with Liaoning General Institutions of Higher Learning (Liaoning Education Department 200612225). His research interests are image analysis and processing, big data, cloud computing. He won the Second Prize of National Defense Science and Technology Award of The Commission of Science, Technology and Industry for National Defense (National Defense Science, Technology and Industry Commission 2005GFJ2126-6) and the First Prize of Science and Technology Award of China North Industries Group corporation (2005-BQJ-1-0019-6).

Mirjana Ivanovic is a Full Professor at the Faculty of Sciences, University of Novi Sad, Serbia, since 2002, and a corresponding member of the Serbian Academy of Sciences and Arts since 2024. She is a member of the Board of Directors of the Institute for Artifcial Intelligence Research and Development of Serbia. Mirjana has authored or co-authored 17 textbooks, 30 edited proceedings, 4 monographs, and more than 540 research articles on multi-agent systems, e-learning and web-based learning, applications of intelligent techniques (CBR, data and web mining), software engineering education, most of which are published in international journals and proceedings of high-quality international conferences. She has served as a member of program committees for more than 500 international conferences and has chaired numerous international conferences as general chair and program committee chair. Additionally, she has been an invited speaker at numerous international conferences and a visiting lecturer in Australia, Thailand, and China. As a leader and researcher, she has participated in highly regarded international projects.

Tao Chen received the PHD. degree from Harbin Engineering University, Harbin. He is currently full professor with the College of Information and Communication Engineering, Harbin Engineering University, Harbin. His research interests are Broadband signal detection, processing and identification, radar signal sorting.

Lin Teng received the M.A. degree from Shenyang Normal University, Shenyang, China, in 2020. She is currently pursuing the Ph.D. degree with the College of Information and Communication Engineering, Harbin Engineering University, Harbin. Her research interests are image processing and semantic segmentation.

Received: November 09, 2025; Accepted: December 23, 2025.

Computer Science and Information Systems 23(1):397–417 https://doi.org/10.2298/CSIS250314079B

Mitigating Out-of-Vocabulary Challenges in Embedded devices Vulnerability Classification: An Ensemble Embedding Approach with Bidirectional Context Modeling ⋆

Aissa Ben Yahya, Hicham El Akhal, and Abdelbaki El Belrhiti El Alaoui

Computer Science and Applications Laboratory, Faculty of Sciences, Moulay Ismail University of Meknes, B 11201 Zitoune, 50000, Meknes, Morocco
{ai,hi}@edu.umi.ac.ma
a.elbelrhitielalaoui@umi.ac.ma

Abstract. Critical infrastructure is increasingly reliant on embedded systems, which are particularly vulnerable to cyberattacks due to their inherent complexity and interconnectivity. Accurate classification of vulnerabilities in these systems is essential for targeted analysis and mitigation strategies. While pre-trained word embeddings such as Word2Vec, GloVe, and FastText are commonly used for this purpose, their effectiveness is limited by reliance on training corpora that lack domain-specific terminology, leading to challenges with Out-of-Vocabulary words and reduced classification performance. To address this limitation, we propose a novel ensemble embedding technique that combines multiple pre-trained embeddings to improve vulnerability classification in embedded systems. Evaluated on benchmark datasets, including the National Vulnerability Database and the China National Vulnerability Database, our method achieves a 91.50% F1-score on unseen data, outperforming traditional single-embedding approaches. This advancement demonstrates significant potential for enhancing cybersecurity in critical infrastructure applications.

Keywords: ensemble embedding, word embedding, vulnerability classification, critical infrastructure devices security.

1. Introduction

The mass adoption of embedded systems in security-critical areas such as healthcare, transportation, and manufacturing automation requires the protection of such systems in order to make them secure. Owing to the inherently sensitive nature of the data being processed by these systems, coupled with distinctive operational specifications, such systems are highly vulnerable to security breaches. Consequences of such violations have the potential to extend significantly and may include service disruptions [12], improper access to sensitive data [16,4], potential equipment destruction [10], and, in the most severe cases, threatening human life [9].

⋆ This manuscript is an extended version of the paper "Enhanced Classification of Embedded System Vulnerabilities Using Ensemble Embedding and BiLSTM Networks" presented at the 28th International Symposium, IDEAS 2024, Bayonne, France, August 26–29, 2024.

The inherent distinction between embedded system vulnerabilities (ESVs) and general-purpose system vulnerabilities (GPSVs) comes from the unique architectural and operational nature of embedded devices. Unlike servers or desktop computers, embedded systems feature severe restrictions in resources, including limited memory (RAM), processing power (CPU), and power budgets [7]. By virtue of these restrictions, traditional security strategies, such as resource-hungry antivirus software or frequent, large-scale updates[1] , become infeasible. Furthermore, these systems employ domain-specific communication protocols (e.g., CAN bus in the automotive sector, Modbus in industrial controls, and Zigbee in IoT), which often have no built-in, end-to-end security measures inherent in traditional internet protocols. This, in turn, produces quite dissimilar exploit profiles and threat models, where attackers may target physical interfaces[2] (such as JTAG or UART), firmware, or real-time operating systems (RTOS) in ways that are utterly irrelevant in a normal IT environment. This problem is further compounded by the long operational life and the infrequent update schedule of many embedded systems, meaning that weaknesses may persist for years or even decades after discovery. Because of these inherent differences, the current treatment of ESVs and GPSVs under one monolithic category falls short and poses a risk. Without a differentiated scheme of categorization, security professionals cannot accurately prioritize threats relevant to these systems in real-time for analysis. This research, to the best of our knowledge, is the very first to present an automated system of classification that carefully distinguishes ESVs and their GPSV equivalents so that this critically important infrastructure could be secured with much higher precision and efficacy.

Recent advancements in machine learning and deep learning have facilitated the development of automatic classification mechanisms, which have been successfully applied to the realm of vulnerability classification [18,8,6,3,14]. The current state-of-the-art in embedded systems vulnerability classification is represented by the work in [2], which employs Term Frequency Inverse Document Frequency (TF-IDF) to generate weighted word vectors and Support Vector Machines (SVM) as a classifier. However, this approach has limitations, including its reliance on TF-IDF, which overlooks the co-occurrence and similarity relationships between words. Additionally, it lacks a mechanism to handle out-of-vocabulary (OOV) words and relies on SVM, which can be sensitive to feature scaling and fails to capture contextual information, leading to poor generalization to new data. [19] extended this line of research by proposing an algorithm that classified vulnerabilities by associating Common Vulnerabilities and Exposures (CVE) entries with their corresponding CWE-ID. Furthermore, they introduced an improved TF-IDF in [20] that weights words based on their category. Although this algorithm achieved promising results, it has no mechanism of dealing with OOV words.

Traditional machine learning approaches, while effective in general text classification tasks, often face limitations when applied to more specialized contexts. These limitations arise from their reliance on pre-trained word embeddings, which are susceptible to the OOV problem [13]. When words that were not encountered during training appear, they are inadequately represented in the model, resulting in diminished classification performance.

[1] https://www.mottasec.com/article/faq-why-is-embedded-security-so-hard-to-get-right

[2] https://interrupt.memfault.com/blog/diving-into-jtag-part-6

To address these challenges, we introduce a new framework of ensemble embeddings that integrates multiple word embedding approaches, such as GloVe, Word2Vec, and FastText, to have enhanced word representations. This approach markedly minimizes the issues concerning OOV words. Our methodology implements an OOV word fallback approach where OOV word vectors are replaced with those of the most available semantically similar words. This supposes that semantically related words frequently coexist in similar contexts. This approach is implemented in a bidirectional gated recurrent unit (BiGRU) model that characterfully extracts contextual dependency. Also included in our research is the creation of a benchmark dataset that was synthesized from multiple databases, presenting a versatile and representative set of data that could function equally well both to train and test the model. Our research's contributions consist of:

- A new methodology to differentiate and categorize vulnerability reports so that those of embedded systems could be differentiated from general-purpose systems to allow concentrated research.
- A new ensemble embedding scheme that simultaneously uses several schemes of word embeddings with a dynamically determined threshold to reduce the effect of OOV words.
- A Benchmark data set, which is compiled from several sources, presents a diverse and representative collection of data that helps in real-world deployment assessment.

This paper is structured as follows: we provide an overview of relevant literature in section 2, followed by a descriptive account of our methodology in section 3. section 4 consists of the results of the assessment in reference to our proposed mechanism. section 5 concludes the paper with a summary of our primary findings and conclusions from our research.

2. Related Work

Software vulnerability classification has also seen tremendous breakthroughs with the use of ML and DL methods. Current text representation methods have been largely grouped into two classes: (1) conventional feature engineering approaches, and (2) encoding with the help of word embedding schemes. Below, we critically analyze these methods, highlight their limitations, and explain how they motivate our proposed approach.

2.1. Traditional Feature Engineering Methods

Earlier works concentrated almost entirely on term-frequency-based text representation and dimension reduction methods. Specifically, [21] employed a document-term matrix (DTM) alongside chi-square feature selection to treat high-dimensional data. Likewise, [15] advanced a multi-label prediction model that used DTM-based word representation and weighting to attribute vulnerability characteristics. Although these approaches perform well in dealing with simple term distributions, they lack semantic relationships or contextual importance of words, hence yielding suboptimal performance when applied in texts with specialized domains. Likewise, [6] integrated TF-IDF alongside information gain (IG) and deep neural networks (DNNs). However, TF-IDF ignores semantic and local contextual knowledge, and IG focuses on worldwide discriminative features at the

expense of specialized ones in the given domains. Such drawbacks impair them in separating fine-grained categories of vulnerabilities. Extending this paradigm, [1] presented ThreatZoom, a framework that associates Common Vulnerabilities and Exposures (CVEs) with Common Weakness Enumerations (CWEs) based on n-gram features with TF-IDF weights. Though local sequences of words have been preserved with n-grams, the scheme borrows all the inherent flaws in the TF-IDF weight assignment scheme. Emphasizing the weaknesses of global term weights in more detail, [2] used Support Vector Machines (SVM) with TF-IDF to extract embedded system vulnerabilities. Although TF-IDF successfully downweights popular terms (e.g., "vulnerability" or "system"), its failure to integrate word co-occurrences and similarity lowers precision in multi-domains. This necessitates a process that inserts semantic and contextual knowledge into the word vectors, overcoming traditional weaknesses.

2.2. Word Embedding Techniques

To overcome the semantic shortcomings of TF-IDF, alternative strategies have been investigated by several studies. For example, [14] incorporated pre-trained GloVe embeddings and Convolutional Neural Networks (CNNs). Their approach, however, suffers from OOV words since GloVe has been pretrained on a static corpus and lacks adaptability to adapt to the emerging new terms. Likewise, [5] incorporated pre-trained Word2Vec embeddings and CNNs to learn sentence contexts. Although this model enhanced semantic understanding, it still had issues with OOV words since it also uses static pre-trained embeddings. Taking it a step further, [17] proposed a self-attention-based Deep Neural Network (SA-DNN) that seeks to emphasize important words in vulnerability descriptions. Nonetheless, self-attention mechanisms are prone to noise, which results in the model emphasizing background or irrelevant words, hence suffering in terms of robustness. Despite these efforts that have gone into building vulnerability classifiers, these works have been constrained by the failure to adopt a good OOV handling scheme, hence limiting software vulnerability classification. With these weaknesses in mind, a more adaptive and inclusive solution that minimizes OOV word influence and achieves contextual and semantic understanding on a good note becomes important.

2.3. Research Gaps and Motivation

Classification approaches suffer severely from the weaknesses in applying word embedding approaches, which in turn also affect solving the OOV word problem. There always remain two fundamental issues: (1) the inefficiency of relying on a single pre-trained approach to capturing the subtle relationship between terms in specialized fields, and (2) the incapability of the existing pre-trained embeddings, such as Word2Vec, GloVe, or FastText, to represent rare appearing or new vulnerability-related terms and thereby suffering from OOV problems. Such weaknesses point to the urgency in developing a new solution that both efficiently deals with OOV words and incorporates both co-occurring and contextual knowledge to attain superior vulnerability classification performance. Our proposed solution exploits these issues by adopting a new ensemble embedding approach in conjunction with a BiGRU model. By combining multiple word embedding models, such as GloVe, Word2Vec, and FastText, our solution produces more descriptive word

representations. This not only reduces OOV-related issues but also augments contextual knowledge and hence produces a stronger and more accurate vulnerability classification solution.

3. The proposed approach

The overall method used to create the embedded system vulnerability classification model is shown in Figure 1. The process has three main phases.

First, in the data collection and preprocessing phase, we gathered a raw dataset from various publicly available sources, including NVD, CNNVD, and the Zero Day Initiative (ZDI). We extracted relevant vulnerability descriptions with a script. Next, we balanced and preprocessed the data to reduce bias and ensure the text was formatted correctly for the model. Then, during the data split and representation phase, we divided the processed data into training, validation, and benchmark sets. To make sure our experiments reflected real-world challenges, we constructed the benchmark set using data unique to each source database and included manually curated, difficult examples. After the split, we applied word embedding to represent the textual data in numerical form. Finally, the Model architecture was established to process the training and validation data. The architecture begins with an Embedding layer, followed by a stack of three Bidirectional LSTM (BiLSTM) layers designed to extract rich contextual information from the descriptions. These are succeeded by two Dense layers. Notably, both the BiLSTM and Dense blocks incorporate Dropout and Batch Normalization to stabilize training and prevent overfitting. The pipeline concludes with a Softmax layer to generate prediction probabilities for the Output classes. The reserved benchmark data is then used to evaluate the performance of the finalized Model.

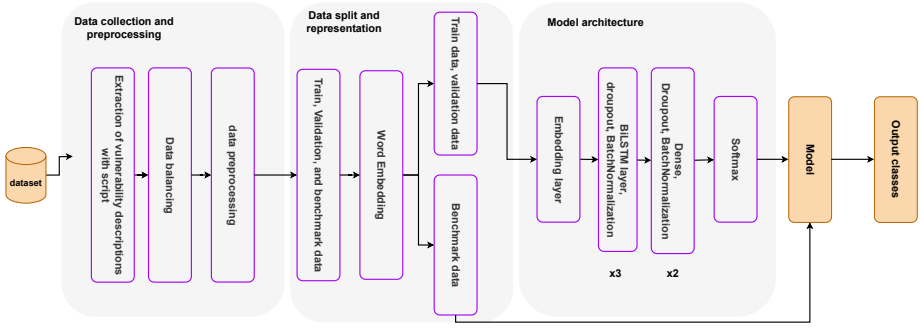


Fig. 1. Main method

3.1. Ensemble Embedding Technique

In natural language processing (NLP), word embedding techniques (WE) such as Word2Vec, GloVe, and FastText are widely used to map words into dense vector representations, cap-

turing syntactic and semantic relationships. However, a critical limitation of these methods is their susceptibility to OOV words, which are typically assigned null vectors if absent from the training corpus. This issue is particularly severe in domain-specific classification tasks, such as software vulnerability classification, where rare or emerging terminology is important for better classification performance. OOV words degrade model generalizability and robustness, as their misrepresentation introduces noise into tasks like classification.

To tackle this challenge, we suggest an ensemble embedding method that combines multiple word embeddings to lessen the impact of out-of-vocabulary words on performance. Our approach is based on two main ideas: (1) Different word embedding techniques trained on various datasets will capture more words, and (2) Similar words in the vector space will probably have similar meanings and functions in the text. Unlike typical methods that depend on a single source of word embeddings, our strategy uses a hierarchical structure. A primary word embedding acts as the main representation, while secondary embeddings provide backup options.

To illustrate the limitations of single WE approaches, consider the CVE in Figure 2 describing a vulnerability in the IEEE 802.15.4 Zigbee protocol. Critical terms such as *802.15.4, Zigbee, coordinator*, and *PHY* are domain-specific and absent in Word2Vec, as shown in Table 1. Traditional methods relying on a single WE would assign a zero vector to these OOV words, degrading classification accuracy.

Our proposed ensemble embedding technique implements the fallback mechanism to reduce the effect of these OOV words. The steps are as follows:

Our proposed technique implements an intelligent fallback mechanism. When a word is not found in the primary embedding, the algorithm searches for it in the secondary embeddings. If found, it retrieves the top-10 most similar words from that secondary model. However, simply choosing the first available substitute is not optimal. To ensure only the most relevant substitutions are made, we introduce a dynamic similarity threshold. For example, if the OOV word `Zigbee` is found in GloVe or FastText, our algorithm analyzes its top-10 neighbors as shown in Table 2, calculates an appropriate threshold (see subsection 3.2) for this specific word, and selects the best candidate. In our example, the dynamic threshold for the GloVe neighborhood of ”zigbee” is approximately 0.386; therefore, the following words have scores above this threshold and exist in the primary (Word2Vec) embedding: `hdcp (Score: 0.44)` and `aminoglycosides (Score: 0.41)`, the one with higher similarity score and exist in primary WE will be selected and use as a substitute for the missing word.

CVE-2024-3043

An unauthenticated IEEE 802.15.4 'co-ordinator realignment' packet can be used to force Zigbee nodes to change their network identifier (pan ID), leading to a denial of service. This packet type is not useful in production and should be used only for PHY qualification.

Fig. 2. A CVE describing a vulnerability in the IEEE 802.15.4 Zigbee protocol

Table 1. Comparison of embedded devices vulnerability word Coverage

Words	Word2vec	Glove	Fasttext
unauthenticated	✓	✓	✓
IEEE	✓	✓	✓
802.15.4	✗	✓	✓
co-ordinator	✗	✓	✓
realignment	✓	✓	✓
Zigbee	✗	✓	✓
nodes	✓	✓	✓
identifier	✓	✓	✓
denial	✓	✓	✓
packet	✓	✓	✓
production	✓	✓	✓
PHY	✗	✓	✓
qualification	✓	✓	✓

Table 2. Top similar words to *zigbee* in secondary embeddings (GloVe and FastText). Substitutes marked with (✓) exist in the primary embedding (Word2Vec).

GloVe			**FastText**		
Word	**Similarity**	**In Primary**	**Word**	**Similarity**	**In Primary**
802.15.4	0.58	✗	Zigbee	0.82	✗
z-wave	0.47	✗	Z-Wave	0.68	✗
hdcp	0.44	✓	ZigBee	0.66	✗
homeplug	0.44	✗	**DeviceNet**	0.65	✓
upnp	0.42	✗	1-Wire	0.63	✗
wml	0.41	✗	**Modbus**	0.63	✓
nmea	0.41	✗	**HomePlug**	0.62	✓
aminoglycosides	0.41	✓	**ecobee**	0.61	✓
passivhaus	0.40	✗	AllJoyn	0.61	✗
displayport	0.40	✗	**BACnet**	0.60	✓

3.2. Dynamic Threshold Calculation

A key innovation of our approach is adapting a dynamic threshold that is based on the local statistics of a word's semantic neighborhood. The process is as follows:

1. For an OOV word found in a secondary embedding, retrieve the top-10 most similar words and their cosine similarity scores.
2. Calculate the mean (μ) and standard deviation (σ) of these 10 scores. The mean represents the average similarity of the neighborhood, while the standard deviation measures its dispersion.

3. The dynamic threshold for this specific word is then set to $T = \mu - \sigma$. Where T is the threshold.

This data-driven threshold is strict for words with tight, high-similarity neighborhoods and more forgiving for words with looser neighborhoods. This ensures that substitutions are always contextually appropriate, maximizing the relevance of the fallback mechanism. The algorithm 1 formalizes the complete procedure.

4. Evaluation

4.1. Data collection

The dataset utilized in this research was assembled from various esteemed sources of vulnerability reports, which include the National Vulnerability Database (NVD) [3], the Chinese National Vulnerability Database (CNNVD) [4], the China National Vulnerability Database (CNVD) [5], VarIoT [6], the Cyber Emergency Response Team (ICS-CERT) [7], Fortiguard [8], and the Zero Day Initiative (ZDI) [9]. The aforementioned sources contributed a heterogeneous and comprehensive array of software vulnerability reports, thus facilitating the development of a robust dataset [10] aimed at the classification of vulnerabilities in embedded systems.

In preparing to get the NVD data, we downloaded JSON files from the NVD feeds [11]. We, however, excluded those entries with a flag that had been set to RESERVED, REJECTED, or DISPUTED since they are unconfirmed vulnerabilities and could not form part of training our models.

We extracted data in XML format from both CNNVD's and CNVD's official websites. Since the descriptions and corresponding info were in Chinese, all entries were translated into English through the Google Translate API [12] to ensure consistency in our dataset.

Due to the absence of a direct download feature in VarIoT, we created a web scraper utilizing Python to retrieve the required data. Meanwhile, data obtained from ICS-CERT, FortiGuard, and ZDI were used exclusively for testing and were excluded from our training dataset. In this study, these external sources are referred to collectively as "Unseen." To ensure the maintenance of high standards of data quality and relevance, we undertook a manual curation of their data to ensure its alignment with our research objectives.

A comprehensive overview of the collected dataset is presented in Table 3.

4.2. Data preprocessing

To prepare the data for modeling, we applied the following preprocessing steps:

[3] https://nvd.nist.gov/
[4] https://www.cnnvd.org.cn/
[5] https://www.cnvd.org.cn/
[6] https://www.variotdbs.pl/vulns/
[7] https://www.cisa.gov/
[8] https://www.fortiguard.com/
[9] https://www.zerodayinitiative.com/
[10] https://www.kaggle.com/datasets/aissaultimate/ensemble-embedding-data
[11] https://nvd.nist.gov/vuln/data-feeds
[12] https://pypi.org/project/googletrans/

Algorithm 1 Ensemble Embedding Approach

Input: Set of word embeddings models $E = \{E_1, E_2, \ldots, E_n\}$, where E_1 is the primary embedding model.
Input: Vocabulary V of words to be embedded.
Input: Vector dimension d.
Output: Word vector matrix M of size $|V| \times d$.
for each word $w \in V$ **do**
    $is_assigned \leftarrow$ false
    ▷ **Step 1: Primary Embedding Search**
    if $w \in E_1$ **then**
        $v_w \leftarrow E_1(w)$
        $is_assigned \leftarrow$ true
    end if
    if not $is_assigned$ **then**
        ▷ **Step 2: Secondary Embedding Fallback**
        for each secondary embedding $E_i \in \{E_2, \ldots, E_n\}$ **do**
            if $w \in E_i$ **then**
                Let S be the list of (word, score) pairs from the top-10 most similar words to w in E_i.
                if S is not empty **then**
                    Let $Scores$ be the list of scores from S.
                    $\mu \leftarrow$ mean$(Scores)$
                    $\sigma \leftarrow$ standard_deviation$(Scores)$
                    $dynamic_threshold \leftarrow \mu - \sigma$
                    for each $(s_j, score_j) \in S$ **do**
                        if $score_j \geq dynamic_threshold$ and $s_j \in E_1$ **then**
                            $v_w \leftarrow E_1(s_j)$
                            $is_assigned \leftarrow$ true
                            break ▷ Break from inner loop (substitutes)
                        end if
                    end for
                end if
            end if
            if $is_assigned$ **then**
                break ▷ Break from outer loop (embeddings)
            end if
        end for
    end if
    if not $is_assigned$ **then**
        ▷ **Step 3: Final Subword Fallback**
        ▷ Assumes E_1 is a subword-capable model like FastText.
        $v_w \leftarrow$ generate_subword_vector(E_1, w)
    end if
    Store v_w in matrix M.
end for
return Matrix M.

– **Lowercase conversion:** We converted all text to lowercase to reduce dimensionality and improve model performance.

Table 3. Data set collected from different databases

	ESV	**GPSV**
NVD	13515	32387
CNNVD	12026	27535
CNVD	7457	14364
VarIoT	6775	4586
ICS-CERT, ZDI, Fortiguard (Unseen)	68	100

- **Digit removal:** We removed all digits from the text to focus on the linguistic features of the data.
- **Contraction expansion:** We expanded contractions (e.g., ”you’re” to ”you are”, ”i’m” to ”i am”) to improve tokenization.
- **Stopword removal:** We removed common stopwords, such as ”the”, ”and”, etc., that do not add significant value to the text, to improve model accuracy.
- **Porter Stemming:** We applied Porter Stemming to reduce words to their base form, thereby reducing dimensionality and improving model performance.
- **Special character removal:** We removed special characters from the text to focus on the linguistic features of the data.
- **Base form conversion:** We converted words to their base form (e.g., ”ran” to ”run”) to further reduce dimensionality.

4.3. Data labeling

We utilized a Python script[13] to label the dataset using a two-tiered keyword-based methodology. This script operated with two distinct keyword sets: one specific to embedded systems and another for general-purpose systems. These keyword lists were meticulously compiled through an extensive review of online resources and generative large language models (LLMs), ensuring the inclusion of relevant terms associated with embedded devices, protocols, and software. The same process was applied to construct the keyword list for general-purpose systems. A sample of these keyword sets is provided in Table 4.

The script then processed each vulnerability description, scanning for the presence of keywords from both lists. If a description contained embedded system-related keywords while lacking general-purpose system keywords, it was classified as an embedded system vulnerability. Conversely, if general-purpose system keywords were detected without any embedded system keywords, the vulnerability was categorized accordingly. To reduce ambiguity during model training, entries containing keywords from both lists were omitted from the dataset.

This keyword-based classification strategy was inspired by prior research by Pap et al. [11], which utilized a similar methodology for vulnerability categorization. By adopting this approach, we ensured that our dataset was systematically labeled and optimally structured for model training.

[13] https://github.com/aissa302/CVE-Aanlysis-Project

Table 4. White and black list keywords

Embedded system terms	General-purpose terms
Asus RT	Huawei FusionServer
Huawei Quidway switches	Windows Desktop
Cisco 4000 Series	Apple OS X
Zhone GPON	Xen hypervisor
ZigBee	Google Earth
Modbus	Dell XPS 13
Embedded Linux	Skype
RTOS	HPE ProLiant
Android	Terraform

4.4. Benchmark Dataset Development

Many existing studies rely on the NVD for training and testing machine learning models. While this approach is convenient, it presents a major drawback: it does not accurately reflect real-world conditions. As a result, models trained solely on NVD data may overfit, achieving high accuracy on similar datasets but lacking robustness when applied to diverse or unseen data.

To overcome this limitation, models should be evaluated on datasets that differ in structure, content, and linguistic style, providing a more realistic measure of their generalizability. To support this need, we developed a comprehensive benchmark dataset incorporating vulnerability descriptions from CNNVD, CNVD, VarIoT, and a manually curated collection. These sources introduce greater diversity, ensuring that models are tested under conditions that closely resemble real-world challenges.

For dataset construction, we extracted vulnerability descriptions from each source and categorized them based on whether they had a CVE ID. To ensure uniqueness and prevent overlap with NVD, our benchmark dataset consists exclusively of entries without CVE IDs. This ensures that the dataset encompasses vulnerabilities not already documented in the NVD, providing a more rigorous evaluation framework.

Table 5 provides a summary of the benchmark dataset composition, detailing the number of vulnerability descriptions sourced from each database.

Table 5. Benchmark Dataset Composition

	ESV	GPSV
CNNVD	246	866
CNVD	1146	980
VarIoT	922	24
Unseen	68	100

4.5. Evaluation Metrics

To comprehensively evaluate the performance of our classification models, we employed a set of standard evaluation metrics. These metrics provide a holistic view of the models' effectiveness in classifying vulnerability descriptions.

– **Accuracy**: The proportion of correctly classified instances out of the total. It provides an overall measure of model performance.

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{Total Number of Instances}} \tag{1}$$

– **Precision**: The ratio of true positives to all predicted positives. Key for scenarios where false positives are costly.

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{2}$$

– **Recall**: The ratio of true positives to all actual positives. Important when missing a positive instance is critical.

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \tag{3}$$

– **F1-Score**: The harmonic mean of precision and recall, useful for imbalanced datasets.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \tag{4}$$

– **MCC**: Matthews Correlation Coefficient (MCC) accounts for all four quadrants of the confusion matrix and provides a balanced measure, even for datasets with imbalanced classes.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

– **AUC-ROC**: separate predictive accuracy from threshold-specific operating conditions, offering a general measure of how well the model distinguishes between classes.

4.6. Experimental setup

A series of experiments was carried out to determine the effectiveness of our proposed ensemble embedding approach for vulnerability classification. These experiments were executed on a computer equipped with double Xion CPUs running at 2.40GHz each and 62GB of RAM, complemented by NVIDIA GeForce RTX 3090 GPUs with 24GB of VRAM. The operating system used was Ubuntu 22.04 LTS. Throughout our experiments, we made use of Tensorflow version 2.12.1 with Keras version 2.12.0.

4.7. Hyperparameter Tuning and Model Configuration

Sentence length The length of vulnerability descriptions varies significantly depending on the level of detail required to document security flaws. To identify the optimal sequence length for classification, we evaluated input lengths of 100, 200, 300, and 400 tokens, measuring both performance metrics (accuracy) and computational efficiency (training time). As shown in Figure 3, model accuracy peaks at 200 tokens (90.06%), with shorter (100 tokens: 87.57%) or longer sequences (300 tokens: 86.95%, 400 tokens: 86.33%) yielding inferior results. This decline in performance beyond 200 tokens aligns with increased noise from excessive padding or truncation, which disrupts syntactic patterns critical for accurate classification. Conversely, sequences shorter than 200 tokens lack sufficient contextual detail to resolve ambiguities in vulnerability descriptions. Training time scales linearly with sequence length, increasing from 27.33 minutes (100 tokens) to 61.11 minutes (400 tokens). While the 200-token configuration requires 38.31 minutes (41% longer than 100 tokens), its 2.5-point accuracy gain justifies this trade-off.

These results demonstrate that 200 tokens optimally balance contextual retention, classification performance, and computational efficiency. This parameter was adopted for all subsequent experiments to ensure robustness without unnecessary resource overhead.

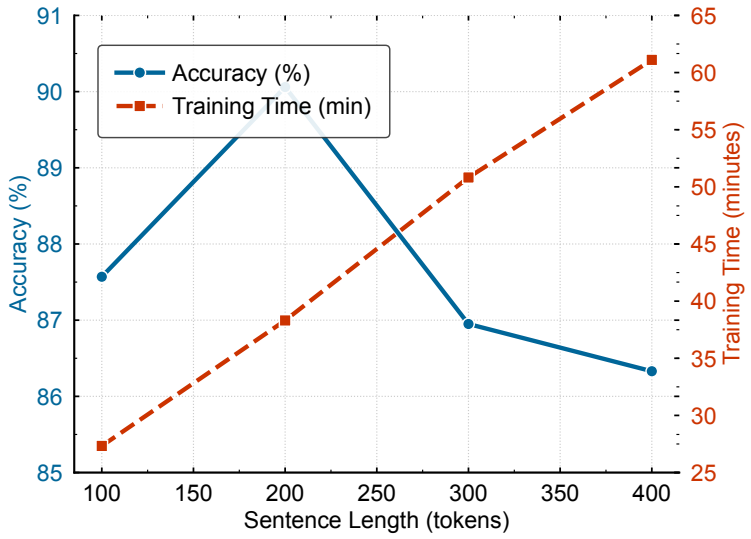


Fig. 3. Training accuracy and training time according to sentence lengths

Impact of Hidden Unit Size The dimensionality of the hidden state, or hidden size, in recurrent neural networks like BiLSTM and BiGRU is a fundamental hyperparameter that directly influences model capacity and performance. To identify the most effective configuration for our task, we systematically evaluated four distinct hidden sizes: 64, 128, 256,

and 512. The experiments, detailed in Figure 4, revealed a clear optimal point, demonstrating that model effectiveness is highly sensitive to this parameter. Our analysis indicates that a hidden size of 128 achieves the best overall performance. At this configuration, the model reaches its peak in several key metrics, including Accuracy (91.93%), F1-Score (91.50%), and Matthews Correlation Coefficient (MCC) (0.83). This result suggests that a hidden size of 128 provides an ideal balance, granting the model sufficient capacity to capture complex dependencies in the data without becoming overly complex and prone to overfitting. When the hidden size was reduced to 64, we observed a slight but consistent decrease in performance across the board. Metrics such as Accuracy (91.30%) and F1-Score (90.64%) were marginally lower than those at the optimal size of 128. This suggests that while still effective, a smaller hidden dimension may slightly limit the model's ability to learn the full complexity of the feature set, bordering on underfitting. Conversely, increasing the hidden size beyond the optimal point to 256 and 512 led to a significant degradation in model performance. For instance, the F1-Score dropped from its peak of 91.50% to 87.25% for a size of 256 and further to 87.10% for 512. A similar sharp decline was observed in the MCC, which fell from 0.83 to 0.75 and 0.74, respectively. Interestingly, while the AUC-ROC peaked at 95.12% for a size of 256, this improvement did not translate to better classification outcomes. This pattern strongly indicates that larger hidden sizes introduced excessive complexity, causing the model to overfit to the training data rather than learning generalizable patterns.

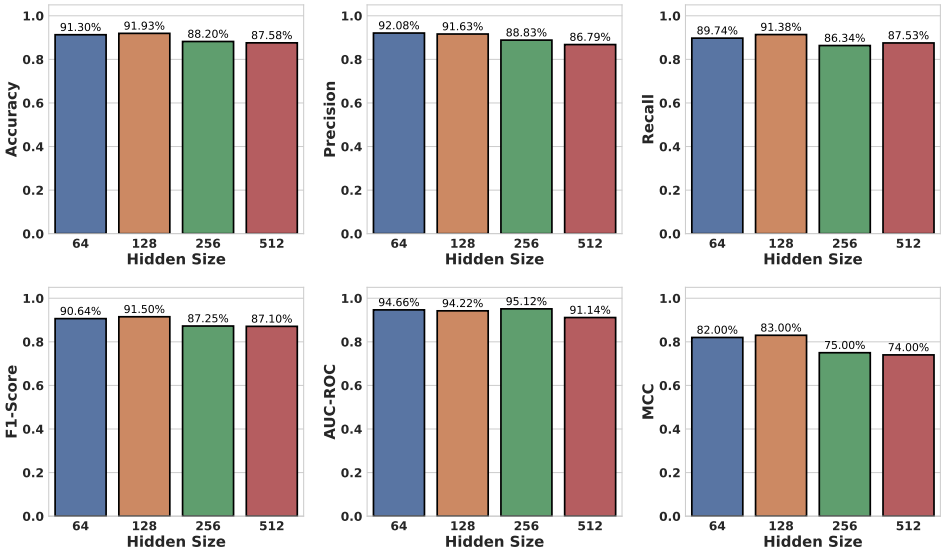


Fig. 4. Performance evaluation of ensemble embedding method across different hidden units sizes

Architectural, Optimization, and Regularization Choices Beyond the empirical tuning of sequence length and hidden unit size, several other hyperparameters were set based

on established best practices to ensure a stable, robust, and well-regularized model. Table 6 illustrates the different hyperparameters and their values. In addition to the hyperparameters in the table, batch normalization is used to avoid early overfitting of the models.

Table 6. Hyperparameters

Hyperparamters	**BiGRU**
Batch-size	32
Epochs	100
Layers	3
Droupout	0.3
Loss function	binary_crossentropy
Optimizer	Adam
Early Stop/patience	val_loss / 5
Kernel Initializer	he_uniform
Kernel Regularizer	l2(0.01)
Total params	17,816,522
Trainable params	1,411,610
Non-trainable params	16,404,912

We employed a stacked architecture of three BiGRU layers. Stacking recurrent layers allows the model to learn hierarchical representations of the input data; the initial layer captures more localized, syntactic patterns, while subsequent layers can integrate these features to learn higher-level, more abstract semantic relationships. A depth of three layers was chosen as it provides sufficient capacity for this hierarchical learning without introducing excessive computational complexity or the risk of vanishing gradients that can occur in deeper architectures.

For the optimization process, the `Adam` optimizer was selected for its widespread adoption and proven effectiveness, as it efficiently combines the advantages of momentum and adaptive learning rates. Given our binary classification task, we used `binary crossentropy` as the loss function. This is the standard choice for two-class problems, as it effectively measures the distance between the predicted probability (output by a final sigmoid activation layer) and the true binary label (0 or 1).

To prevent the model from overfitting by memorizing the training data, we implemented a suite of regularization techniques. We set a maximum of 100 `epochs`, but utilized an `Early Stopping` mechanism that monitors the validation loss and halts training if no improvement occurs for five consecutive epochs, restoring the weights from the best-performing epoch. In addition to early stopping, a `dropout` rate of 0.3 was applied after each BiGRU layer, forcing the network to learn more robust features by randomly deactivating neurons during training. To further constrain the model, `L2 regularization` (with a factor of 0.01) was applied to the kernel weights to penalize large values and promote a simpler model. Finally, to ensure stable training from the outset, the He uniform initializer was chosen. This initializer is specifically designed for layers with ReLU-like internal activations (as in GRU cells) and helps prevent vanishing or exploding gradients.

4.8. Ablation analysis

To evaluate the impact of primary embedding selection on classification performance, we conducted an ablation study comparing three ensemble architectures:

EE1: GloVe (primary) + (Word2Vec, FastText) + BiGRU
EE2: Word2Vec (primary) + (GloVe, FastText) + BiGRU
EE3: FastText (primary) + (Word2Vec, GloVe) + BiGRU

As shown in Table 7, EE3 achieves the highest performance across all metrics (Accuracy: 91.93%, F1-score: 91.63%, AUC-ROC: 94.00%), outperforming both EE1 (Accuracy: 86.96%, F1-score: 86.42%) and EE2 (Accuracy: 86.33%, F1-score: 85.18%) by significant margins. This result underscores the critical role of FastText as the primary embedding, leveraging its subword decomposition mechanism to mitigate OOV challenges. While the ensemble framework reduces OOV issues by cross-referencing secondary embeddings for missing terms using a dynamic threshold, it cannot resolve cases where a term is absent across all embeddings. FastText addresses this limitation by generating vectors for OOV words through averaged subword embeddings. However, this averaging process may dilute semantic precision for terms with non-compositional meanings.

EE2 exhibits the lowest performance, primarily due to Word2Vec's lack of subword support and case sensitivity. For instance, terms like homeplugin (lowercase) are treated as OOV in Word2Vec, even though HomePlugin (mixed case) exists in its vocabulary (Table 2). This inconsistency limits its ability to benefit from secondary embeddings, as substitutions rely on exact lexical matches. By contrast, EE1 demonstrates marginally better performance than EE2 (Accuracy: 86.96% vs. 86.33%), likely due to GloVe's global co-occurrence statistics, which capture broader semantic patterns. GloVe's robustness to syntactic variations allows it to leverage secondary embeddings more effectively, as seen in its substitution of the coordinator with the controller via FastText.

Table 7. Different architecture comparison: Test Performance Metrics

Architectures	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)	MCC
EE1	86.96	86.17	86.73	86.42	91.52	0.72904
EE2	86.33	86.93	84.24	85.18	90.99	0.7112
EE3	**91.93**	**91.63**	**91.38**	**91.50**	**94.00**	**0.8300**

EE1: GloVe + (Wor2Vec, FastText) + BiLGRU
EE2: Word2Vec + (GloVe, FastText) + BiLGRU
EE3: FastText + (Wor2Vec, GloVe) + BiLGRU

4.9. Comparative analysis

To measure the effectiveness of our proposed ensemble framework, we compared it against two types of models: classic word embeddings (Word2Vec, GloVe, FastText) and modern Transformer models (like BERT and RoBERTa). As shown in Table 8, we tested these models on four different datasets. Three of these, VarIoT, CNVD, and CNNVD, contain

familiar vocabulary, while the fourth, our custom Unseen dataset, was created to challenge the models with a high number of new and domain-specific terms (OOV words).

On the datasets with familiar vocabulary (VarIoT, CNVD, and CNNVD), the large Transformer models performed the best, as expected. Models like DistilBERT and ModernBERT achieved nearly perfect scores, showing the power of their deep contextual understanding. While our proposed model did not surpass these large models, it was highly competitive and consistently outperformed the classic word embeddings. For example, on the CNNVD dataset, our model's F1-score of 99.86% was a significant improvement over GloVe (97.78%) and Word2Vec (94.37%), proving it is a strong and efficient alternative.

The most important findings come from the Unseen dataset, which was designed to mimic real-world challenges with new terminology. On this dataset, the performance of the powerful Transformer models dropped significantly. BERT's F1-score, for instance, fell from 99.49% on the familiar CNVD set to just 83.25% in unseen data. This happens because they are good at adapting to the training data. However, when faced with domain-specific OOV, they fail to assign them an appropriate embedding through a weighted average of sub-tokens.

In this challenging environment, our proposed model performed better than all others, achieving the highest scores in every key metric: Accuracy (91.93%), F1-score (91.50%), and Matthews Correlation Coefficient (0.8300). This success is due to the model's flexible design. Its ability to intelligently find and substitute similar words from different embeddings using a dynamic threshold, combined with FastText's power to understand parts of unknown words via sub-word embedding, allows it to handle new terms gracefully and avoid assigning a zero vector. This result highlights our framework's key advantage: while Transformer models are excellent in predictable environments, our ensemble method is more reliable and robust for real-world software vulnerability classification tasks, where new terms appear constantly.

4.10. Limitations and Future Work

Despite the robust performance of our ensemble framework, particularly in high-OOV scenarios, it is important to acknowledge its limitations to guide future research and improvements.

- **Risk of Contextually Inappropriate Substitutions:** While our dynamic threshold is a significant improvement over a fixed value, the framework still faces a risk of contextually inappropriate substitutions. The current mechanism calculates a threshold based on the local semantic neighborhood of an OOV word within a secondary embedding. However, it does not consider the global context of the original sentence in which the word appeared. This creates a potential failure point: a secondary embedding might provide a set of seemingly good synonyms for a word in one domain (e.g., "coordinator" in networking), but if the original sentence used the word in a different domain-specific sense, the chosen substitute (e.g., "controller") might be statistically sound but semantically incorrect for that specific context.
  - **Future Work:** A promising direction to mitigate this is to incorporate a *sentence-level context validation step*. After a potential substitute is identified, its vector could be compared not just to the OOV word's vector, but to the vector representation of the entire source sentence. The substitution would only be accepted if it

Table 8. Performance Comparison of the Models on Cybersecurity Datasets.

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC	MCC
VarIoT Dataset						
Word2Vec	96.81	71.64	96.33	79.03	99.43	0.6333
GloVe	96.91	44.00	91.67	59.46	99.41	0.6232
FastText	96.60	70.64	94.19	77.68	99.33	0.6041
BERT	**99.68**	**88.89**	94.11	**91.43**	99.95	**0.9131**
ModernBERT	99.47	77.27	**100.00**	87.18	99.97	0.8767
CodeBERT	98.94	62.96	**100.00**	77.27	99.93	0.7892
DistilBERT	99.47	77.27	**100.00**	87.18	**100.00**	0.8767
RoBERTa	99.58	80.95	**100.00**	89.47	99.97	0.8978
Proposed Work	99.25	86.31	96.73	90.83	95.58	0.8238
CNVD Dataset						
Word2Vec	98.28	98.18	98.40	98.27	99.95	0.9658
GloVe	98.69	98.66	98.71	98.68	99.92	0.9737
FastText	97.67	97.63	97.68	97.65	99.68	0.9531
BERT	99.53	99.69	99.29	99.49	99.95	0.9905
ModernBERT	99.91	**100.00**	99.80	99.90	99.99	0.9981
CodeBERT	99.01	98.88	98.98	98.93	99.97	0.9801
DistilBERT	**99.95**	**100.00**	**99.90**	**99.95**	**100.00**	**0.9991**
RoBERTa	99.72	99.59	99.80	99.69	99.98	0.9943
Proposed Work	98.68	98.77	98.59	98.67	99.25	0.9737
CNNVD Dataset						
Word2Vec	96.13	97.41	92.03	94.37	99.86	0.8929
GloVe	98.41	98.84	96.81	97.78	99.97	0.9563
FastText	96.04	97.36	91.85	94.24	99.93	0.8904
BERT	99.73	99.77	99.88	99.83	99.99	0.9922
ModernBERT	99.73	99.77	99.88	99.83	**100.00**	0.9922
CodeBERT	99.28	99.20	99.88	99.54	99.97	0.9791
DistilBERT	99.64	99.54	**100.00**	99.77	**100.00**	0.9896
RoBERTa	99.37	99.20	**100.00**	99.60	99.99	0.9817
Proposed Work	**99.91**	**99.80**	99.94	**99.86**	99.99	**0.9973**
Unseen Dataset						
Word2Vec	86.95	86.55	85.88	86.18	93.22	0.7244
GloVe	88.19	87.54	87.75	87.64	92.12	0.7530
FastText	88.20	88.82	86.33	87.24	94.49	0.7512
BERT	88.85	85.29	81.31	83.25	**94.61**	0.7496
ModernBERT	86.31	82.65	75.70	79.02	92.74	0.6903
CodeBERT	85.67	82.98	72.90	77.61	92.03	0.6744
DistilBERT	85.67	77.19	82.24	79.64	92.80	0.6868
RoBERTa	88.22	83.65	81.31	82.46	93.25	0.7361
Proposed Work	**91.93**	**91.63**	**91.38**	**91.50**	**94.61**	**0.8300**

Note: Metrics (Accuracy, Precision, Recall, F1, AUC) are reported in percentages. MCC is reported as a decimal. Bold values indicate the best performance for each metric. We have also assumed that the value ‘1.00‘ for AUC-ROC was intended to be ‘100.00‘ to be consistent with the percentage-based reporting.

maintains high cosine similarity with the overall sentence context, ensuring that the replacement is not just a synonym, but the right synonym for the situation. A large pre-trained language model, such as Sentence-BERT, can be used to check sentence similarity.

- **Dependency on Subword-Capable Primary Embedding:** The success of our framework's final fallback step is highly dependent on using a subword-capable primary embedding like FastText. We have shown this is the superior architecture in our ablation study. However, this dependency makes the framework less modular. If a different, non-subword embedding model were to become state-of-the-art for a specific domain, our current ensemble method would lose its most critical safety net for handling truly novel OOV terms.
  - **Future Work:** To address this, the framework could be enhanced with an independent OOV handling module. One approach would be to train a dedicated, lightweight subword model (like fastText.zip) on the target domain corpus itself, using it exclusively as a final fallback regardless of the primary embedding. Another avenue would be to implement a trainable $¡UNK¿$ token. Instead of relying on subwords, the model could learn a specific vector representation for all unresolved OOV words during the downstream task training, allowing it to adapt to the specific types of unknown words present in the training data.
- **Computational Overhead:** The sequential nature of the fallback mechanism introduces computational overhead compared to a single embedding lookup. While this trade-off is justified by the significant accuracy gains on OOV-rich datasets, it could be a limiting factor in latency-critical applications.
  - **Future Work:** This could be optimized by creating a pre-computed OOV-substitution cache for a given domain. For a known set of domain-specific OOV words, the best substitutes could be calculated offline and stored in a dictionary for near-instantaneous lookups. For a more advanced solution, knowledge distillation could be explored, where a larger, more complex teacher model (our ensemble) is used to train a smaller, faster student model to mimic its behavior, effectively "baking in" the ensemble's knowledge into a single, efficient model.

5. Conclusion

The escalating security challenges posed by interconnected embedded systems in critical infrastructure demand innovative approaches to vulnerability management. To address this, we proposed a novel ensemble embedding technique that synergistically integrates multiple word embedding methods to mitigate the impact of OOV words, a critical limitation in existing vulnerability classification frameworks. Coupled with BiGRU networks, our model achieves robust categorization of vulnerabilities while distinguishing between embedded and general-purpose system vulnerabilities with high precision.

A key contribution of this work is the creation of a comprehensive benchmark dataset, curated from diverse vulnerability databases, which addresses the scarcity of domain-specific data for embedded systems security research. Our experiments demonstrate the framework's superior performance in identifying embedded system-specific vulnerabilities, particularly in high-OOV scenarios, outperforming the baseline word embeddings in accuracy, recall, and semantic coherence.

References

1. Aghaei, E., Shadid, W., Al-Shaer, E.: Threatzoom: Cve2cwe using hierarchical neural network. arXiv preprint arXiv:2009.11501 (2020)
2. Ben Yahya, A., El Akhal, H., El Alaoui, A.E.B.: Machine learning-based collection and analysis of embedded systems vulnerabilities. In: Enhancing Performance, Efficiency, and Security Through Complex Systems Control, pp. 242–261. IGI Global (2024)
3. Chen, J., Kudjo, P.K., Mensah, S., Brown, S.A., Akorfu, G.: An automatic software vulnerability classification framework using term frequency-inverse gravity moment and feature selection. Journal of Systems and Software 167, 110616 (2020)
4. Franceschi-Bicchierai, L.: How this internet of things stuffed animal can be remotely turned into a spy device, https://www.vice.com/en/article/how-this-internet-of-things-teddy-bear-can-be-remotely-turned-into-a-spy-device/, accessed: Dec 2025
5. Han, Z., Li, X., Xing, Z., Liu, H., Feng, Z.: Learning to predict severity of software vulnerability using only vulnerability description. In: 2017 IEEE International conference on software maintenance and evolution (ICSME). pp. 125–136. IEEE (2017)
6. Huang, G., Li, Y., Wang, Q., Ren, J., Cheng, Y., Zhao, X.: Automatic classification method for software vulnerability based on deep neural network. IEEE Access 7, 28291–28298 (2019)
7. Ikegwu, A.C., Uzuegbu, V., Alo, U.: Review of embedded systems and cyber threat intelligence for enhancing data security in mobile health. vol 7, 98–120 (2025)
8. Na, S., Kim, T., Kim, H.: A study on the classification of common vulnerabilities and exposures using naïve bayes. In: Advances on Broad-Band Wireless Computing, Communication and Applications: Proceedings of the 11th International Conference On Broad-Band Wireless Computing, Communication and Applications (BWCCA–2016) November 5–7, 2016, Korea. pp. 657–662. Springer (2017)
9. Newman, Lily Hay: A new pacemaker hack puts malware directly on the device (Accessed: Dec 2025), https://www.wired.com/story/pacemaker-hack-malware-black-hat/, accessed: Dec 2025
10. Nie, S., Liu, L., Du, Y.: Free-fall: Hacking tesla from wireless to can bus. Briefing, Black Hat USA 25, 1–16 (2017), https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf
11. Papp, D., Ma, Z., Buttyan, L.: Embedded systems security: Threats, vulnerabilities, and attack taxonomy. In: 2015 13th Annual Conference on Privacy, Security and Trust (PST). pp. 145–152. ieee (2015)
12. Prigg, M.: How to get green lights all the way to work: Hackers reveal how simple it is to control traffic lights in major cities using just a laptop https://www.dailymail.co.uk/sciencetech/article-2730096/How-green-lights-way-work-Hackers-reveal-simple-control-traffic-lights-major-cities-u html, accessed: Dec 2025
13. Rios, A., Lwowski, B.: An empirical study of the downstream reliability of pre-trained word embeddings. In: Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020) (2020)
14. Sharma, R., Sibal, R., Sabharwal, S.: Software vulnerability prioritization using vulnerability description. International Journal of System Assurance Engineering and Management 12, 58–64 (2021)
15. Spanos, G., Angelis, L.: A multi-target approach to estimate software vulnerability characteristics and severity scores. Journal of Systems and Software 146, 152–166 (2018)
16. Stanislav, M., Beardsley, T.: Hacking iot: A case study on baby monitor exposures and vulnerabilities. Rapid7 Report (2015)

17. Vishnu, P., Vinod, P., Yerima, S.Y.: A deep learning approach for classifying vulnerability descriptions using self attention based neural network. Journal of Network and Systems Management 30, 1–27 (2022)
18. Wang, J.A., Guo, M.: Vulnerability categorization using bayesian networks. In: Proceedings of the sixth annual workshop on cyber security and information intelligence research. pp. 1–4 (2010)
19. Wang, Q., Gao, Y., Ren, J., Zhang, B.: An automatic algorithm for software vulnerability classification based on cnn and gru. Computers & Security p. 103070 (2022)
20. Wang, Q., Gao, Y., Ren, J., Zhang, B.: An automatic classification algorithm for software vulnerability based on weighted word vector and fusion neural network. Computers & Security 126, 103070 (2023)
21. Wen, T., Zhang, Y., Wu, Q., Yang, G.: Asvc: An automatic security vulnerability categorization framework based on novel features of vulnerability data. J. Commun. 10(2), 107–116 (2015)

Aissa Ben Yahya is a Ph.D. Student in the Department of Computer Science and Applications at Moulay Ismail University, Morocco. He received his Master's degree in Networks and Embedded Systems from Moulay Ismail University, Morocco, in 2020. His research interests include using deep learning and Natural Language Processing (NLP) techniques for vulnerability classification and detection, deep learning for NLP, and the use of advanced NLP models such as transformers for various tasks.

Hicham El Akhal received his master's degree in Networks and Embedded Systems from Moulay Ismail University, Morocco, in 2020. He is currently working toward a Ph.D. degree in the Department of Information Science, Faculty of Sciences, at Moulay Ismail University, Meknes, Morocco. His research interests include artificial intelligence and agriculture.

Abdelbaki El Belrhiti El Alaoui is working as a Professor in the Department of Information Science, Faculty of Sciences Meknes, Moulay Ismail University, Morocco. He received his the Ph.D. degree form Metz University, France, in 2002, and the accreditation to supervise research degree from Moulay Ismail University, in 2009. His research areas include Wireless Sensor Networks, Network Security, Embedded Systems and Internet of Things.

Received: March 14, 2025; Accepted: November 29, 2025.

Computer Science and Information Systems 23(1):419–441 https://doi.org/10.2298/CSIS250320078H

Towards Enhancing Data Science Agents with Semantics*

Sayed Hoseini[1], Maximilian Ibbels[1], Maximilian Knoll[1], and Christoph Quix[1,2,**]

[1] Hochschule Niederrhein University of Applied Sciences, Krefeld, Germany
sayed.hoseini@hsnr.de
[2] Fraunhofer Institute for Applied Information Technology FIT, St. Augustin, Germany
christoph.quix@fit.fraunhofer.de

Abstract. Data lakes, initially designed for storing heterogeneous datasets, have recently been extended with ML capabilities to unify data science tasks within a single platform. However, they still lack essential ML-specific features, limiting their effectiveness for end-to-end automation. Automated Machine Learning (AutoML) and Large Language Models (LLMs) offer potential solutions by streamlining various stages of the ML pipeline, yet both have significant limitations.
This paper presents an integration of AutoML frameworks and LLMs within a data lake system. We introduce a metadata model to capture data analytics processes, a Python package wrapping existing AutoML libraries, and a module utilizing LLMs to automate ML tasks. A comparative evaluation indicates that AutoML simplifies pipeline creation but limits user control and lacks robust data preprocessing support. LLMs can automate individual tasks, such as code generation, but struggle to orchestrate complete workflows effectively. Both approaches risk staying as basic prototypes that still need manual improvement.
The primary challenge lies in managing task interdependencies within ML pipelines. Retrieval-augmented generation enables dynamic access to external information but may overlook structured data relationships, leading to incomplete or redundant results. Therefore, we propose an extended vision that integrates multi-agent frameworks for data science with knowledge graphs that capture historical experience from previous ML experiments. We present preliminary results for developing comprehensive, context-aware ML agents and their integration into our data lake system SEDAR.

Keywords: AutoML, LLMs, Semantic Data Lake, MLOps, Data Science Agents.

1. Introduction

As modern data lakes incorporate an ever-growing variety of heterogeneous data sources, the complexity of utilizing this data also increases. The challenge lies in finding suitable methods to analyze these diverse and often unstructured data [36]. Traditionally, profound knowledge of data science methods is required to develop appropriate machine learning (ML) pipelines to optimally process data. Data science workflows are inherently complex, consisting of interdependent tasks like data processing, feature engineering, and model training. Addressing these tasks demands iterative refinement and real-time adjustments, as both the data and requirements are constantly evolving. This poses a significant

* This is an extended version of [20].
** Corresponding author

hurdle, particularly in domain-specific areas like chemistry, where experts often lack the necessary skills. Democratizing access to advanced analytics to a wider audience is thus highly desirable.

AutoML is the process of automating the end-to-end application of ML to real-world problems. However, existing AutoML approaches, reach their limits when it comes to automatically creating data-driven models with sufficiently high accuracy [26,20]. Furthermore, they focus on model selection and hyperparameter tuning, but do not help with other critical steps like data integration, processing and cleaning [42].

Large Language Models (LLMs) can potentially increase ML pipeline automation by assisting at all stages of the process. They can help with answering specific questions or even generate and execute the necessary code entirely and autonomously [14,38]. However, LLMs have difficulty understanding and adapting to specific requirements and contexts leading to generic, erroneous or inappropriate code for specialized tasks [20,25]. Furthermore, while they can be applied to more individual data science tasks [13,31], they still cannot perform holistic workflows and struggle to execute complete data science processes in one go. This is due to an inability to handle real-time changes in intermediate data and adapt dynamically to evolving task dependencies.

Both directions struggle to move beyond being merely used as an initial prototype that requires to be improved by humans manually to achieve the desired performance. Another weakness of these systems is that they start from scratch with each new model generation, without drawing on existing experience or knowledge [46]. Integrating the experience of historical training results, consisting of hyperparameters and various metrics related to result quality, can significantly accelerate and improve this process.

Recently, several techniques have been proposed to tackle these challenges with LLMs. One of the most notable is Retrieval-Augmented Generation (RAG), which leverages information retrieval methods to identify data relevant to the question at hand. Relevant information can then be provided to the LLM as additional context during inference, often leading to improved answer quality [12]. Another promising approach is combining LLMs (and potentially RAG) with knowledge graphs (KGs) [17]. Knowledge graphs use graph-based representations to organize, integrate, query, and reason about diverse collections of data and knowledge in a structured manner.

In this article, we address the question of how the integration of automated machine learning can augment the capabilities of a data lake system and compare the efficacy of AutoML tools with LLMs in this context. In addition, we examine the question of how such general-purpose AI systems can be customized and optimized for data science activities by integrating knowledge from structured KGs. Our contributions include:

1. We propose an integrated framework *AutoMLWrapper* for heterogeneous AutoML systems, introducing a uniform abstraction layer for job orchestration.
2. We introduce a methodology for LLM-assisted configuration and automation of ML pipelines in data lake environments. The approach combines generative reasoning with structured metadata management to improve data science workflows.
3. We present a conceptual framework that unifies RAG, KGs, and LLM-based data science agents, highlighting their complementary roles: RAG for contextual retrieval, KGs for semantic grounding, and LLMs for adaptive reasoning and synthesis.

4. We investigate the semantic integration of LLMs with the OpenML knowledge graph, aiming to exploit metadata on prior ML experiments for the development of semantics-aware data science agents.
5. To validate the proposed concepts, we implemented a prototype framework within our *SEDAR* data lake system [16]. The prototype operationalizes the integration of AutoML components and LLM-based agents, demonstrating the feasibility and practical relevance of the approach.
6. We empirically evaluate the proposed framework on complex, non-standard ML tasks in the domain of chemistry, demonstrating its applicability and discussing the relative performance and interpretability of LLM-based and AutoML-based techniques.

This paper is an extended version of [20], in which we presented a preliminary version of our framework, in particular the *AutoMLWrapper* and its comparison with LLM-generated ML pipelines. These results are presented in section 3. Compared to [20], we have updated the discussion of related work in Section 2, to include recent approaches on AutoML, LLMs and their application as data science agents. Section 4 is an original contribution of this paper. It proposes an architecture for a KG-enhanced data science agent for more comprehensive ML workflows. The evaluation in Section 5 has also been extended to include results from the KG-enhanced approach.

2. Related Work

During his presentation at the NeurIPS 2024 conference, Ilya Sutskever – one of the researchers behind ResNets [29] and co-founder of OpenAI – argued that the era of pre-training neural networks, which has primarily focused on scaling compute and data, has reached a limit [39]. While computational power continues to grow through larger clusters and advancements in hardware and algorithms, the data side has plateaued, as the global Internet, essentially the only vast data resource, is already more or less fully utilized. From this saturation of data, he argues, emerges the increasing importance of autonomous agents moving forward. In the context of AI/ML, agents typically refer to autonomous systems that can take actions, make decisions, and interact with their environment to achieve specific goals [10]. LLM-based agents for data science (short: data science agents) activities essentially resolve to instruct a "large" AI to create "smaller" ones.

Early works on LLMs for AutoML Tornede et al. [40] review the current state of the integration between LLMs and AutoML in both directions. Currently, interacting with an AutoML system is often still challenging for non-expert users. They highlight the potential to substantially improve the human interaction component of AutoML tools, alleviate the tedious task of correctly configuring the tool which often requires experts, assist in interpreting the output, and improve several internal components through knowledge. Simultaneously, the integration also bears risks such as inadequate evaluation and catastrophically wrong behavior of AutoML systems due to hallucinations [25] of an LLM-powered component, and ever-increasing resource demands.

Sun et al. [37] present a survey on LLM-based data science agents including a timeline of publications, separated into commercial and publicly available ones. They find

major obstacles to be the integration of domain-specific knowledge and handling multi-modal inputs, i.e. user instructions alongside charts, tables, code, etc. as well as planning complete data workflows (see below).

An early framework that uses *ChatGPT* as a *Virtual Interactive Data Scientist* is presented in [11]. The research represents a systematic approach to creating a personal data scientist, but the authors admit that there are still severe problems to be addressed, such as erroneous recognition of the correct current state by the LLM.

Several works utilize LLMs for hyper-parameter optimization [51,50,32,35]. These methods help with model selection and allow an agent to conduct experiments with specific hyper-parameters for selected algorithms to iteratively optimize them based on historical trials while autonomously processing the task information. Chen et al. [2] integrate LLMs as adaptive operators in an evolutionary neural architecture search, Wei et al. [43] deploy LLM-based agents to devise tailored solutions for diverse graph-structured data and learning tasks. In [23,8] we find a novel framework for benchmarking AI research agents. *AutoMMLab* [45] is an LLM-empowered AutoML system that follows the user's language instructions to automate model production workflows for computer vision tasks. Other methods focus on feature engineering [13,30,33].

Microsoft's *CoML* [50] (formerly MLCopilot) promises to derive knowledge in textual form, which can be reused for new machine learning requirements. The idea is to perform knowledge-based reasoning, that is to leverage LLMs to conduct reasoning and task solving based on previous knowledge. This knowledge is extracted from historical data and elicited into an experience pool in text form that is provided to the LLM at runtime for a given task. However, it relies on LLM-generated embeddings of task descriptions and the prototype[3] can only return hyperparameters for a predefined method.

End-To-End Data Science Agents All of the aforementioned methods only focus on substeps in the ML pipeline. A major challenge is the complexity of planning the entire pipeline, mainly due to the interdependencies between each step. For example, the type of dataset influences preprocessing and neural network design, which in turn affects the hyperparameters that need optimization for a specific downstream task. These inter-step dependencies expand the search space to account for all possible combinations of related steps. Additionally, supporting multiple downstream tasks further complicates the process, as each task has its specific requirements [41,14]. For example, data cleaning and feature engineering are essential prerequisites, but they also must be tailored to each case.

The *Data Interpreter* [14], part of the multi-agent framework *MetaGPT* [15], first constructs a dynamic plan graph (*task graph*), which is used to capture dependent tasks and arrange them in a meaningful sequence. For each of the seven tasks (data exploration, correlation analysis, outlier detection, feature engineering, model training, model evaluation, and visualization) the LLM creates an *action graph* to execute actions in the form of generated code. Those are executed locally to obtain partial results. Errors are sent back to the LLM for autonomous debugging. Additional customized scripts can be provided as tools, for example, for data exploration/analysis.

DS-Agent [9] integrates LLMs with Case-Based Reasoning (CBR) to automate data science tasks. In the development stage, it structures an automatic iteration pipeline using the CBR framework to leverage expert knowledge from platforms like Kaggle and facili-

[3] https://github.com/microsoft/CoML

tates consistent performance improvement through a feedback mechanism. This paradigm adapts past successful solutions for direct code generation, which significantly reduces the demand for the foundational capabilities of LLMs, enabling efficient deployment with lower resource requirements. As opposed to RAG, which focuses on retrieving and integrating real-time information from external data sources to improve the model's response, CBR focuses on case libraries that store detailed solutions to specific problems, which are reused and adapted for new situations.

AutoML-Agent [41], a multi-agent framework takes user's task descriptions and implements collaboration between specialized LLM agents to deliver deployment-ready models. Five different agents interact with each other each responsible for a different task, in particular managing prompts, data, models, operations and agents. The framework includes agent specifications, a prompt parsing module, a retrieval-augmented planning strategy, a prompting-based plan execution, and a multi-stage verification. Experiments on seven ML tasks demonstrate that *AutoML-Agent* outperforms existing methods in terms of success rate and downstream task performance.

AgentKv1.0 [7] manages the data science lifecycle through reasoning frameworks that dynamically process memory using past experiences to guide future decisions. The system integrates various tools and libraries like *Torchvision* and *HuggingFace* to handle multimodal data from computer vision, natural language processing, and tabular data. The agent is evaluated through a competitive *Kaggle*[4] benchmark. The evaluation framework assesses end-to-end capabilities, from task retrieval to submission on *Kaggle* and achieved a 92.5% success rate in automating tasks, spanning multiple domains and data types. The agent ranks in the top 38% of 5,856 human Kaggle competitors, performing at a skill level comparable to Expert-level users (Grandmaster level on *Kaggle*). Notably, *AgentK* has won 6 gold medals, 3 silver medals, and 7 bronze medals in *Kaggle* competitions. Unlike traditional LLM methods that rely on chains of thought, *AgentK* employs a memory module for continuous improvement and adaptation that allows for a flexible learning paradigm, where the agent can generalize across different tasks by leveraging past experiences.

To summarize, LLMs still work relatively well on classical tasks like tabular classification and regression as compared to traditional AutoML libraries, such as AutoGluon, but LLM-based agents work significantly better in complex tasks. However, tasks such as reinforcement learning and recommendation systems pose particular challenges in both paradigms. On one hand, existing frameworks place less emphasis on these areas, and on the other, contemporary LLMs have received less specialized training to effectively address them. In terms of limitations, even though these agents offer increased flexibility across ML tasks and data modalities, the absence of skeleton code increases the risk of code hallucination. Additionally, the code generation quality varies heavily when using different backbones, e.g. GPT or open-source models. Developing a more robust framework less reliant on the LLM backbone is imperative [41].

[4] https://www.kaggle.com/

3. Enhancing the Machine Learning Capabilities in SEDAR with AutoML and LLMs

This section is largely based on our article [20]. We use the resulting findings to motivate researching semantics-aware agents for data science tasks.

3.1. Integrating AutoML and LLMs with *SEDAR*

SEDAR is designed in a modular, easily extendable, and flexible six-layer architecture based on open-source technologies deployed as microservices (see **??**). To realize a true polystore [27], the *storage layer* is composed of different storage systems to support efficient storage for heterogeneous data structures based on their specific data model. By default, *Hadoop* with its file system *HDFS* is used to store files with arbitrary structures including audio, images, video, text, and ML model artifacts. The *transformation layer* implements various operations for data preparation, e.g., cleansing, transformations, or joins. We employ *Apache Spark* as a unified data transformation engine for large-scale data processing. *Spark*'s engine can process arbitrary data sources in the form of *DataFrames*. A programmatic interface to communicate with *SEDAR* is provided by the *SEDAR-API*, an external Python module. To ensure generality, the *ingestion layer* is independent of any formats and data sources, not limited to particular systems in the storage layer, and allows for versioning and the addition of further databases. The metadata service extracts and manages information about versions, updates, profiles, etc. automatically. The *interaction layer* facilitates comprehensive metadata management while ensuring ease of use for non-technical users.

One of *SEDAR*'s core objectives is to facilitate knowledge management on top of the stored data. Semantic data can enrich basic metadata, such as schema and data types, by adding context information not originally present in the data source. One way to achieve this is to map raw data from heterogeneous data sources to semantically rich models to increase the usability and interpretability of data. For this, a *semantic data management layer* [21] has been incorporated that allows to associate semantics to datasets. Using these semantics, *SEDAR*'s semantic query engine allows querying heterogeneous data sources using a uniform semantic data model via ontology-based data access [21].

SEDAR is deployed in Industry 4.0 settings [49] in the chemical domain [**?**,20], which increasingly demands modern data lakes to aid in creating, managing, and deploying ML applications while also providing necessary data [5]. To facilitate Machine Learning Operations (MLOps) [28], in a *data analytics layer SEDAR* utilizes MLFlow [47], an open-source framework to manage the inputs and outputs of ML applications [47]. This includes tracking model versions, code, and tuning parameters, ensuring reproducibility of results, and facilitating model deployment. **??** illustrates the general view of the machine learning tab that lists existing experiments and deployed models at the top and shows details about past runs associated with an experiment at the bottom. *Jupyter Notebooks* have been integrated into the interaction layer and are hosted from a *JupyterHub*, allowing users to have personalized work environments. Authentication between web services is centralized using the OAuth protocol and JWT tokens. Our self-hosted GitLab instance serves as the identity provider and central repository for storing code. Within *SEDAR*,

users can select datasets, and then choose the desired learning method, and the system will automatically generate a notebook with the necessary code to load the *DataFrames* from storage via the *SEDAR API*, register their experiments, track individual runs and software environments, store model artifacts, and deploy their applications.

3.2. AutoMLWrapper

The *AutoMLWrapper* is a stand-alone Python module acting as an interface for three open-source AutoML frameworks: *AutoKeras*, *AutoGluon* and *AutoSklearn*[5]. Special emphasis is placed on the standardized construction of AutoML jobs by configuring the underlying libraries using a common set of parameters. Object-oriented programming for library-unspecific logic facilitates easy integration of new AutoML libraries without redesigning the entire interface. To use the package with code as well as in combination with *SEDAR*'s frontend, presenting information about possible settings for AutoML experiments required a form that can be easily understood and displayed. To meet these requirements, each of the three frameworks is equipped with a configuration file essential for the correct call to the underlying library. Then, there is a global configuration file that contains meta-information about the available libraries' scope and assigns data and problem types. For example, the file can be used to decide in the frontend which (library) functions are available to the user for image segmentation.

The *AutoMLWrapper* is integrated into *SEDAR* within a dedicated virtual Python environment built into the *Jupyterhub*'s default image. Similar to the previous manual training scripts a complete *Jupyter Notebook* is created and populated with information from the frontend. By design, the package also integrates with *MLFlow* to manage experiments and save the generated models. Despite the idea of automation, the package also provides enough freedom for experienced users allowing them to add their logic and specific customizations. This notebook serves as the basis for running the AutoML job, of which relevant metrics and parameters are automatically recorded and tracked by *MLFlow*.

3.3. LLM-Orchestrated ML

LLMs assist the user at multiple locations in the *SEDAR* framework. A new user interface based on *LangChain* has been implemented to enable communication between the user and LLM providers, allowing for the dynamic integration of additional information from the system's metadata. The current implementation offers a simple chat functionality from the frontend. The interface can be used with various LLM providers.

JupyterAI is an extension for *Jupyter* that enables the use of LLMs within a separate chat window and the cells of a notebook. It distinguishes between language and embedding models. The embedding model is used to create a compressed representation of the contents of a particular directory, and the language model is used to ask specific questions about the file content. Users can either have methods or code from the notebook explained to them, or have code sections and entire notebooks generated. *JupyterAI* can be connected to a range of default models, but also to *Hugging Face*, *GPT4All* and *Ollama* which include a variety of open-source LLMs of varying sizes.

[5] github.com/automl/auto-sklearnl/keras-team/autokerasl/autogluon/autogluon

Furthermore, the *CAAFE* library [13] is used to automate feature engineering of classification problems with tabular data using LLMs. The library has been extended to image data in classification and segmentation problems. *CAAFE* processes a dataset D in an iterative process. An LLM is dynamically prompted to generate code. By executing this code on D, a new dataset D' is created, which contains the feature changes of this iteration. An identical model is trained for both datasets and problem-dependent metrics are measured. These metrics are communicated to the LLM for the next iteration, which receives D' in the next iteration if there is an improvement.

4. Unifying Knowledge Graphs with Data Science Agents

Building on the insights from related work, it becomes evident that, despite their transformative potential, LLMs still face several fundamental challenges that limit their reliability and applicability. According to Hogan et al. [12], these challenges include the tendency of LLMs to generate information that is not grounded in factual data (often referred to as hallucination), the lack of transparency regarding the sources and reasoning behind their outputs (opaqueness), the difficulty of maintaining up-to-date knowledge due to the high cost of retraining (staleness), and the inherent inability to provide exhaustive or complete responses as a consequence of their probabilistic nature (incompleteness).

As we have seen in Section 2, to address these issues, data science agents often rely on RAG mechanisms. RAG operates by combining the generative capabilities of LLMs with a retrieval mechanism that actively searches for relevant information. The retrieval process acts as a bridge to external knowledge, allowing the agent to pull in additional information that might not be delivered from the model's internal parameters only. This external data is then used by the LLM to generate more accurate and context-aware responses. In the context of data science, this can be historical experience, i.e. prior solutions to similar problems as seen in frameworks like *DS-Agent*, *AutoML-Agent*, and *AgentK* [9,41,7] (see Section 2), where retrieval is used to ensure that the models can dynamically adjust to new, unseen problems by leveraging a pool of knowledge about past experiences.

The connection to search engines (SE) and information retrieval (IR) systems is key here. Just as search engines index and retrieve the most relevant documents based on user queries, RAG enables LLMs to access and integrate external data sources. This improves their ability to provide solutions to complex data science tasks, particularly when dealing with multimodal data, and when addressing issues that require dynamic knowledge updates. While search engines excel in providing broad coverage and real-time updates, with transparent and efficient access to vast amounts of data, they struggle with returning complete and precise answers due to noisy or incomplete data, and they cannot generate or synthesize new content from their indexed sources [12]. In addition, Peng et al. [34] identify several limitations inherent to current Retrieval-Augmented Generation (RAG) approaches. First, conventional RAG methods often fail to capture the structured relationships that exist among pieces of textual information. While semantic similarity enables retrieval of contextually relevant content, it overlooks relational knowledge such as citation links or entity associations that are essential for a comprehensive understanding of interconnected data. Second, RAG systems tend to introduce redundancy when retrieved documents are concatenated into prompts, resulting in unnecessarily lengthy and repetitive inputs. Finally, RAG architectures frequently lack mechanisms to integrate informa-

tion at a global level: although relevant snippets can be retrieved from multiple sources, these fragments are not always synthesized into a coherent or contextually consistent representation of the underlying knowledge.

In contrast, knowledge graphs offer highly curated, structured data that ensures more precise, complete, and consistent results, especially in domain-specific contexts [6]. They explicitly store rich factual, interconnected knowledge, providing a structured, semantically rich representation of domain-specific relationships and entities, enabling more precise and context-aware retrieval of relevant information. However, they are difficult to construct and evolve by nature, making it challenging to generate new facts and represent unseen knowledge [24]. As previously described, LLMs have significant limitations due to the lack of external knowledge [1]. So-called *KG-enhanced LLMs* integrate knowledge graphs with LLMs to address the difficulty in recalling factual knowledge by aiming to generate knowledge-grounded contents [22]. An example is *KG-Rank* [44], where the authors leverage a medical knowledge graph with ranking techniques, aiming to improve free-text question-answering in the medical domain. Upon receiving a question, they initially retrieve factual information from a medical knowledge graph and then apply a ranking to obtain the most relevant triplets which then, combined with the task prompt, are input into LLMs for answer generation. A similar idea is presented by *CoML* [50], where, although it is not powered by knowledge graphs, the LLM is assisted by a so-called knowledge pool to retrieve relevant experience based on the task description. In

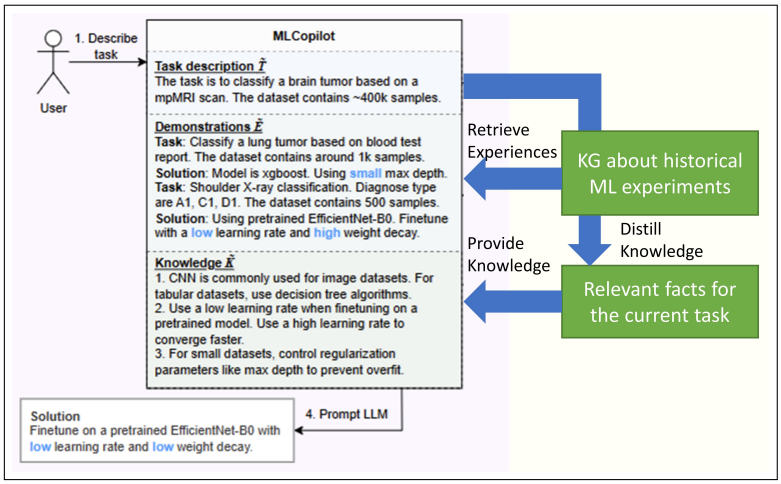


Fig. 1. Example of a KG-enhanced data science agent, extended from *CoML* [50]

contrast, Figure 1 applies this idea to knowledge graphs and shows how factual knowledge about past ML experiments can potentially be distilled and provided to an LLM for automated data science. The performance is improved by adapting dynamically to historical data relevant to the task at hand. Importantly, in this example, both RAG and knowledge graphs are used in synergy. While RAG provides dynamic access to external data sources, the knowledge graph ensures that the retrieved information is structured, precise, and con-

textually relevant. Therefore, it is reasonable to view not only search engines and LLMs but also knowledge graphs as complementary technologies whose integration has the potential to produce synergy, capitalizing on the strengths of each while mitigating their respective weaknesses. **??** provides a comparison between the discussed technologies, LLMs, search engines, knowledge graphs as presented by Hogan et al. [12].

Search engines excel at providing broad coverage of diverse topics quickly, but their results may be imprecise or incomplete due to noisy data. They rely on indexed content and cannot generate or synthesize new information. Knowledge graphs offer highly structured, domain-specific data, ensuring precise and complete results, but their coverage is narrower compared to search engines, and they require more complex querying. LLMs are versatile and generative, capable of creating novel content and synthesizing data from multiple sources, but they may produce hallucinations, generating incomplete or inaccurate results. While search engines and knowledge graphs are more transparent in terms of the data used, LLMs are often opaque, making it harder to understand how answers are derived. Search engines and LLMs are user-friendly with natural language input, whereas knowledge graphs are typically harder to use due to their reliance on structured queries. Efficiency-wise, search engines and knowledge graphs handle queries efficiently, while LLMs require more resources.

4.1. Towards a KG-enhanced Data Science Agent

In the following, we describe current research directions in our lab examining the question of how to extend the well-known integration of search engines and LLMs by a component that also leverages knowledge graphs.

We have seen how information retrieval as well as knowledge graphs can enhance the performance of LLMs. While RAG-based information retrieval has certain advantages, for data science tasks it is necessary to provide accurate factual knowledge capturing the interconnection between multiple resources, for which only knowledge graphs represent viable solutions. The essential role of knowledge provider will be covered by *MLSea* [4] in our approach.

In *CoML* [50], the source of historical data is not clearly defined. In contrast, *MLSea* aggregates ML datasets, experiments, software, and scientific works from *OpenML*, *Kaggle*, and *Papers with Code* into a structured knowledge graph. Key components of *MLSea* include the *MLSO* ontology, *MLST* taxonomies, and the *MLSea-KG* knowledge graph, which collectively aims to improve the search, explainability, and reproducibility of ML pipelines. *MLSea* thus provides a unified view of ML processes, helping users discover datasets, models, algorithms, hyperparameters, and relevant publications in one place. This system may address the challenges of accessing diverse ML knowledge and support research in automated ML systems and experiment documentation. To effectively apply RAG to the *MLSea* knowledge graph, it is essential to first understand the underlying structure of the graph. Adapting the retrieval mechanism to the knowledge graph's data model — such as the types of entities (e.g., algorithms, models, tasks) and their connections (to datasets, hyperparameters, runs, etc.) — ensures the ability to accurately query the graph for relevant information.

When exploring *MLSea*, we quickly observed a strong overall heterogeneity, i.e., varying quality of descriptions, incompleteness in different patches of the graph, heterogeneous notation, etc. Here, *AssistML* [46] can provide valuable assistance. It proposes a

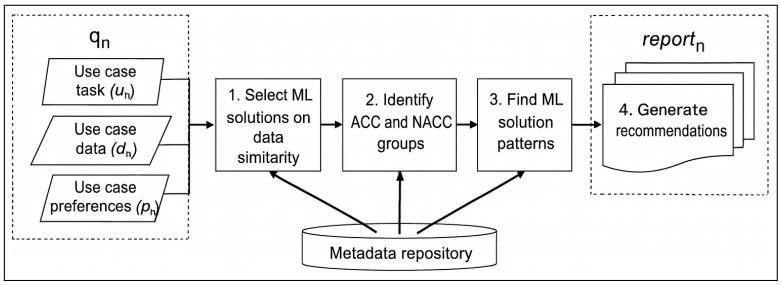


Fig. 2. Overview of the workflow of AssistML [46]

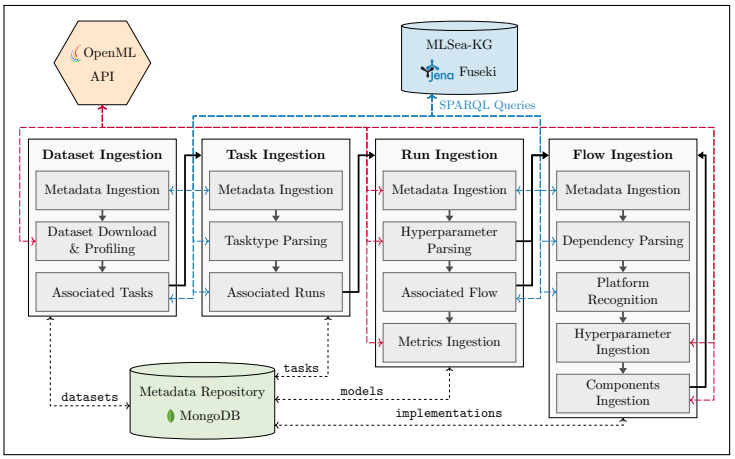


Fig. 3. Ingestion pipeline

metadata model that standardizes different datasets and task types, making them comparable and enabling it to suggest solutions for target metrics without relying on LLMs. As shown in fig. 2, its original workflow is divided into four steps: (1) select ML solutions based on data similarity, (2) identify acceptable (ACC) & nearly acceptable (NACC) ML solutions, (3) find ML solution patterns and finally (4) generate a list of recommendations. Our idea is to populate metadata repository with data from MLSea.

For this, we extended this implement the ingestion pipeline shown in Figure 3. We decided to limit the implementation to data related to *OpenML* to decrease complexity because the bulk of data is stored here and the remaining two subgraphs show higher incompleteness and heterogeneity. The ingestion extracts, transforms, and loads *OpenML Datasets*, *Tasks*, *Flows*, and *Runs*[6] into a structured framework.

Each dataset is analyzed using a data profiler to understand its characteristics, which are then stored in a *MongoDB*. Once datasets are processed, the pipeline retrieves and processes ML tasks linked to them, such as classification, regression, clustering. Task data establishes relationships between tasks and their corresponding datasets. The pipeline then processes implementations by extracting metadata, including dependencies and soft-

[6] https://docs.openml.org/concepts/, accessed on 11.03.2025

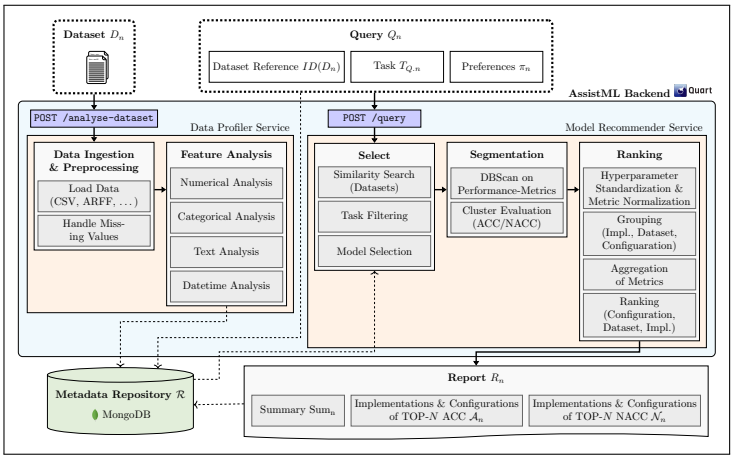


Fig. 4. Backend for generating reports

ware requirements, from *OpenML*. Implementations are then stored in the database and linked to relevant tasks. For model processing, the pipeline retrieves experimental runs associated with tasks and extracts model performance data. It constructs model setups by combining hyperparameters, implementations, and tasks to create a complete experiment configuration. Evaluation metrics are derived from run data, enabling performance assessment and comparison.

MLSea serves as a semantic layer that connects datasets, tasks, implementations, and models using well-defined relationships. SPARQL queries help retrieve dependencies and software requirements, which are structured as linked entities in the knowledge graph. This way, the pipeline can dynamically fetch available implementations and cross-check them with *OpenML* before storing them. The knowledge graph also plays a role in model processing to retrieve existing experiment configurations, hyperparameters, and performance metrics. The structured representation ensures that models are linked to their corresponding datasets, tasks, and implementations. This enables efficient retrieval of past experiments for knowledge-driven recommendations providing a structured foundation for AutoML workflows.

Once the metadata repository is populated with the historical data, the backend, shown in Figure 4 can generate reports. This means for a given query, composed of a dataset, a task and user preferences, it generates model recommendations. The pipeline processes the dataset first using the data profiler, identifying feature types and target attributes. The model recommender is designed to suggest models with hyperparameters and configurations based on dataset characteristics and user-defined performance criteria. The module leverages the metadata from existing ML solutions to facilitate the rapid identification of suitable models, thereby streamlining the model selection process for users. Users can specify performance preferences, such as accuracy, training time, or resource consumption and the recommender system aligns its suggestions with these criteria. This efficiency contrasts with traditional AutoML systems, which may require extensive computational resources and time to explore various model configurations. The system offers explana-

tions for its recommendations, making it accessible to users with varying levels of ML expertise.

The output of the recommender component is a report, a structured evaluation for a given query, which provides a summary of the best models, their scores, and explanations regarding why they were chosen. In the report, models are ranked according to multiple factors, including dataset similarity, implementation characteristics, and metric performance, to provide recommendations on the most suitable models for a given dataset and task. In the final stage of the workflow (see Figure 5), models from the report are included in a prompt that are passed to a LLM to generate executable ML pipeline code. The produced code is then executed in the local environment on the specified dataset, and evaluation metrics are computed. The results and metrics are returned as output, providing an empirical assessment of the generated pipeline.

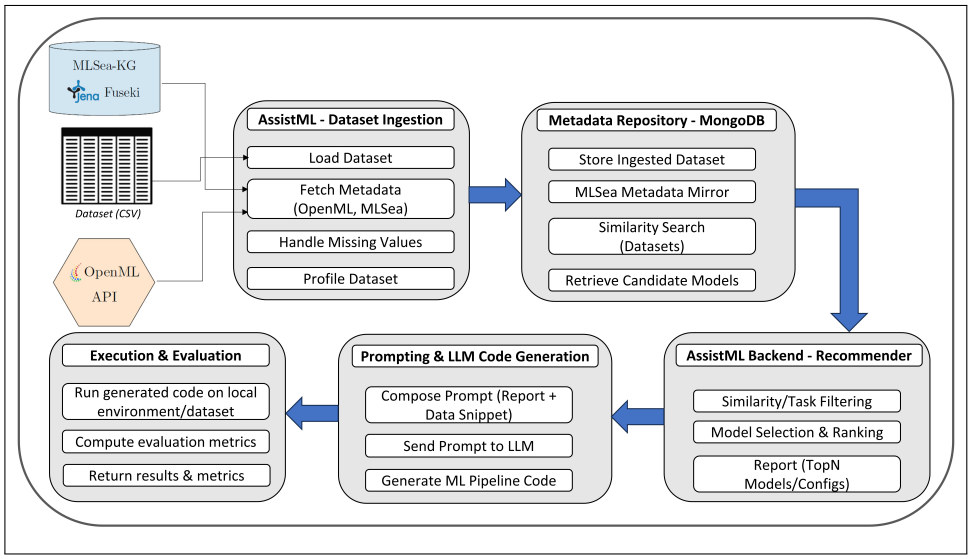


Fig. 5. Code generation pipeline with LLMs and MLSea-enhanced AssistML recommendations

5. Evaluation

5.1. Evaluation Setup, Datasets, and Criteria

We evaluate the proposed frameworks against problems from real-world use cases at the Institute for Surface Technology of the Niederrhein University of Applied Sciences (HIT). The HIT researches (organic) coatings and paints which play a vital role in safeguarding surfaces. ML is applied in this context for experiment suggestions and data analysis. All datasets, code, and training details for this study have been exported from *SEDAR* and made available online. For detailed information about dataset sizes and class distributions,

we refer to the corresponding publication, as we have tried to reproduce the experimental settings accordingly as much as possible. The experiments are performed in the described *Jupyter* environment equipped with a *Nvidia Quadro RTX8000* graphics card, two *Intel Xeon Gold 6230R* processors, and 754 GB RAM. Training for all models is initially limited to one hour. For the *AutoKeras* library, for which no time limit can be defined, a division into 50 models for 50 epochs is used.

Because working with LLMs can be freely designed by the user, rules to improve reproducibility are defined. LLMs in *GPT4All* are designed for the English language, hence the conversations are conducted in English. Problems are described by the user. It is assumed that the user cannot provide an exact definition of the sequence of experiments due to a lack of experience. Hence, ideally, the user does not use any ML terminology when working with LLMs. For example, it should not be mentioned that the class distribution of the data set is unbalanced. Instead, it should be checked whether the contextual information provided by the *SEDAR* AutoML solution is sufficient for the LLM to act accordingly. Any further steps identified by the LLM through continued querying for a more robust ML pipeline are tracked as further effort. The two open-source LLMs that are available in

```
%%ai chatgpt Generate a Python code snippet for an Image Segmentation pipeline using the libraries Tensorflow or Pytorch and other
appropriate libraries, optimized for GPU usage. The code should handle a dataframe called gitter_trainv1 with columns 'image' and
'label'. 'image' contains the path to the image file and 'label' contains the path to the mask file. The dataset contains 348 images.
All images are 256px in height, 256px in width, in GRAY mode, and saved as png. Focus on a simple yet effective neural network architecture
suitable for beginners. You may use a pre-trained model and fine-tune it on the dataset. Include necessary imports, dataset loading and
preprocessing steps, model architecture, and training loop. The code should be ready to run in a Python interpreter without modifications.
datagen = ImageDataGenerator(rescale=1./255)
image_generator = datagen.flow_from_dataframe(dataframe=gitter_trainv1, x_col='image', y_col='label',
                                              target_size=(256, 256), color_mode='grayscale',
                                              class_mode='input', batch_size=32)
base_model = MobileNetV2(input_shape=(256, 256, 1), include_top=False)

x = base_model.output
x = Conv2D(128, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

model = Model(inputs=base_model.input, outputs=x)
model.compile(optimizer=Adam(), loss='binary_crossentropy')
model.fit(image_generator, epochs=10)
```

Fig. 6. Example prompt using *Jupyter Magic* with *GPT4*'s response to generate the necessary code for solving the image segmentation task. Shortened for brevity

JupyterAI through *GPT4All nous-hermes-llama2-13b* (Hermes) and *mistral-7b-openorca* (Orca) are used by default, as two representatives with varying parameter sizes and instruction and chat model respectively. Of these, we include the best-performing model in the evaluation. The experiment begins with a single prompt (Figure 6) to solve the problem in one shot. If this prompt does not create a valid ML pipeline, e.g., due to errors in the created code, the prompt is modified, for example, to split the problem into several prompts. In this case, it is tracked to what extent additional effort is required from the user and how demanding the changes are. If no approach with open-source LLMs leads to success, *GPT4* is used.

5.2. Datasets

The first dataset [18] contains images of four different types of surface defects. The images were collected via Laser-scanning microscopy to obtain height profiles on the surfaces and produce an artificial RGB image by a threefold repetition of the height data. This data is evaluated for **object detection**. While the authors dealt with Few-Shot Object Detection [3], these solutions are not included in the AutoML libraries, and except for *GPT*4, none of the LLMs had information about such models. Hence, we compare to the *RetinaNet* baseline given in the original paper [18].

For the same dataset [18] of surface defects optical images are available for **image classification**. Here we limit the number of available training samples per class to simulate a Few-Shot setting.

The next dataset [48] consists of images of scratch tests on surfaces to be evaluated for **image segmentation**. Zhang et al. identify the area where the so-called cross-cut test has removed the surface material. A binary mask with the marked regions is available for each image. They use several image augmentations to increase the amount of training data and make the model more robust. For this dataset, only the original images are available, without augmentations. However, feature engineering using *CAAFE* is performed.

The last dataset comes from [19], where methods for analyzing the resistance of coatings and paints are discussed again. The breakthrough point of a needle on a coating is identified by using sensor data about the movement and the force exerted by the needle on the surface. Using this data, the authors conduct **tabular regression** to find the position of the penetration through the surface. Based on the force exerted on the needle, they infer the scratch resistance of the evaluated coating.

Criteria A series of evaluation criteria are defined to conduct a structured evaluation of how effective the AutoML- & LLM-based approaches are in solving various ML tasks within the SEDAR system. If, for example, the effort is evaluated, more effort corresponds to a less favorable value, which is why an ascending color coding is selected (■■■■■■). If the completeness of a solution is being evaluated, the process is reversed; higher values are better (■■■■■■).

The performance evaluates the predictive capabilities of the models created by the AutoML solutions. Various metrics are considered, e.g. *Mean Average Precision* (MAP) for object detection, and *Intersection over Union* (IuO) for image segmentation. Where possible, the values are determined using the built-in functions of the AutoML library. If the appropriate metric is not implemented, this is highlighted negatively and a corresponding implementation is imported.

For efficiency, the time and resources required to experiment cover the process from data selection to the finished model. The time calculation in turn includes a) the preparations for the experiment (t_{prepare}), such as the dialogs, b) the time spent in the source code (t_{code}), for example by adapting function parameters, and c) the computing time required (t_{compute}). The complexity of the configuration of an experiment or the effort required for any changes to the code is measured by the following color-coded scale as *work*: (1) immediate execution, (2) simple, (3) medium, (4) extensive configuration and (5) individualized configuration with external resources.

The amount of prior knowledge and programming skills required for the configuration and changes to the code for a functioning pipeline is rated.

Finally, the initial model created by the AutoML process is assessed. A good starting situation either directly provides a versatile selection of models and parameters that the user can continue to use, or, especially when working with LLMs, provides sufficient hints to improve the designed pipeline.

5.3. Results of AutoMLWrapper and LLM-based Pipelines

Table 1. Overview of the experiments related to object detection. Only *AutoGluon* can perform object detection. The open-source models were unable to deal with this task, hence *GPT*4 had to be involved. Distribution of train/test sets: 140/36
More effort corresponds to a less favorable value, thus ascending color coding (🟩🟩🟨🟧🟥).
For completeness of a solution the process is reversed; higher values are better (⬛⬛⬛⬛⬛).

	$\text{Wrapper}_{\text{gluon}}$	LLM_{gpt4}	[18]
MAP	0.46	0.4	0.458
t_{prepare}(min.)	2	1	
t_{code}(min.)	< 1	10	
t_{compute}(min.)	60	12	
work	🟩⬜⬜⬜⬜	🟩🟩🟨⬜⬜	
Prior knowledge	🟩⬜⬜⬜⬜	🟩🟩⬜⬜⬜	
Programming expertise	🟩⬜⬜⬜⬜	🟩🟩🟨⬜⬜	
Initial model	⬛⬛⬛⬛⬜	⬛⬛⬛⬛⬛	

Table 2. Overview of the experiments in the segmentation data set. Only *AutoGluon* can perform image segmentation. Distribution of train/val/test sets: 174/35/8

	$\text{Wrapper}_{\text{gluon}}$	LLM_{Orca}	[48]
IoU	0.82	0.91	0.96
t_{prepare}(min.)	3	1	
t_{code}(min.)	3	15	
t_{compute}(min.)	9	20	
work	🟩🟩⬜⬜⬜	🟩🟩🟨🟧⬜	
Prior knowledge	🟩🟩🟨⬜⬜	🟩🟩🟨⬜⬜	
Programming expertise	🟩⬜⬜⬜⬜	🟩🟩🟨⬜⬜	
Initial model	⬛⬛⬛⬜⬜	⬛⬛⬛⬜⬜	

The results of the experiments (Tables 1 to 3) show that the *AutoMLWrapper* can considerably simplify creating ML pipelines. All results are from a single run. The user only has to make minimal entries in the frontend and can still create powerful models, which is particularly advantageous for users without much experience. The *AutoMLWrapper* can autonomously train and evaluate models in a short time if the appropriate settings are made. The performances in terms of evaluation metrics are competitive compared to the

Table 3. Overview of the image classification experiments with limited training data. *AutoGluon*'s few-shot setting leads to the best results. Distribution of train/test sets: first row 20/79 and second row 80/79

	$\text{Wrapper}_{\text{gluon}}$	$\text{Wrapper}^{\text{few-shot}}_{\text{gluon}}$	$\text{Wrapper}_{\text{keras}}$	$\text{LLM}_{\text{Hermes}}$
F1	0.62	0.87	0.6	0.5
F1	0.85	0.99	0.6	0.55
t_{prepare}(min.)	2	2	2	2
t_{code}(min.)	< 1	<1	15	10
t_{compute}(min.)	< 1	< 1	13	17
work	■□□□□	■□□□□	■■■□□	■■■■□
Prior knowledge	■□□□□	■□□□□	■■■□□	■■■■□
Programming expertise	■□□□□	■□□□□	■■□□□	■■□□□
Initial model	■■■■□	■□□□□	■■■□□	■■■■□

given baselines for both the *AutoMLWrapper* and the LLMs. Especially for the classification of paint defects, the training times were remarkably short, which made it possible to quickly create a selection of prototypes. Although the expected user-friendliness is a strength, it also results in reduced flexibility. The user has less control over specific model parameters and the optimization of these. In addition, the performance of the AutoML-Wrapper is directly dependent on the capabilities of the underlying libraries. In summary, the module offers an efficient and user-friendly way to quickly perform experiments that also result in powerful models.

5.4. Results of the KG-enhanced Approach

Table 4. Overview of the regression experiments (1 & 2) on tabular data. Distribution of train/val/test sets: 204/10/41

	$\text{Wrapper}_{\text{gluon}}$	$\text{Wrapper}_{\text{keras}}$	$\text{Wrapper}_{\text{sklearn}}$	LLM_{Orca}	[19]	KGE_1	KGE_2
Mean-Absolute-Error	8.86	64.68	7.38	53.39	4.58	15.67	0.69
t_{prepare}(min.)	2	2	2	2		1	1
t_{code}(min.)	<1	<1	<1	5		< 1	< 1
t_{compute}(min.)	<1	7	60	18		< 1	25
$work_{\text{code}}$	■□□□□	■□□□□	■□□□□	■■□□□		■□□□□	■□□□□
Prior knowledge	■□□□□	■□□□□	■□□□□	■□□□□		■□□□□	■□□□□
Programming expertise	■□□□□	■□□□□	■□□□□	■□□□□		■□□□□	■□□□□
Initial model	■■■■□	■■■□□	■■■■■	■■■■□		■□□□□	■□□□□

In Table 4, we present an evaluation of the selected dataset for tabular regression using three approaches: the *AutoMLWrapper*, LLM-orchestrated ML code generation, and our proposed knowledge-graph–enhanced method (KGE). Since OpenML is primarily designed for tabular data and contains only limited image-related datasets, this regression task represents the sole case study we report for the KGE approach. The corresponding output is illustrated in Figure 7, which shows the report generated by the AssistML backend when extended with the MLSea knowledge graph. The report lists several candidate

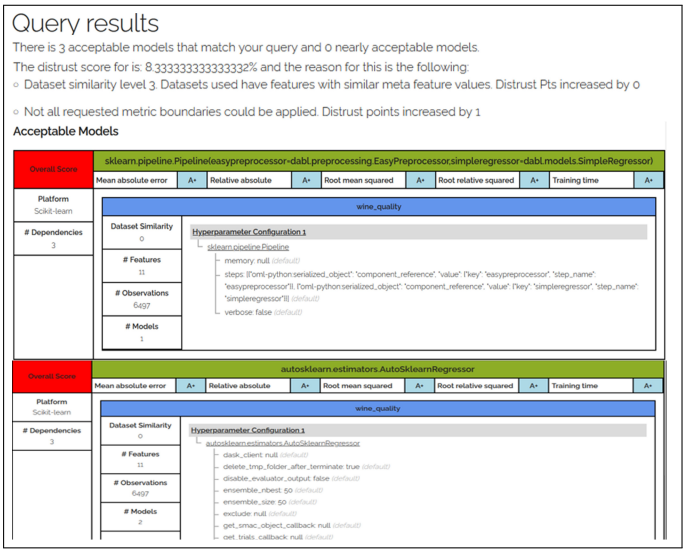


Fig. 7. A report with two model recommendations generated by AssistML [46] for the tabular regression dataset.

models, ordered by decreasing relevance. Notably, the top-ranked recommendation exhibits poor predictive performance, whereas the second-ranked recommendation, an AutoSklearn model, achieves near-perfect alignment with the test data, outperforming even the baseline established in [19]. The superior regression performance of the configuration recommended by AssistML can be primarily attributed to the explicit configuration which departs from the AutoMLWrapper by dramatically increasing the memory limit (102 GB vs. the 4 GB default) and replacing cross-validation with a holdout strategy, thereby enabling more complex regressors to be evaluated within the time budget. Together with constraining the ensemble to the top 50 models, these adjustments substantially broaden the effective search space and account for the observed reduction in regression error. However, due to this configuration, the execution time is much longer.

5.5. Discussion

Applying LLMs for code generation and process understanding in ML represents an innovative and promising approach. The open-source models offer the advantages of free availability, easy access to experimentation without direct costs, and special hardware. However, the open-source models' generated answers were often incomplete or lacked precision for a direct application. Many times the code had errors, was not executable. This is also true for older proprietary models such as GPT3.5 (see this notebook for a comparison between various open-source models and GPT3.5, 4o & 5). For the AutoSklearn model as recommened by the KGE method (see fig. 7), GPT3.5 ommited important model configurations. Especially for more complex queries for model creation and optimization, the answers were cut off prematurely or were so erroneous that considerable effort was required from the user. The results indicated that the models have difficulty understanding

and adapting to specific requirements and contexts. This led to generic or inappropriate code, especially for specialized tasks such as image segmentation. In contrast, *GPT*4 provided a significantly improved experience, generating more precise and relevant code that was better adapted to the problem. The answers were almost complete and pointed to additional aspects that could be used directly as an initial model for further questions to improve the model. For example, the open-source models were unable to utilize the COCO format of the object detection dataset, which was no problem for *GPT*4.

6. Conclusion

In this paper, we explored the integration of AutoML, LLMs, and knowledge graphs to enhance the automation of data science workflows in a data lake environment. While classical AutoML frameworks provide a structured approach to model selection and optimization, they lack flexibility in handling complex, domain-specific tasks. LLMs, on the other hand, offer a more adaptable solution for automating various stages of the ML pipeline but suffer from hallucinations, opaqueness, and difficulty in managing dependencies between tasks.

The evaluation, conducted on real-world industrial datasets, demonstrated that AutoML simplifies pipeline creation but limits user control, while LLM-based approaches require significant manual intervention to refine generated solutions. The findings indicate that neither AutoML nor LLMs alone can fully address the challenges of ML automation, highlighting the need for hybrid approaches.

To address these limitations, we propose a unified approach that integrates knowledge graphs with LLM-based data science agents, leveraging structured knowledge to improve model recommendations. By incorporating *MLSea*, a knowledge graph that aggregates ML datasets, tasks, and models, we introduce a recommendation pipeline that leverages historical ML knowledge for enhanced automation. Initial experiments indicate that incorporating structured machine-learning knowledge into data-science agents enhances the accuracy of automated ML workflows, thereby providing a successful proof of concept that will be subjected to comprehensive evaluation in a subsequent study.

Future work will focus on refining the synergy between these technologies, enhancing the explainability of recommendations, and optimizing multi-agent collaboration for end-to-end ML automation. The combination of the strengths of both frameworks could result in a new ML agent, based on *MLSea* and the derived knowledge, which could propose both an appropriate method for a new dataset and the required hyperparameters. Integrated into a data science agent, such as the *Data Interpreter*, this system has the potential to significantly enhance the efficiency, accuracy and explainability of end-to-end automated ML pipelines.

Acknowledgments. This work has been sponsored by the German Federal Ministry of Education and Research in the funding program "Forschung an Fachhochschulen", project i^2DACH (grant no. 13FH557KX0). We thank Vincent Hermann for his contribution.

References

1. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X.: A survey on evaluation of large language models. ACM Trans. Intell. Syst. Technol. 15(3) (mar 2024)
2. Chen, A., Dohan, D.M., So, D.R.: Evoprompting: Language models for code-level neural architecture search (2023)
3. Chen, T., Liu, Y., Su, H., Chang, Y., Lin, Y., Yeh, J., Hsu, W.H.: Dual-awareness attention for few-shot object detection. CoRR abs/2102.12152 (2021)
4. Dasoulas, I., Yang, D., Dimou, A.: Mlsea: A semantic layer for discoverable machine learning. In: European Semantic Web Conference. pp. 178–198. Springer (2024)
5. Durković, M., Lukovac, P., Hadžić, D., Barać, D., Bogdanović, Z.: Data-driven traffic management: Enhancing road safety through integrated digital twin technology. Computer Science and Information Systems (00), 58–58 (2025)
6. Galluzzo, Y.: A comprehensive review of the data and knowledge graphs approaches in bioinformatics. Computer Science and Information Systems 21(3), 1055–1075 (2024)
7. Grosnit, A., Maraval, A., Doran, J., Paolo, G., Thomas, A., Beevi, R.S.H.N., Gonzalez, J., Khandelwal, K., Iacobacci, I., Benechehab, A., et al.: Large language models orchestrating structured reasoning achieve kaggle grandmaster level. arXiv preprint arXiv:2411.03562 (2024)
8. Gu, K., Shang, R., Jiang, R., Kuang, K., Lin, R.J., Lyu, D., Mao, Y., Pan, Y., Wu, T., Yu, J., et al.: Blade: Benchmarking language model agents for data-driven science. arXiv preprint arXiv:2408.09667 (2024)
9. Guo, S., Deng, C., Wen, Y., Chen, H., Chang, Y., Wang, J.: DS-agent: Automated data science by empowering large language models with case-based reasoning. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 16813–16848. PMLR (2024)
10. Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X.: Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680 (2024)
11. Hassan, M.M., , et. al.: ChatGPT as your personal data scientist (2023)
12. Hogan, A., Dong, X.L., Vrandečić, D., Weikum, G.: Large language models, knowledge graphs and search engines: A crossroads for answering users' questions. arXiv preprint arXiv:2501.06699 (2025)
13. Hollmann, N., et al.: Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. Advances in Neural Information Processing Systems 36 (2024)
14. Hong, S., Lin, Y., Liu, B., Liu, B., Wu, B., Zhang, C., Wei, C., Li, D., Chen, J., Zhang, J., et al.: Data interpreter: An llm agent for data science. arXiv preprint arXiv:2402.18679 (2024)
15. Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., Zhou, L., et al.: Metagpt: Meta programming for multi-agent collaborative framework. arXiv preprint arXiv:2308.00352 (2023)
16. Hoseini, S., Ali, A., Shaker, H., Quix, C.: SEDAR: A semantic data reservoir for heterogeneous datasets. CIKM '23, ACM (2023)
17. Hoseini, S., Burgdorf, A., Paulus, A., Meisen, T., Quix, C., Pomp, A.: Challenges and opportunities of llm-augmented semantic model creation for dataspaces. In: Meroño-Peñuela, A., Corcho, Ó., Groth, P., Simperl, E., Tamma, V., Nuzzolese, A.G., Poveda-Villalón, M., Sabou, M., Presutti, V., Celino, I., Revenko, A., Raad, J., Sartini, B., Lisena, P. (eds.) The Semantic Web: ESWC 2024 Satellite Events - Hersonissos, Crete, Greece, May 26-30, 2024, Proceedings, Part II. Lecture Notes in Computer Science, vol. 15345, pp. 183–200. Springer (2024), https://doi.org/10.1007/978-3-031-78955-7_17

18. Hoseini, S., et al.: Automated defect detection for coatings via height profiles obtained by laser-scanning microscopy. MLWA 10, 100413 (2022)
19. Hoseini, S., et. al.: Coatings intelligence: Data-driven automation for chemistry 4.0. In: 2024 IEEE 7th (ICPS). pp. 1–8 (2024)
20. Hoseini, S., Ibbels, M., Quix, C.: Enhancing machine learning capabilities in data lakes with automl and llms. In: Tekli, J., Gamper, J., Chbeir, R., Manolopoulos, Y. (eds.) Advances in Databases and Information Systems - 28th European Conference, ADBIS 2024, Bayonne, France, August 28-31, 2024, Proceedings. Lecture Notes in Computer Science, vol. 14918, pp. 184–198. Springer (2024), https://doi.org/10.1007/978-3-031-70626-4_13
21. Hoseini, S., Theissen-Lipp, J., Quix, C.: A survey on semantic data management as intersection of ontology-based data access, semantic modeling and data lakes. Journal of Web Semantics 81, 100819 (2024)
22. Hu, L., Liu, Z., Zhao, Z., Hou, L., Nie, L., Li, J.: A survey of knowledge enhanced pre-trained language models. IEEE Transactions on Knowledge and Data Engineering (2023)
23. Huang, Q., Vora, J., Liang, P., Leskovec, J.: Benchmarking large language models as ai research agents (2023)
24. Iglesias, E., Jozashoori, S., Vidal, M.E.: Scaling up knowledge graph creation to large and heterogeneous data sources. Journal of Web Semantics 75 (2023)
25. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Comput. Surv. 55(12) (mar 2023)
26. Karmaker, S.K., et al.: Automl to date and beyond: Challenges and opportunities. ACM Comput. Surv. 54(8) (oct 2021)
27. Khan, Y., Zimmermann, A., Jha, A., Gadepally, V., d'Aquin, M., Sahay, R.: One size does not fit all: querying web polystores. Ieee Access 7, 9598–9617 (2019)
28. Kreuzberger, D., Kühl, N., Hirschl, S.: Machine learning operations (mlops): Overview, definition, and architecture. IEEE Access 11, 31866–31879 (2022)
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25 (2012)
30. Li, D., Tan, Z., Liu, H.: Exploring large language models for feature selection: A data-centric perspective. ACM SIGKDD Explorations Newsletter 26(2), 44–53 (2025)
31. Li, X., Döhmen, T.: Towards efficient data wrangling with llms using code generation. In: Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning. pp. 62–66 (2024)
32. Liu, S., Gao, C., Li, Y.: Large language model agent for hyper-parameter optimization. arXiv preprint arXiv:2402.01881 (2024)
33. Malberg, S., Mosca, E., Groh, G.: Felix: Automatic and interpretable feature engineering using llms. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 230–246. Springer (2024)
34. Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., Tang, S.: Graph retrieval-augmented generation: A survey. arXiv preprint arXiv:2408.08921 (2024)
35. Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems 36 (2024)
36. Singh, S.K., Cha, J., Kim, T.W., Park, J.H.: Machine learning based distributed big data analysis framework for next generation web in iot. Computer Science and Information Systems 18(2), 597–618 (2021)
37. Sun, M., Han, R., Jiang, B., Qi, H., Sun, D., Yuan, Y., Huang, J.: A survey on large language model-based agents for statistics and data science. arXiv preprint arXiv:2412.14222 (2024)
38. Sun, Y., Yan, B., Shao, Z.: Ai large models bring great opportunities to reusable design of cad software. Computer Science and Information Systems (00), 46–46 (2024)
39. Sutskever, I.: Sequence to sequence learning with neural networks: What a decade. https://www.youtube.com/watch?v=WQQdd6qGxNs, access on 19.02.2025 (2024), neurIPS 2024

40. Tornede, A., et al.: Automl in the age of large language models: Current challenges, future opportunities and risks. arXiv preprint arXiv:2306.08107 (2023)
41. Trirat, P., Jeong, W., Hwang, S.J.: Automl-agent: A multi-agent llm framework for full-pipeline automl. arXiv preprint arXiv:2410.02958 (2024)
42. Truong, A., Walters, A., Goodsitt, J., Hines, K., Bruss, C.B., Farivar, R.: Towards automated machine learning: Evaluation and comparison of automl approaches and tools. In: 2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI). pp. 1471–1479. IEEE (2019)
43. Wei, L., He, Z., Zhao, H., Yao, Q.: Unleashing the power of graph learning through llm-based autonomous agents (2023)
44. Yang, R., Liu, H., Zeng, Q., Ke, Y.H., Li, W., Cheng, L., Chen, Q., Caverlee, J., Matsuo, Y., Li, I.: Kg-rank: Enhancing large language models for medical qa with knowledge graphs and ranking techniques. arXiv preprint arXiv:2403.05881 (2024)
45. Yang, Z., , et. al.: Autommlab: Automatically generating deployable models from language instructions for computer vision tasks (2024)
46. Zacarias, A.G.V., Reimann, P., Weber, C., Mitschang, B.: Assistml: an approach to manage, recommend and reuse ML solutions. Int. J. Data Sci. Anal. 16(4), 455–479 (2023), https://doi.org/10.1007/s41060-023-00417-5
47. Zaharia, M., et al.: Accelerating the machine learning lifecycle with mlflow. IEEE Data Eng. Bull. 41(4), 39–45 (2018)
48. Zhang, G., et al.: Deep learning-based automated characterization of crosscut tests for coatings via image segmentation. JCTR 19, 671–683 (2021)
49. Zhang, K.: Using deep learning to automatic inspection system of printed circuit board in manufacturing industry under the internet of things. Computer Science and Information Systems 20(2), 723–741 (2023)
50. Zhang, L., Zhang, Y., Ren, K., Li, D., Yang, Y.: Mlcopilot: Unleashing the power of large language models in solving machine learning tasks. arXiv preprint arXiv:2304.14979 (2023)
51. Zhang, S., Gong, C., Wu, L., Liu, X., Zhou, M.: Automl-gpt: Automatic machine learning with gpt. arXiv preprint arXiv:2305.02499 (2023)

Sayed Hoseini is a researcher at the Hochschule Niederrhein University of Applied Sciences (HSNR) pursuing a PhD degree from RWTH Aachen University, Germany. His research focuses on data architectures specifically data lakes, data integration, and data management for machine learning. He has served as a program committee member of database conferences such as CIKM, ADBIS, and as journal reviewer for MLWA.

Maximilian Ibbels is a software engineer who graduated in 2024 from HSNR with a Master of Science in Computer Science. His master's thesis focused on extending the SEDAR data lake system with automated machine learning capabilities. He developed and evaluated the AutoMLWrapper and implemented LLM-based code generation to support automated ML model creation.

Maximilian Knoll is a software engineer who graduated in 2025 from HSNR with a Master of Science in Computer Science. His master's thesis focused on developing an LLM-based Data Science agent for automated machine learning model generation. He integrated the MLSea knowledge graph with the AssistML framework to create a semantics-aware agent capable of supporting and automating the ML model development process.

Christoph Quix is professor for Information Systems and Data Science at HSNR. He is also lead scientist for smart data management and analytics in the Fraunhofer Institute for Applied Information Technology FIT. His research focuses on data integration, data science, management of large, heterogeneous data sets, and metadata management in different application areas, e.g, chemistry 4.0 and industry 4.0. He has more than 100 publications in international journals and conferences.

Received: March 20, 2025; Accepted: November 15, 2025.

Computer Science and Information Systems 23(1):443–473 https://doi.org/10.2298/CSIS250303077B

Towards Explainable Sequential Learning*

Giacomo Bergami[1], Emma Packer[2], Kirsty Scott[2], and Silvia Del Din[2,3]

[1] School of Computing, Newcastle University
giacomo.bergami@newcastle.ac.uk
[2] Translational and Clinical Research Institute, Faculty of Medical Sciences, Newcastle University
{e.packer,kirsty.scott-singer}@newcastle.ac.uk
[3] National Institute for Health and Care Research (NIHR), Newcastle Biomedical Research Centre (BRC), Newcastle University and The Newcastle upon Tyne Hospitals NHS Foundation Trust
silvia.del-din@newcastle.ac.uk

Abstract. This paper offers a hybridly explainable temporal data processing pipeline, DataFul Explainable MultivariatE coRrelatIonal Temporal Artificial inTElligence (EMeriTAte+DF), bridging numerical-driven temporal data classification with an event-based one through verified artificial intelligence principles, enabling human-explainable results. This was possible through a preliminary *a posteriori* explainable phase describing the numerical input data in terms of concurrent constituents with numerical payloads. This further required extending the event-based literature to design specification mining algorithms supporting concurrent constituents. Our previous and current solutions outperform state-of-the-art algorithms for multivariate time series classifications over four dataset considered in the present paper, thus showcasing the effectiveness of the proposed methodology premiering the extraction of explainable correlations across Multivariate Time Series (MTS) dimensions with dataful features.

Keywords: Verified AI; eXplainable AI (XAI); polyadic logs; Data Trends; Poly-DECLARE; Multivariate Time Series Classification

1. Introduction

The current literature on temporal data analysis is divided into two main branches: one, mainly considering numerical fluctuations in different dimensions as Multivariate Time Series (MTS) [18,32,23,43], and the other focusing on a characterization as a linear sequence of discrete events, be they durative [3,37,5] or pointwise [17,11]. The first class denotes the classification task as Multivariate Time Series Classification (MTSC), which requires capturing behavioural correlations between different dimensions [38]. Cross-dimension correlations can be represented via convolutions [43] or kernels that potentially squash such correlations to a single numerical value [18]. Despite the possibility of doing so, the explanations thus obtained are not easily understandable by humans. Furthermore, recent results have shown that these solutions cannot capture correlations when considering complex and multifactorial problems, such as clinical ones, and give results with low accuracy [16]. These considerations require us to solve the problem from a different angle.

* This journal paper constitutes a major extension of our previous work [16].

Recent approaches are starting to bridge the numerical with the event-driven characterization of temporal data by clearly showing the possibility of discretizing Data Trends (DT) occurring into durative constituents [24]. Notwithstanding this, these events cannot summarize numerical information precisely characterizing the trend of choice, thus only characterizing growth, variability, or decrease patterns without providing any further numerical references to better describe the event of choice. This demands a more general representation of DT patterns, which should also encompass numerical features, albeit summarized (e.g. 22 CAnonical Time-series CHaracteristics (catch22) [30]), to characterize the underlying data from a more data-driven perspective.

Within the second branch, Business Process Management [1,17] enables, through declarative process mining, to temporally correlate single pointwise events through a human-readable data description while bridging across different *activity labels* (distinguishing different action/event types) occurring. This literature premiered the extraction of declarative specifications from event-based data while enriching information concerning temporal correlations with additional numerical [27] and categorical [14] data predicates. Despite the inherent ability of such techniques to provide a human-explainable characterization of the data, the former techniques cannot be straightforwardly applied to solve the MTSC problem, even after DT-discretization. *First*, despite the possibility of representing temporal specifications as concurrent ones [1], these are always interpreted as a temporally-ordered succession of non-concurrent and non-overlapping events [15], thus invalidating the possibility of directly exploiting the same algorithms for capturing multiple events co-occurring at the same time across dimensions. In fact, despite the possibility for the current XES [2] standard to express durative events, this cannot clearly represent and group together co-occurring events within the same trace across dimensions. *Second*, as such techniques often involve small and curated datasets, they have been proven several times as not scalable over real-world ones [11]. At the time of the writing, the most efficient temporal specification miner, Bolt2 [11], extracts data-less temporal clauses and, therefore, does not consider data payload information associated with each activity label, as required after extending DT with data payloads. As this was the one used in our previous work to characterize correlations across temporal dimensions after a minor extension for supporting durative and co-occurring events [16], the extraction of a declarative dataful specification requires to expand the latter algorithm further also to enable the extraction of dataful features from the declarative clauses.

This paper aims to bridge all approaches above to achieve a generally explainable sequential learning explanation through DataFul Explainable MultivariatE coRrelatIonal Temporal Artificial inTElligence (EMeriTAte+DF)[4]. The proposed technique works by first discretizing MTS into machine-readable polyadic logs through dataful DT mining (Sect. 4.1), for then extracting declarative descriptions of the previously-mined representation while pertaining dataful features (Sect. 4.2). We then use such clauses to derive the features satisfied, violated, or vacuously satisfied by the traces, for then deriving a propositional representation of the classes associated with arbitrary MTS segments through white-box classifier explanation extraction (also Sect. 4.2). By doing so, each temporal class is then defined through the satisfaction of a proportional formula, where predicates state temporal properties of the data. This approach further generalises pre-existing de-

[4] `https://github.com/datagram-db/knobab/releases/tag/v3.1`

viance mining solutions, which only encompass the conjunctive-based characterization of such temporal classes [14].

This constitutes a major improvement over our previous solution Explainable MultivariatE coRrelatIonal Temporal Artificial inTElligence (EMeriTAte) [16], where *(i)* the original implementation of DT mining was not extracting a numerical summarised description of the data and *(ii)* was using a straightforward brute-force algorithm for considering all the possible sliding windows of all possible sizes; *(iii)* the extraction of the declarative clauses for each polyadic log (Sect. 3.1) did not consider dataful clauses, and *(iv)* the generation of the embedding from the mined clauses was regarded as a distinct phase from the one where the trace embeddings were generated as in previous literature [14], thus adding significant computational overhead. The application of the aforementioned changes to our previous solution enables us to design its dataful variant (EMeriTAte+DF) outperforming EMeriTAte in terms of efficiency, accuracy, precision, and recall. Last, we extend our experimental set-up to consider other MTS datasets, thus including univariate ones [26,21] and others related to general human mobility, thus remarking the inherent difficulty of characterizing dyskinetic/off events in Parkinson's Disease patients.

1.1. Notation

The following table provides the notation used in the rest of the paper.

Mathematical Notation	
$\langle \dots \rangle$	sequence of index-ordered elements
$\{\dots\}, \emptyset$	set (bag) of elements, empty set
$D, D_{\text{train}}, D_{\text{test}}$	dataset, train d., test d.
$\mathbb{N} = \{0, 1, \dots\}$	set for natural numbers
$\mathbb{R}, \mathbb{R}^d$	sets for real numbers and d-dimensional real-valued vectors (resp.)
$\lvert S \rvert \in \mathbb{N}$	cardinality (or rank, size) of S
$f \colon A \to B$	function f with domain in A and codomain in B
$\tilde{p} \colon A \rightharpoonup B$	finite function $\tilde{p}$ with finite domain in A and codomain in B
$\mathrm{dom}(f) = A$	domain functor dom, it returns f's domain A
$\tilde{p} = [a \mapsto b, c \mapsto d]$	using the graph of a function to define a finite function $\tilde{p}$ with $\mathrm{dom}(\tilde{p}) = \{a, b\}$: $\tilde{p}(a) = b$ and $\tilde{p}(c) = d$
$[b, e] \simeq \langle b, b+1, \dots, e \rangle$	numeric iterval, isomorphic to a sequence of natural numbers
$f \oplus g$	finite function concatenation [8]
$\mathbf{1}_b = \begin{cases} 1 & b = \mathbf{true} \\ 0 & \text{oth.} \end{cases}$	indicator function
$\mathbb{B}(b) = \mathbf{1}_b - \mathbf{1}_{\neg b}$	A function mapping **true** boolean b values to 1 and **false** ones to -1.
$\mathbb{I}_F([B, E])$	set of maximal non-overlapping intervals in [B,E] sharing the same F values (Eq. 1)
Time Series	
$t, B, E, b, e, \beta, \eta$	(point) time indices (in $\mathbb{N}$)

$x \in \{\mathtt{i}, \mathtt{a}, \mathtt{s}, \mathtt{v}\}$	(numerical) condition index (Table 2)
$T \colon \mathbb{N} \rightharpoonup \mathbb{R}$	univariate Time Series (TS)
T^x	TS X satisfying a numerical condition x (string)
$T(t)$	TS value at time t
$T[\![B, E]\!] = \langle T(B), \ldots, T(E) \rangle$	univariate projection
$\mathfrak{E}, \tau \colon \mathbb{N} \rightharpoonup \mathbb{R}^d$	d-dimensional Multivariate Time Series (MTS) (or environment)
$\psi \in [1, d]$	MTS dimension index
$\mathfrak{E}_\psi$	TS representing MTS $\mathfrak{E}$ values for dimension ψ
$\mathfrak{E}^x_\psi$	MTS $\mathfrak{E}$ satisfying a numerical condition x over dimension ψ (string)
$\mathfrak{E}_\psi(t), \mathfrak{E}(t)(\psi)$	MTS value at time t and dimension ψ
$\mathfrak{E}_d(t) = \mathfrak{E}(t)(d) = \chi$	MTS being in class χ at time t ($d \geq 2$)
$\mathfrak{E}[B, E]$	MTS projection over a time interval $[B, E]$ (Eq. 2)
$val^{\mathfrak{E}}_x(t)$	numerical condition x value for MTS $\mathfrak{E}$ at time t (Table 2)
$P^{\mathfrak{E}}_x(t)$	point-wise predicate for MTS $\mathfrak{E}$ and condition x at time t (Table 2)

Polyadic Logs, DECLARE [36]

φ	DECLARE clause index (specification)
$i, j, k \in \mathbb{N}$	trace, event, and constituent index (resp.)
$\mathsf{A}, \mathsf{B}, T^{\mathtt{i}}, \mathsf{IncreaseRapidly}(T^{\mathtt{i}})$	activity labels (or strings)
$*$	always-matching activity label (or jolly, wildcard)
p, p', q, q'	data conditions
c_l	DECLARE template (string, Fig. 1b)
$c_l(\mathsf{A}, p)$	unary DECLARE clause
$c_l(\mathsf{A}, p, \mathsf{B}, q),\ c_l(\mathsf{A}, p, \mathsf{A'}, p')$	binary DECLARE clauses
$\Phi, \Phi_1, \Phi_2, \ldots$	set of specifications (with conjunctive interpretation)
$G_\psi = \langle N_{G_\psi}, R_{G_\psi}, \ell_{G_\psi} \rangle$	ℓ_{G_ψ}-rooted Direct Acyclic Graph with nodes in N_{G_ψ} and edges in R_{G_ψ} [12]
$\varsigma^i_{j,k} = \langle \mathsf{A}, \tilde{p}, \mathfrak{s} \rangle$	k-th constituent for p. event σ^i_j, with an activity label A, a data payload $\tilde{p}$, and a duration span of $\mathfrak{s}$
$\lambda(\varsigma^i_{j,k}) = \mathsf{A}$	activity label projection
$\varpi(\varsigma^i_{j,k}) = \tilde{p}$	payload projection
$\delta(\varsigma^i_{j,k}) = \mathfrak{s}$	span projection
$\kappa(\varsigma^i_{j,k})$	information concatenation (Eq. 3)
$\sigma^i_j = \langle \langle \varsigma^i_{j,1}, \ldots, \varsigma^i_{j,m} \rangle, \chi \rangle$	j-th poliadic event of p. trace σ^i with m constituents and class χ
$\sigma^i = \langle \sigma^i_1, \ldots, \sigma^i_o \rangle$	i-th polyadic trace, with $\lvert\sigma^i\rvert = o$
$\mathcal{L} = \langle \sigma^1, \ldots, \sigma^n \rangle$	sequence of n polyadic traces
$\mathfrak{S} = \langle \mathbb{G}, \mathcal{L} \rangle, \mathbb{G} = \{G_1, \ldots, G_n\}$	polyadic log (or system), taxonomies' set [35]
Σ	Alphabet, set of activity labels occurring in a dataset of logs (Algorithm 2, Line 2)

Algorithms	
$s \| \| t$	string concatenation[5], with implicit operands' casting
yield v	A function containing a **yield** statement is a generator function, returning all the v-s in a collection in order of generation[6].
INITFRESH(n)	Initializes all the auto-increment natural-valued variables ζ_i to n for all $i \in \mathbb{N}$
`fresh`(k)	Returns the current value of ζ_k before incrementing it. The term fresh comes from "fresh variable" [20]
$l \leftarrow r$	The reference to l is updated with the value r. If l does not exist, it gets declared[7].
DF	DataFrame [31]: a double-entry table providing cell value access v through row i and column j index, $DF[i, j] = v$
ZIP($t_1, \dots$)= $\dots, \langle t_1(i), t_2(i), \dots \rangle, \dots$	zip returns a sequence which i-th element is the sequence containing all the operands' i-th elements[8].
$\mathcal{G}_{x,b}^{\psi,\mathfrak{E}}$	Indexing $\mathfrak{E}$ for all maximal contiguous (time) intervals for which the values from dimension ψ always satisfy ($b =$ **true**) $P_x^{\mathfrak{E}}$.

2. Related Works

2.1. Hybridly Explainable Artificial Intelligence

The observations from which the utility of defining a hybrid explainable AI is deduced have their origins in the formalization of verifiable AI systems. At the basis of this framework [39], we always consider a model of the system to be verified ($\mathfrak{S}$), a model of the environment ($\mathfrak{E}$), and the formal property to be verified (φ), where all of these components might be expressed in logical form, so to provide correctness guarantees. Specification mining algorithms [11] extract φ from given system or environment of interest $\mathfrak{S} \vee \mathfrak{E}$, where deviance mining algorithms [14] are a specific case of the former, assuming *(i)* that environment states can be labelled and *(ii)* that we can extract one single specification per environment label, which provides the characterizing behaviour distinguishing a class from the others. A later survey [13] generalised the former into *hybridly explainable artificial intelligence* after observing that we can always transform unstructured data representations into more structured and logical-driven by pre-processing the data into an *a priori* phase, which can include both data cleaning capabilities [4] and specification mining ones by enriching structured data representation with contextual interpretations of the raw data. After this, we can carry out the actual learning phase, in which *ad hoc*

[5] `https://datatracker.ietf.org/doc/html/rfc8017#section-2`

[6] PHP: `https://www.php.net/manual/en/language.generators.syntax.php`, Python: `https://docs.python.org/3/reference/expressions.html#yieldexpr`

[7] `https://www.overleaf.com/learn/latex/Algorithms`

[8] Python: `https://docs.python.org/3.3/library/functions.html#zip`

explainability will heavily depend on the methodology and the data-representation format of choice. Last, we can consider precision/recall metrics as well as model extractions from both white-box and black-box classifiers as *ex post* explanations of the learning phase. This paper will focus on providing algorithmic solutions for the first two phases, as narrated in the text.

To better explain the former framework, we will categorize the forthcoming sections by pigeonholing curring literature over the aforementioned framework.

2.2. A Priori Explanability

Numerical Data. An (univariate) TS $T\colon \mathbb{N} \rightharpoonup \mathbb{R}$ [22] of size $|T| = |\operatorname{dom}(T)|$ where $\operatorname{dom}(T) = \{1, \ldots, |T|\}$ is a finite function mapping time units $t \in \operatorname{dom}(T)$ into real values $T(t)$. Given a time interval $[b, e]$ and a TS X of at least size e, we denote $T[\![b, e]\!]$ as its *univariate projection* returning an ordered sequence $\langle T(b), T(b+1), \ldots, T(e)\rangle$. Thus, we consider intervals as a shorthand for an ordered sequence, $[b, e] \simeq \langle b, b+1, \ldots, e\rangle$. DT specifications (Fig. 1a) provide a $\mathrm{LTL_f}$ characterization for numerical increase $\mathtt{i}$ trends in (univariate) TS X after propositional discretization within a mining phase [24]: this requires a preliminary pre-processing step for which X is associated with durative constituents ς^T_{t+1} with activity labels $T^{\mathtt{i}}$ or $T^{\neg\mathtt{i}}$ depending on the value $T(t)$ of TS Tat time t: ς^T_{t+1} has activity label $T^{\mathtt{i}}$ ($\lambda(\varsigma^T_{t+1}) = T^{\mathtt{i}}$) if $T(t+1) > T(t)$ and $T^{\neg\mathtt{i}}$ otherwise. We can then determine the occurrence of a DT pattern by simply joining constituents $T^{\mathtt{i}}$ and $T^{\neg\mathtt{i}}$ into durative ones referring to the same temporal span, which can then be matched and rewritten into a durative event having the activity label described in the first column of Fig. 1a. Authors mainly use this declarative representation for TS forecasting purposes, but give no evidence for exploiting this in the context of MTS while correlating disparate DT across MTS variables, which might be expressed through DECLARE (see **Event Data**). To achieve this, our previous work premiered their combination. Although DT discards the numerical information associated with such trends, it generalises the common shapelet approach [23] to arbitrary growth and decrease patterns expressed declaratively with human-understandable patterns.

Event Data. DECLARE [36] (Fig. 1b) is a temporal declarative language considering possible temporal behaviour in terms of correlations between activated events and targeted conditions, where the former refers to the necessary but not sufficient condition for satisfying a clause. Both activation and target conditions are defined through *activity labels* referring to a specific action of interest; these can be enriched with dataful propositional formulæ, predicating payload conditions over the activated constituents. Unless stated otherwise, if the clause has no activation, it is trivially satisfied (*vacuously*). E.g., Precedence might also be vacuously violated due to the presence of just target conditions [11]. DECLARE clauses can be instantiated over a specific alphabet Σ referring to activity labels associated with constituents and they can be composed to generate *conjunctive specifications* $\Phi = \{\varphi_1, \ldots, \varphi_n\}$; we say that a trace satisfies a conjunctive specification if it satisfies all its clauses. These conjunctive specifications can be extracted from temporal data represented as a (non-polyadic) log by efficiently exploiting a specific-to-general lattice search from which we can preventively prune the generation of specific clauses [11].

Activity Label	TS	Trace $(T^{\mathtt{i}}, T^{\neg\mathtt{i}})$
IncreaseRapidly$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, \ldots, T^{\mathtt{i}}$
IncreaseSlowlyI$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, T^{\mathtt{i}}, \ldots, T^{\mathtt{i}}$
IncreaseSlowlyII$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, T^{\neg\mathtt{i}}, T^{\mathtt{i}}, \ldots, T^{\mathtt{i}}$
IncreaseSlowlyIII$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, \ldots, T^{\mathtt{i}}, T^{\neg\mathtt{i}}, T^{\mathtt{i}}$
IncreaseSlowlyIV$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, \ldots, T^{\mathtt{i}}, T^{\neg\mathtt{i}}$
HighVolatilityI$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, T^{\mathtt{i}}$
HighVolatilityII$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, T^{\mathtt{i}}, T^{\neg\mathtt{i}}, T^{\mathtt{i}}$
HighVolatilityIII$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, T^{\mathtt{i}}, \ldots, T^{\mathtt{i}}, T^{\neg\mathtt{i}}$
HighVolatilityIV$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, T^{\neg\mathtt{i}}, \ldots, T^{\neg\mathtt{i}}, T^{\mathtt{i}}$
HighVolatilityV$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, T^{\neg\mathtt{i}}, T^{\mathtt{i}}, T^{\neg\mathtt{i}}$
HighVolatilityVI$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, T^{\neg\mathtt{i}}$
DecreaseSlowlyI$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, \ldots, T^{\neg\mathtt{i}}, T^{\mathtt{i}}$
DecreaseSlowlyII$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, \ldots, T^{\neg\mathtt{i}}, T^{\mathtt{i}}, T^{\neg\mathtt{i}}$
DecreaseSlowlyIII$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, T^{\mathtt{i}}, T^{\neg\mathtt{i}}, \ldots, T^{\neg\mathtt{i}}$
DecreaseSlowlyIV$(T^{\mathtt{i}})$		$T^{\mathtt{i}}, T^{\neg\mathtt{i}}, \ldots, T^{\neg\mathtt{i}}$
DecreaseRapidly$(T^{\mathtt{i}})$		$T^{\neg\mathtt{i}}, \ldots, T^{\neg\mathtt{i}}$

(a) Expressing DT activity labels as TS silhouettes and pointwise constituents for the increase numerical condition, `i` [24].

Clauses φ from template c_l	Non-polyadic logs interpretation $[\![c]\!]$
†Init(A, p)	The trace should start with an activation
†End(A, p)	The trace should end with an activation
†Exists(A, p)	Activations should occur at least once
Absence(A, p)	Activations should never occur
Choice(A, p, A', p')	One of the two activation conditions must appear.
ExclChoice(A, p, A', p')	Only one type of activation is admitted per trace.
RespExistence(A, p, B, q)	An activation requires at least one target.
CoExistence(A, p, B, q)	$[\![$RespExistence(A, **true**, B, **true**)$]\!] \wedge [\![$RespExistence(B, **true**, A, **true**)$]\!]$
†Precedence(A, p, B, q)	Events preceding the activations should not satisfy the target.
†Response(A, p, B, q)	An activation leads to a future target event.
Succession(A, p, B, q)	$[\![$Precedence(A, **true**, B, **true**)$]\!] \wedge [\![$Response(A, **true**, B, **true**)$]\!]$
†ChainPrecedence(A, p, B, q)	An activation leads to an *immediately preceding* target.
†ChainResponse(A, p, B, q)	An activation leads to an *immediately following* target.
ChainSuccession(A, p, B, q)	$[\![$ChainPreced.(B, **true**, A, **true**)$]\!] \wedge [\![$ChainResponse(A, **true**, B, **true**)$]\!]$

(b) Our dataful DECLARE [36] subset of interest, where A (respectively, B) denote activation (resp., target), and p (resp., q) denote the associated dataful payload condition. conditions. Dataless variants can be expressed with p (p' and q) as **true**. Given an unary (respectively, binary) template c_l, c_l(A, p) (resp., c_l(A, p, B, q) or c_l(A, p, A', p')) denotes an unary (resp., binary) clause φ. †remarks clauses subject to dataful refinement in the present paper.

Fig. 1. Declarative Languages for *a priori* explanability over *(a)* TS or *(b)* non-polyadic logs.

Bolt2 [11] mines non-polyadic logs to extract conjunctive DECLARE specifications, where events are non-durative and where every single event contains one and only one constituent. The adoption of the KNOwledge Base for Alignments and Business process modelling (KnoBAB) [11] for storing the log and traces in primary memory allows fast constituent retrieval by activity label and trace scanning. This was possible as its primary goal was to maximize formal verification tasks over logs of traces. By offering indexed relational tables with temporal, payload, and event-counting data represented in a columnar way, we ensure fast data access for scanning multiple events occurring in a log. Bolt2 achieves efficient specification mining not only by the means of efficient data access via KnoBAB, but by also testing DECLARE dataless clauses where the most general clause is tested first, followed by the ones enabling the other. This generates a lattice of DECLARE clauses: the clause search stops by adopting a quality metric, thus ensuring the capture of the most specific behaviour of the single log without returning underrepresented ones as not generalizing over the data. This solution does not contemplate any further refinement of the clauses into dataful ones so to associate non-trivially true data predicates to either the activation, target, or correlation conditions. Other specification mining algorithms overcome this limitation [27] and, despite not using an efficient general-to-specific visit as the former, consider clustering over the activation conditions for deriving the data's propositional representations. This consideration was blindly adopted in Deviant Model Learning (DML) [14] to refine each mined declare clause to generate dataful clauses differentiating traces belonging to different classes. This poses a significant limitation in this scenario, as the mining approach was not initially designed for this purpose, thus not necessarily helping to refine the activation conditions according to the class associated with each single activation or target condition. This paper proposes a completely different approach, where white-box classifiers are used to characterize the activation (and target) predicates with the clear intent to capture distinct class behaviours and not merely aggregate data by shared data features, which might not be necessarily related to a class of interest.

2.3. Ad Hoc Explanability

Numerical Data. *This subsection describes the main competitors considered from this paper for EMeriTAte+DF, as they constitute MTS classifiers.*

(k-nearest neighbors (KNN)-based) Clustering for time series [25] works by first identifying time series clusters depending on a distance function of choice and then associating to each cluster the majority class; this induces the possibility of boiling down a clustering problem to a classification one by associating each numerical-based representation to the element being the most similar to the majority class similarly to [7]. Euclidean k-nearest neighbors clustering (E-KNN) uses a straightforward Euclidean metric can determine the pointwise distance of the MTS across dimensions. The main drawback of this is that it neither considers the evolution of DTs within the MTS nor allows for aligning similar trends being displaced in time, similar to dynamic time warping. Furthermore, no mechanism efficiently tests different patterns at stake by considering a hierarchy of temporal patterns, one being the generalization of the other. Doing so can greatly boost the mining task while increasing the overall number of temporal patterns of interest.

Rocket [18] exploits randomized convolutional kernels to extract relevant features from the MTS, which are then fed to a linear classifier by associating such features to

a class of interest. While this feature extraction approach cannot adequately capture DTs and variations across dimensions, it also guarantees scarce explainability as information is summarised into kernel values. The Multivariate Time Series Classifier with Attentional Prototypical Network (TapNet) [43] improves the latter by exploiting attention networks for selecting the best classification features while still heavily relying on up-front feature selection mechanisms such as dimensionality reduction, which might lose relevant information to the detriment of the classification precision and accuracy. Canonical Interval Forest (CIF) [32] achieves explainability by exploiting white-box classifiers such as decision trees over catch22 features [30] describing a selection of time intervals describing different types of numerical variations rather than changes in trends and data patterns and their temporal correlations. Shapelet Transform Classifier (STC) [23] characterize MTS in terms of distinctive temporal features, the *shapelets*, being frequently occurring trends occurring across all TS, for then describing each MTS in terms of their distance with each selected shapelet. Notwithstanding this approach considers numerical features that are completely discarded through DT, it fails to establish correlations across trends occurring in different dimensions, thus not establishing correlations across differently observed behaviours.

Event Data. DML [14] extracts specifications for trace classes $y \in \mathcal{Y}$ through declarative (e.g., DECLARE clauses) and procedural (i.e., association rules) features previously mined from the data. Each trace can be represented as an n-dimensional embedding $\vec{x}_\sigma \in \mathbb{R}^n$, where n is the overall number of features. E.g., when a Declare clause φ is chosen as the ψ-th feature, a value of $\vec{x}_\sigma[\psi] = -1/0/n$ denotes that the trace σ does not satisfy, or vacuously satisfies, or satisfactorily activates φ n times. For the features providing declarative characterizations, DML extracts those via the ADMS specification mining algorithm [11]. Our previous approach, EMeriTAte [16], generalised over the former algorithm in its *ad hoc* phase by considering a different log model, where each event (and not just a single trace) is associated with a class and where each event is composed by different *constituents* representing concurrent activities starting at the same time and with different durations. These characteristics also enable the Event Data algorithms to consider numerical-driven problems with different dimensions. At the same time, previous temporal declarative-driven approaches were limited to only considering univariate TS [19], and were only used for verification tasks, but not for specification mining ones. We further expand on this data model to consider multiple taxonomies of activity labels for considering which events might be regarded as sub-events of one more general type (Sect. 3.1), so as to reduce better the amount of relevant correlations (Sect. 4.2).

3. Problem Statement

We define $\mathbb{B}(b) = \mathbf{1}_b - \mathbf{1}_b$ where $\mathbf{1}_b$ is the indicator function for predicates p. Given a function F having as a domain a closed interval in $\mathbb{N}$, the set $\mathbb{I}_F([B, E])$ of the maximal intervals in $[B, E] \subseteq \mathbb{N}$ is a set of largest non-overlapping intervals in $[B, E]$ sharing the same value for F where all the remaining intervals will have different values in F:

$$\mathbb{I}_F([B,E]) = \{[b,e] \mid B \leq b \leq e \leq E, (b > B \Rightarrow F(b) \neq F(B)), \\ (e < E \Rightarrow F(e) \neq F(E)), \forall b \leq t \leq e . F(t) = F(b) = F(e)\} \tag{1}$$

Given a d-dimensional MTS $\mathfrak{E}\colon \mathbb{N} \rightharpoonup \mathbb{R}^d$ where its last dimension d reflects the fluctuation of the class label in time, we refer to its *size* $|\mathfrak{E}|$ as the number of time (points) counting from 1, thus $\text{dom}(\mathfrak{E}) = \{1, \ldots, |\mathfrak{E}|\}$. We also denote $\mathfrak{E}(t)(\psi)$ as the value of the MTS $\mathfrak{E}$ at time point $t \in \text{dom}(\mathfrak{E})$ in dimension $1 \leq \psi \leq d$. Given a time interval $[b,e]$ and a MTS $\mathfrak{E}$ of at least size e, we denote $\mathfrak{E}[b,e]$ as its *projection* considering a subsequence of the events occurring within this interval:

$$\mathfrak{E}[b,e](t) = \mathfrak{E}(t+b-1) \textbf{ if } t+b-1 \in \text{dom}(\mathfrak{E}) \textbf{ and } 1 \leq t \leq e-b+1 \tag{2}$$

Given a dimension $1 \leq \psi \leq d$, we denote $\mathfrak{E}_\psi$ as the TS providing $\mathfrak{E}$ values for dimension ψ: $\mathfrak{E}_\psi(t) = \mathfrak{E}(t)(\psi)$ for all $t \in \text{dom}(\mathfrak{E})$.

Given a MTS $\mathfrak{E}$ within a training dataset D and a set of maximal and non-overlapping temporal subsequences $[b,e] \in \mathbb{I}_{\mathfrak{E}_d}([1, |\mathfrak{E}|])$ targeting class fluctuations in dimension d, we want to learn a function h minimising the classification error $\sum_{b \leq t \leq e} |h(\mathfrak{E}[b,e]) - \mathfrak{E}(t)(d)|$ for each maximal interval $[b,e]$ and from all $\mathfrak{E}$ in D. This is a more general formulation than the one posed by current literature, considering time series where $\mathbb{I}_{\mathfrak{E}_d}([1, |\mathfrak{E}|]) = \{[1, |\mathfrak{E}|]\}$, where one classification label is associated to each timestamp.

3.1. Polyadic Logs

A *polyadic log* $\mathfrak{S}$ is a pair $\langle \mathbb{G}, \mathcal{L} \rangle$, where *(i)* $\mathcal{L}$ is a collection of distinct *polyadic traces* $\langle \sigma^1, \ldots, \sigma^n \rangle$ referring to the auditing of a specific environment $\mathfrak{E}$ of interest (e.g., a MTS); each trace σ^i of length o is a list of temporally ordered *polyadic events* $\langle \sigma^i_1, \ldots, \sigma^i_o \rangle$, where each of these σ^i_j of length m is defined as a pair of a set of *durative constituents* and a class $\chi \in \mathbb{R}$ $\sigma^i_j = \langle \langle \varsigma^i_{j,1}, \ldots, \varsigma^i_{j,m} \rangle, \chi \rangle$, where all constituents start at the same time j but possibly with different activity labels and duration span; in particular, each durative constituent $\varsigma^i_{j,k}$ is expressed as a triplet $\langle \mathsf{a}, \tilde{p}, \mathfrak{s} \rangle$ where a is the activity label, $\tilde{p}$ is the payload collecting the raw data being associated to the durative constituent, and $\mathfrak{s}$ denotes its temporal span s.t. $\mathfrak{s} \leq n-j+1$. In this, we denote with λ, ϖ, and δ the function extracting the activity label ($\lambda(\varsigma^i_{j,k}) = \mathsf{a}$), its payload ($\varpi(\varsigma^i_{j,k}) = \tilde{p}$), and its span ($\delta(\varsigma^i_{j,k}) = \mathfrak{s}$). We denote $\kappa(\varsigma^i_{j,k})$ as the information concatenation function merging all of this information in the following finite function:

$$\varpi(\varsigma^i_{1,k}) \oplus [__\text{label} \mapsto \lambda(\varsigma^i_{1,k}), __\text{span} \mapsto \delta(\varsigma^i_{1,k})] \tag{3}$$

Last, *(ii)* $\mathbb{G}$ is a collection of *taxonomies* [35,12] represented as Direct (Acyclic) Graphs $G_\psi = \langle N_{G_\psi}, R_{G_\psi}, \ell_{G_\psi} \rangle$ rooted in ℓ_{G_ψ} determining the relationship between activity labels within the log $\mathfrak{S}$; the entities are represented as a set of nodes N_{G_ψ}, and R_{G_ψ} is the set of the is-a relationships.

3.2. Poly-DECLARE

Polyadic traces trivially violate the temporal non-simultaneity assumption from non-polyadic logs [10] for which each non-polyadic event cannot be associated with multiple activity labels, while now one single polyadic event might contain constituents with different activity labels. As a result, the former algorithms did not consider concurrent violation conditions: e.g., for Precedence(A,**true**,C, **true**), we now also prescribe that a C-labelled constituent shall also never co-occur with an A-labelled constituent (cfr. the occurrence of $\leq$, Algorithm 4). Achieving all these desiderata requires completely revising the previous algorithms [11,14] to be adapted to the novel log assumptions (see Algorithm 5).

Given that each event might contain multiple constituents, we extend the traditional DECLARE semantics by checking whether some or all the constituents activated within a specific event satisfy the given declarative clause. We retain the former as the default interpretation of the declared clauses, thus retaining the same template name while, for indicating the others, we explicitly append a All prefix to the templates c_l from Fig. 1b as $\mathsf{All}\|c_l$.

4. EMeriTAte+DF

This section introduces the algorithmic extensions moving from EMeriTAte to its DATAFUL (DF) *variant (EMeriTAte+DF). Unlike our previous contribution, we characterize our algorithms as Explainable and Verified AI. While the A Priori phase adds more contextual information on the data series, the Specification Mining and the White-Box model learning for learning the explainable specifications from the data is carried out in the Ad Hoc phase. The evaluation section provides the Post Hoc evaluation of the trained models under different datasets using the default metrics.*

4.1. A Priori Explainability

The a priori explainability phase was improved from [24] as Algorithm 1 by adequately indexing MTS to support faster DT mining algorithms while also simplifying over the redundant patterns from Fig. 1a. Differently from the previous, we extend each constituent having DT activity labels, once represented as a dataless durative constituent, with catch22 features as data payloads for the mined durative constituents.

Algorithm 1 A Priori Explainability

1: **procedure** TSLOADINDEX($\mathfrak{E}; \varepsilon$)
2: **for all** $[B, E] \in \mathbb{I}_{\mathfrak{E}_d}([1, |\mathfrak{E}|])$ **do**
3: $\tau \leftarrow \mathfrak{E}[B, E]$; $\chi \leftarrow \tau(1)(d)$
4: **for** ψ **from** 1 to $d - 1$ **do**
5: **for all** $x \in \{\mathtt{i}, \mathtt{a}, \mathtt{s}, \mathtt{v}\}$ **do**
6: $\mathcal{L}_x^{\psi,\tau} \leftarrow \langle \sigma_1^{\psi,\tau}, \ldots, \sigma_{E-B+1}^{\psi,\tau} \rangle$ ▷ *Loading*
7: **for all** $1 \leq t \leq E - B + 1$ **do**
8: $\varsigma_{t,x}^{\psi,\tau} \leftarrow \langle P_x^\tau(t), [\psi \| x \mapsto \mathit{val}_x^\tau(t)], 1 \rangle$
9: $\sigma_t^{\psi,\tau} \leftarrow \langle \langle \varsigma_{t,x}^{\psi,\tau} \rangle, \chi \rangle$
10: $\mathcal{G}_{x,\textbf{true}}^{\psi,\tau} \leftarrow \{ [b, e] \in \mathbb{I}_{P_x^\tau}([B, E]) \mid \forall b \leq t \leq e.\, P_x^\tau(t) \}$ ▷ *Indexing,* P_x^τ *via* $\mathcal{L}_x^{\psi,\tau}$
11: $\mathcal{G}_{x,\textbf{false}}^{\psi,\tau} \leftarrow \{ [b, e] \in \mathbb{I}_{P_x^\tau}([B, E]) \mid \forall b \leq t \leq e.\, \neg P_x^\tau(t) \}$

Loading and Indexing. The first step is to *load* the MTS discretized as a preliminary pointwise log of events expressing the verification or not of a specific *numerical condition* x expressing different types of numeric variations (Table 2) such as increase (i), absence (a), stationarity (s), and variability (v), and by creating pointwise events satisfying or not some associated condition $P_x^\tau(t)$. This will then be used to mine the DT patterns.

Table 2. Point-wise predicates P_x^τ and values val_x^τ for discretizing time series τ over dimension i and numerical conditions x

x		$\mathit{val}_x^\tau(t)$	$P_x^\tau(t)$
i	$\tau(t+1)(i) - \tau(t)(i)$		$\mathit{val}_\mathtt{i}^\tau(t) > 0$
a	$\lvert\tau(t)(i)\rvert$		$\mathit{val}_\mathtt{i}^\tau(t) \leq \varepsilon$
s	$\lvert \mathit{val}_\mathtt{i}(t) \rvert$		$\mathit{val}_\mathtt{s}^\tau(t) \leq \varepsilon$
v	$\begin{cases} 0 \\ +\infty \\ \frac{\mathit{val}_\mathtt{i}^\tau(t)}{\tau(t)(d)} \end{cases}$	$\begin{cases} P_\mathtt{s}^\tau(t) \\ P_\mathtt{a}^\tau(t) \\ \text{oth.} \end{cases}$	$P_\mathtt{s}^\tau(t)$

We perform a linear scan of each multivariate time series $\mathfrak{E}$ so to identify the maximal intervals $[B, E] \in \mathbb{I}_{\mathfrak{E}_d}([1, |\mathfrak{E}|])$ associated with the same class reported in dimension d (Line 2), thus considering a projection τ for each of these $[B, E]$. For each dimension of interest (Line 4), we discretize each dimension and timestamp as in our previous paper [16] by considering numerical variations (Line 5). All such constituents are then stored in events (Line 9) contained in $\mathcal{L}_x^{\psi,\tau}$ for each dimension ψ, segment $[B, E]$ as τ, and numerical variation x (Line 6).

Next, we *index* the previously-discretized logs according to the satisfaction (Line 10) or violation (Line 11) of the predicate P_x^τ. We ensure that this information is computed only once by identifying regions of a specific time series' dimension where numerical

function $\mathrm{DL}(\ell, \nu; \mathfrak{E}^x_\psi)$ ▷ *Generating DT activity labels*
  if $\ell = \mathtt{S}$ **then return** (ν **?** IncreaseRapidly **:** DecreaseRapidly) $\|(\mathfrak{E}^x_\psi)$
  if $\ell = \mathtt{HV43}$ **then return** (ν **?** HighVolatilityIII **:** HighVolatilityIV) $\|(\mathfrak{E}^x_\psi)$
  if $\ell = \mathtt{1H}$ **then return** (ν **?** IncreaseSlowlyI **:** DecreaseSlowlyIV) $\|(\mathfrak{E}^x_\psi)$
  if $\ell = \mathtt{2H}$ **then return** (ν **?** IncreaseSlowlyII **:** DecreaseSlowlyIII) $\|(\mathfrak{E}^x_\psi)$
  if $\ell = \mathtt{E1}$ **then return** (ν **?** IncreaseSlowlyIII **:** DecreaseSlowlyII) $\|(\mathfrak{E}^x_\psi)$
  if $\ell = \mathtt{E2}$ **then return** (ν **?** IncreaseSlowlyIV **:** DecreaseSlowlyI) $\|(\mathfrak{E}^x_\psi)$

function C22SHORT($\mathfrak{E}_\psi, b, e$) ▷ *Combining the TS patterns with the time ones*
  return $\mathrm{C22}(\mathfrak{E}_\psi[\![b, e]\!]) \oplus \mathrm{C22}$([b,e])

procedure DTMINESTEP($\tau, \psi, x, \nu, [b, e]; \mathfrak{E}$)
  $n \leftarrow e - b + 1$
  yield $\varsigma^{\mathfrak{E}}_{b,\mathtt{fresh}(b)} \leftarrow \langle \mathrm{DL}(\mathtt{S}, \nu; \mathfrak{E}^x_\psi), \mathrm{C22SHORT}(\mathfrak{E}_\psi, b, e), n\rangle$
  for all $1 \le t \le n$ **do** ▷ *Determining the span $\mathfrak{s}$ from t*
    for all $b \le \beta \le e - t + 1$ **do**
      $\xi \leftarrow (e+1) \in \mathcal{G}^{\psi,\tau}_{x,\mathbf{not}\,\nu}$ **and** $\beta = e$; $\eta \leftarrow \beta + t - 1$
      if $\beta = b$ **then**
        if $(b-1) \in \mathcal{G}^{\psi,\tau}_{x,\mathbf{not}\,\nu}$ **then**
          yield $\varsigma^{\mathfrak{E}}_{\beta-1,\mathtt{fresh}(\beta-1)} \leftarrow \langle \mathrm{DL}(\mathtt{1H}, \nu; \mathfrak{E}^x_\psi), \mathrm{C22SHORT}(\mathfrak{E}_\psi, \beta - 1, \eta), t+1\rangle$
          if ξ **then**
            yield $\varsigma^{\mathfrak{E}}_{\beta-1,\mathtt{fresh}(\beta-1)} \leftarrow \langle \mathrm{DL}(\mathtt{HV43}, \nu; \mathfrak{E}^x_\psi), \mathrm{C22SHORT}(\mathfrak{E}_\psi, \beta - 1, \eta + 1), t+2\rangle$
          if $(\beta - 2) \in \mathcal{G}^{\psi,\tau}_{x,\nu}$ **then**
            yield $\varsigma^{\mathfrak{E}}_{\beta-2,\mathtt{fresh}(\beta-2)} \leftarrow \langle \mathrm{DL}(\mathtt{2H}, \nu; \mathfrak{E}^x_\psi), \mathrm{C22SHORT}(\mathfrak{E}_\psi, \beta - 2, \eta), t+2\rangle$
      if $\beta + t - 1 = e$ **and** ξ **then**
        found $\leftarrow$ **false**
        if $[e+1, e+2] \in \mathcal{G}^{\psi,\tau}_{x,\mathbf{not}\,\nu}$ **and** $e + 2 \in \mathcal{G}^{\psi,\tau}_{x,\nu}$ **then**
          found $\leftarrow$ **true**
          yield $\varsigma^{\mathfrak{E}}_{\beta,\mathtt{fresh}(\beta)} \leftarrow \langle \mathrm{DL}(\mathtt{E1}, \nu; \mathfrak{E}^x_\psi), \mathrm{C22SHORT}(\mathfrak{E}_\psi, \beta, \eta + 2), t+2\rangle$
        if not *found* **then**
          yield $\varsigma^{\mathfrak{E}}_{\beta,\mathtt{fresh}(\beta)} \leftarrow \langle \mathrm{DL}(\mathtt{E2}, \nu; \mathfrak{E}^x_\psi), \mathrm{C22SHORT}(\mathfrak{E}_\psi, \beta, \eta + 1), t+1\rangle$

procedure DTMINE($\mathfrak{E}$)
  $S^{\mathfrak{E}} \leftarrow \emptyset$; $\sigma^{\mathfrak{E}} \leftarrow \langle\rangle$; INITFRESH()
  for all $[B, E] \in \mathbb{I}_{\mathfrak{E}_d}([1, |\mathfrak{E}|])$ **do**
    $\tau \leftarrow \mathfrak{E}[B, E]$; $\chi \leftarrow \tau(1)(d)$
    for all $1 \le t \le E - B + 1$ **do**
      $\varsigma^{\mathfrak{E}}_{B+t-1,1} \leftarrow \langle$__raw_data, $[\mathtt{dim_}\psi \mapsto \tau(t)(\psi)]_{1 \le \psi \le d-1}, 1\rangle$
      $S^{\mathfrak{E}}_{B+t-1} \leftarrow [\varsigma^{\mathfrak{E}}_{B+t-1,1}]$; $\sigma^{\mathfrak{E}}_{B+t-1} \leftarrow \langle S^{\mathfrak{E}}_{B+t-1}, \chi\rangle$; $\sigma^{\mathfrak{E}} \leftarrow \sigma^{\mathfrak{E}} \cup [\sigma^{\mathfrak{E}}_{B+t-1}]$
    for ψ **from** 1 to $d - 1$ **do**
      for all $x \in \{\mathtt{i}, \mathtt{a}, \mathtt{s}, \mathtt{v}\}$ **do**
        for all $[b, e] \in \mathcal{G}^{\psi,\tau}_{x,\nu}$ **do**
          $\varpi(\varsigma^{\mathfrak{E}}_{B-b+1,1}) \leftarrow \varpi(\varsigma^{\mathfrak{E}}_{B-b+1,1}) \oplus \varpi(\varsigma^{\psi,\tau}_{B-b+1,x})$
          $S^{\mathfrak{E}} \leftarrow S^{\mathfrak{E}} \cup \mathrm{DTMINESTEP}(\tau, \psi, x, \mathbf{true}, [b, e]; \mathfrak{E})$
          $S^{\mathfrak{E}} \leftarrow S^{\mathfrak{E}} \cup \mathrm{DTMINESTEP}(\tau, \psi, x, \mathbf{false}, [b, e]; \mathfrak{E})$
    for all $\varsigma^{\mathfrak{E}}_{j,k} \in S^{\mathfrak{E}}$ **do** $S^{\mathfrak{E}}_j \leftarrow S^{\mathfrak{E}}_j \cup [\varsigma^{\mathfrak{E}}_{j,k}]$
  return $\sigma^{\mathfrak{E}}$

variation conditions continue to hold or not. As the intervals in such indices are never overlapping, we can store the intervals in order of increasing beginning time and query those in $O(\log |\mathcal{G}^{\psi,\tau}_{x,_}|)$ to check whether the time series at dimension i satisfies the numerical variation condition x at a specific time t. Hence, we denote and define such query as $t \in \mathcal{G}^{\psi,\tau}_{x,_} \Leftrightarrow \exists [b,e] \in \mathcal{G}^{\psi,\tau}_{x,_}.\ t \in [b,e]$: this notion is used in the forthcoming subsection.

Lemma 1 (Indexing and Loading Time Complexity). *Given a collection of N MTS of d dimensions with maximum length t, the time complexity for* TSLoadIndex *is in* $O(Ndt)$.

Proof. Given that all insertions on indices can be built while linearly scanning the data, and assuming that each $P^{\tau}_{x}(x)$ can be tested in constant time, the time complexity of scanning each maximal time interval $[B,E]$ and each value in it boils down to the maximal time series length t. Given that the number of point-wise predicates is constant, we can conclude that the linear scan is provided for each dimension and MTS within the collection, thus leading to $O(Ndt)$. □

Indexed DT Mining. This phase enables the representation of a time series as a polyadic trace, where the first constituent for each j-th event reflects the values associated with each dimension at time j, i.e. $\mathfrak{E}(j)$ (Lines 49-50), while the remainder is derived from the DT-mined pointwise constituents from the previous phase (Lines 55-56). The composition of the $\mathcal{G}$-indexed intervals generate DT patterns distinguishable from a constituent activity label (Line 12), where the name of the DT pattern reports the dimension ψ and the specific numerical variation x. The algorithm proceeds by scanning each maximal interval $[B,E]$ for a time series $\mathfrak{E}$ over each dimension ψ, numerical variation x, and $\mathcal{G}$-indexed satisfiability interval $[b,e]$ (Lines 55-56). The segmentation of the time series through the $\mathcal{G}$ indices allows to promptly derive that any subsequent or preceding interval within the same dimension will violate some x constraints if the current interval satisfies them, and vice versa. Differently from our previous paper, for each mined constituent referring to a specific dimension i and interval $[\beta,\eta] \subseteq [b,e]$, we generate a payload containing the univariate catch22 [30] features (C22) capturing the dynamic properties of the ψ-th dimension in $\mathfrak{E}_{\psi}[\beta,\eta]$, which are paired with the time fluctuations (Line 20). By doing so, we generalise over the CIF classifier [32], which focuses on distinguishing time series from the behaviours occurring within a specific fixed-size sliding window while potentially considering the features for all possible sliding windows captured as DT patterns. The computation of the Catch22-payload is parallelized due to the high computational nature of the metrics used to describe the numerical fluctuations of the data. The main mining algorithm considers a subset of all the DTs patterns described in Section 2.2, where some high volatility patterns are discarded so to favour the recognition of shorter volatility patterns rather than preferring joining prolonged variations. Overall, this helps to reduce the number of patterns to be returned through the combination of subsequent patterns while reducing the time to mine the clauses without considerably undermining the predictive power of the classifier. At the end of the process, each time series $\mathfrak{E}$ is discretized into its constituents and collected in a single polyadic trace $\sigma^{\mathfrak{E}}$.

Lemma 2. *The worst-case time complexity for the indexed DT mining from* DTMine *is superpolynomial over the maximum length of the segmented interval $\mathfrak{s}$ and linear over the number and size of the MTS.*

Proof. Note that the worst-case scenario time complexity does not happen for MTS exhibiting stable values, as this would lead to just one maximal interval $[B, E]$ per class. In fact, these situations have ξ and the condition at Line 28 as false, as there will be no other neighbouring intervals to consider. This will only lead to returning the durative constituent at Line 23, where still catch22 statistics are computed. As the time complexity of computing all the catch22 statistics depends on the length of the interval of interest, we denote this as $C_\jmath$.

Thus, the worst time complexity happens when we have at least two intervals, leading not only to the generation of the aforementioned durative constituent, but potentially to other patterns occurring within the nested iteration. Given that within the nested loop leading to $O(\jmath^2)$ iterations we test queries having a time in $O(\log \jmath)$, we obtain an overall time complexity of $O(\jmath^2 \log \jmath C_\jmath \frac{t}{\jmath} N)$. □

Serialization At serialization time, we avoid writing any potential redundant constituent having the same activation label but referring to a shorter time interval to the other sibling constituents, thus capturing the behaviour associated with the longest span. We generate a collection of taxonomies for reconducing each constituent label to the original MTS dimension the pattern was referring to: we generate a taxonomy $G_\psi = \langle N_{G_\psi}, R_{G_\psi}, \ell_{G_\psi} \rangle$ for each dimension ψ, where N_{G_ψ} contains all the DTs associated with $\mathfrak{E}^x_\psi$ thus including $\mathfrak{E}^x_\psi$ acting as the root ℓ_{G_ψ}; R_{G_ψ} connects only the root with the associated DT labels. Given the impossibility of serializing such data in XES [2] due to their strict assumptions on the log representation, we serialize the resulting information on a customary representation in `json`.

4.2. Ad Hoc Explainability

We load the single log composed of many traces with classes changing in time as many segmented logs $\mathfrak{S}_y$ as the total number of classes $y \in \mathcal{Y}$ distinctly occurring by segmenting the traces in maximal intervals where the same class holds. Then, differently from our previous solution, where each segmented log was mined separately via Bolt2 [11] to then derive the trace embedding in a later phase, we contemporarily load and mine all the segmented logs containing maximal sequences referring to the same classes. We then proceed to the DECLARE clause refinement for the ones shared across the logs as a result of the mining phase: as generic dataless DECLARE clauses with DT activity labels might be not sufficient to temporally characterize the classes, we try to specialize each clause according to the specificity of each log/class. The rationale is the following: if the same dataless clause is frequently mined from two segmented logs, at least one trace exists from each of these where the dataless clause is both activated and holds. Then, the only way to differentiate the clause across the two segmented logs is to change the satisfiability outcome of the trace by refining the original dataless clause with dataful predicates for either the activation or the target condition. Suppose the payload referring to the constituents activated by the clause can be differentiated. In that case, we can refine the original dataless clause as two dataful ones, each with an activation condition (p and p') satisfying only the activations from one segmented log. This leads to one dataful clause being then activated over the traces of one segmented log, but being vacuously satisfied by the traces of the other. If that will not suffice, then the only further possible way to differentiate is to

generate target predicates similar to those above: in this other scenario, the missed satisfaction of the target condition will lead to a violation of the clause, thus further refining the trace behaviour in terms of clause (un)satisfaction.

This significantly differs from other deviance mining approaches [27], where activation conditions are mainly clustered not necessarily according to the target classes of interest [14], thus not achieving a truthful clause separation for differentiating them across the distinct logs as discussed above. By exploiting a decision tree as a classifier, we can ensure that the data predicates used to differentiate the classes will summarize the data properties as unary dataful predicates, which might reconstructed by traversing each branch of the learned decision tree by interpreting it as a single propositional formula π.

Trace Loading and Segmentation. Similarly to our previous paper, for each trace $\sigma^{\mathfrak{E}_A}$ in the `json`-serialized log $\mathfrak{S}$ describing a distinct MTS $\mathfrak{E}$, we scan all polyadic events, and we group them by maximal contiguous sequences associated to the same class y. Each class y will then correspond to one single final log $\mathfrak{S}_y$, where each maximal contiguous sequence of polyadic events will correspond to one single new polyadic trace in $\mathfrak{S}_y$. As this transformation also preserves the __raw_data information associated with each polyadic event, we later collect the raw data associated with the polyadic traces for later on running atemporal event classification tasks by just correlating the class within the __raw_data payload with the other available values. As polyadic events now supports multiple constituents, this required us to update our KnoBAB solution to both extend the events' tabular representation (*activity table*), as well as extending the secondary index structures for accessing multiple immediately preceeding or subsequent constituents within the same trace.

Polyadic Deviant Model Learning. Algorithm 2 showcases the procedure for extracting an explainable specification capturing MTS classes' temporal characterization by setting χ as the `class` associated with the i-th trace from $\mathfrak{S}_\chi$ (Line 7). This is achieved by considering DECLARE clauses, be they data or dataless, as features for describing class-segmented traces. By interpreting the numerical values as a violation, satisfaction, or vacuous satisfaction of a specific clause, we then use a white box classifier such as Decision Tree to extract a propositional representation for the clauses' satisfaction.

Similarly to our previous implementation [11,16], we consider all the most frequent itemsets from FREQUENTITEMSETS for at most two activity labels and support of at least θ (Line 8). Unary DECLARE clauses such as Init, End, and Exists are mined from the frequently-occurring unary patterns in Φ_χ as per Bolt2 (Line 9). We further proceed on the refinement for each of the clauses into their dataful counterpart if and only if the payload-based refinement allows us to substantially differ the traces' behaviour according to the constituents' payloads and, otherwise, we backtrack to the original dataless variants (Line 21). Similar considerations can be carried out for the remaining binary clauses in $\mathsf{FreqPairs}_\chi$: given all the possible binary frequent patterns occurring across all logs (Line 12), we perform no refinement if such pair is frequent only in one segmented log (Line 14) and, otherwise, we proceed by refining the clauses when possible (Line 17). Last, we train the white-box classifier over the extracted embedding for each class-segmented trace stored as a DF row, from which we derive a propositional [13] and declare-based [14] characterization for each MTS class (Line 22).

Algorithm 2 Ad Hoc Explainability

1: **function** POLYADICDATAAWARE($\mathfrak{S}_1, \ldots, \mathfrak{S}_n; \theta$)
2: $\Sigma \leftarrow \bigcup_{\mathfrak{S}_\chi = \langle \mathbb{G}_\chi, \mathcal{L}_\chi \rangle, \langle N_{G_\psi}, R_{G_\psi}, \ell_{G_\psi} \rangle \in \mathbb{G}_\chi} N_{G_\psi}$ ▷ *Determining the alphabet*
3: $\Phi \leftarrow \emptyset$ ▷ *Clauses not requiring dataful refinement*
4: **global** DF ▷ *Double-entry table initialization*
5: **for all** $1 \leq \chi \leq n$ **do**
6: **for all** $\sigma^i \in \mathcal{L}, \mathfrak{S}_\chi = \langle \mathbb{G}_\chi, \mathcal{L}_\chi \rangle$ **do**
7: DF$[(\chi, i), \texttt{class}] \leftarrow \chi$ ▷ *Segmented trace/class assoc.*
8: $S_\chi \leftarrow$ FREQUENTITEMSETS($\mathfrak{S}_\chi, \theta$) ▷ *Maximal length of 2*
9: $\mathsf{FreqPairs}_\chi, \Phi_\chi \leftarrow$ GENERATEUNARYCLAUSES(S_χ) ▷ *Unary clauses* [11]
10: $\Phi \leftarrow$ UNARYREFINE($\Phi_1, \ldots, \Phi_n, \Phi; \mathfrak{S}_1, \ldots, \mathfrak{S}_n, \Sigma$) ▷ Algorithm 3
11: FP $\leftarrow \bigcup_{1 \leq \chi \leq n} \text{FreqPairs}_\chi$
12: **for all** $\langle \mathsf{A}, \mathsf{B} \rangle \in$ FP **do**
13: $\Phi_{\mathsf{AB}} \leftarrow \{c_l(\mathsf{A}, \mathbf{true}, \mathsf{B}, \mathbf{true}) | c_l \in \textit{BinaryTemplates}\}$ ▷ *Probing all clauses*
14: **if** $\exists! j. \langle \mathsf{A}, \mathsf{B} \rangle \in \text{FreqPairs}_j$ **then**
15: $\Phi \leftarrow \Phi \cup \Phi_{\mathsf{AB}}$
16: **else**
17: $\Phi \leftarrow \Phi \cup$ BINARYREFINE($\mathsf{A}, \mathsf{B}, \mathfrak{S}_1, \ldots, \mathfrak{S}_n$; **true**) ▷ Algorithm 4
18: **for all** $1 \leq \chi \leq n$ **do** ▷ *Dataless [16]*
19: **for all** $\sigma^i \in \mathcal{L}, \mathfrak{S}_\chi = \langle \mathbb{G}_\chi, \mathcal{L}_\chi \rangle$ **do**
20: **for all** $\varphi \in \Phi$ **do**
21: DF$[(\chi, i), \varphi] \leftarrow \mathbb{B}(\sigma^i \models \varphi)$
22: **return** DECISIONTREE($\Big\{ ([k \mapsto \text{DF}[r, k]]_{k \in \texttt{cols(DF)} \setminus \{\texttt{class}\}}, \text{DF}[r, \texttt{class}])$
23: $\Big| \, r \in \texttt{rows}(\text{DF}) \Big\}$)

Unary Refinement. Algorithm 3 outlines the dataful refinement of the unary clauses of (All)Init, (All)End, and (All)Exists, which is attempted if and only if we can mine the same clause from at least two class-segmented logs (Lines 17, 23, and 29). When this occurs, we collect in D all the payloads associated with the constituents occurring at the beginning (Line 20), end (Line 26) or from any constituent (Line 32) according to the clause to be refined. Lines 9 and 10 achieve the polyadic extension for DECLARE by considering the $\mathsf{All}\|c_l$ clause variaties: this is achieved by not considering one payload at a time, but by considering the payload of all the constituents of interest cotemporarily, as stored in $\mathcal{D}$. We also consider all the predicates given within the decision tree and representing one class of interest and put them in disjunction, thus considering conditions generally identifying classes (Line 11). No refinement is provided if the decision tree cannot adequately separate the activation payloads according to the associated class to generate refined dataful activation predicates (Line 6).

As the explanation generated from the decision tree's *ex post* phase provides a propositional formula of declarative clauses, we discard the refinement for Absence, as this can still appear as the negation of any Exist clause [10].

Binary Refinement. Algorithm 4 describes refining some Polyadic DECLARE binary clauses. Given that the clauses expressing (Excl)Choice, CoExistence, and RespExis-

Algorithm 3 Ad Hoc Explainability: unary clause dataful refinement

function REFINEATTEMPT($D, \mathcal{D}, c_l, n; \vartheta = .7$)
  global DF
  $D_{\text{train}}, D_{\text{test}} \leftarrow$SPLIT($D, \vartheta$)
  $\mathcal{M} \leftarrow$ DECISIONTREE(D_{train})
  $\mathcal{Y} \leftarrow [\chi \mapsto \bigwedge_{\pi \in \mathcal{M}, \pi = H \Rightarrow \chi} H]_{1 \leq \chi \leq n}$
  if ACCURACY($\mathcal{M}, D_{\text{test}}$)$\leq 50\%$ **then return false**
  for all $(\chi, \sigma^i) \in \text{dom}(\mathcal{D})$ **do**
    for all $\pi \in \mathcal{M}$ **do**
      DF$[(\chi, i), \mathsf{All} \| c_l(*, \pi)] \leftarrow \mathbb{B}(|\mathcal{D}(\chi, \sigma^i)| = \sum_{p \in \mathcal{D}(\chi, \sigma^i)} \mathbf{1}_{\pi(p)})$
      DF$[(\chi, i), c_l(*, \pi)] \leftarrow \mathbb{B}(0 < \sum_{p \in \mathcal{D}(\chi, \sigma^i)} \mathbf{1}_{\pi(p)})$
    for all $\chi \in \text{dom}(\mathcal{Y})$ **do**
      $\pi \leftarrow \mathcal{Y}(\chi)$
      DF$[(\chi, i), \mathsf{All} \| c_l(*, \pi)] \leftarrow \mathbb{B}(|\mathcal{D}(\chi, \sigma^i)| = \sum_{p \in \mathcal{D}(\chi, \sigma^i)} \mathbf{1}_{\pi(p)})$
      DF$[(\chi, i), c_l(*, \pi)] \leftarrow \mathbb{B}(0 < \sum_{p \in \mathcal{D}(\chi, \sigma^i)} \mathbf{1}_{\pi(p)})$
  return true

function UNARYREFINE($\Phi_1, \ldots, \Phi_n, \Phi; \mathfrak{S}_1, \ldots, \mathfrak{S}_n, \Sigma$)
  refineInit $\leftarrow \exists \mathsf{a} \in \Sigma. \exists i, j \in \mathbb{N}. \mathsf{Init}(\mathsf{a}, \textbf{true}) \in \Phi_i \cup \Phi_j$
  if refineInit **then** ▷ Init *refinement*
    $\mathcal{D} \leftarrow \left[(\chi, \sigma^i) \mapsto S\right]_{1 \leq \chi \leq n, \mathfrak{S}_\chi = \langle \mathbb{G}, \mathcal{L} \rangle, \sigma^i \in \mathcal{L}, \sigma^i_1 = \langle S, \chi \rangle}$
    $D \leftarrow \left\{ (\kappa(\varsigma^i_{1,k}), \chi) \;\middle|\; \varsigma^i_{1,k} \in \sigma^i_1, \sigma^i \in \mathcal{L}, \mathfrak{S}_\chi = \langle \mathbb{G}, \mathcal{L} \rangle, 1 \leq \chi \leq n \right\}$
    refineInit $\leftarrow$ REFINEATTEMPT($D, \mathcal{D}, \mathsf{Init}, n$)
  if not refineInit **then** $\Phi \leftarrow \Phi \cup \{\mathsf{Init}(\mathsf{b}, \textbf{true}) | \mathsf{Init}(\mathsf{b}, \textbf{true}) \in \bigcup_{1 \leq \chi \leq n} \Phi_\chi\}$
  refineEnd $\leftarrow \exists \mathsf{a} \in \Sigma. \exists i, j \in \mathbb{N}. \mathsf{End}(\mathsf{a}, \textbf{true}) \in \Phi_i \cup \Phi_j$
  if refineEnd **then** ▷ End *refinement*
    $\mathcal{D} \leftarrow \left[(\chi, \sigma^i) \mapsto S\right]_{1 \leq \chi \leq n, \mathfrak{S}_\chi = \langle \mathbb{G}, \mathcal{L} \rangle, \sigma^i \in \mathcal{L}, \sigma^i_{|\sigma^i|} = \langle S, \chi \rangle}$
    $D \leftarrow \{(\kappa(\varsigma^i_{|\sigma^i|,k}), \chi) \mid \varsigma^i_{|\sigma^i|,k} \in \sigma^i_{|\sigma^i|}, \sigma^i \in \mathcal{L}, \mathfrak{S}_\chi = \langle \mathbb{G}, \mathcal{L} \rangle, 1 \leq \chi \leq n\}$
    refineEnd $\leftarrow$ REFINEATTEMPT($D, \mathcal{D}, \mathsf{End}, n$)
  if not refineEnd **then** $\Phi \leftarrow \Phi \cup \{\mathsf{End}(\mathsf{b}, \textbf{true}) | \mathsf{End}(\mathsf{b}, \textbf{true}) \in \bigcup_{1 \leq \chi \leq n} \Phi_\chi\}$
  refineEx $\leftarrow \exists \mathsf{a} \in \Sigma. \exists i, j \in \mathbb{N}. \mathsf{Exists}(\mathsf{a}, \textbf{true}) \in \Phi_i \cup \Phi_j$
  if refineEx **then** ▷ Exists *refinement*
    $\mathcal{D} \leftarrow \left[(\chi, \sigma^i) \mapsto \left\{ \varpi(\varsigma^i_{j,_}) \;\middle|\; \varsigma^i_{j,_} \in \sigma^i_j, \sigma^i \in \mathcal{L} \right\}\right]_{1 \leq \chi \leq n, \mathfrak{S}_\chi = \langle \mathbb{G}, \mathcal{L} \rangle}$
    $D \leftarrow \left\{ (\kappa(\varsigma^i_{j,k}), \chi) \;\middle|\; \varsigma^i_{j,k} \in \sigma^i_j, \sigma^i \in \mathcal{L}, \mathfrak{S}_\chi = \langle \mathbb{G}, \mathcal{L} \rangle, 1 \leq \chi \leq n \right\}$
    refineEx $\leftarrow$ REFINEATTEMPT($D, \mathcal{D}, \mathsf{Exists}, n$)
  if not refineEx **then** $\Phi \leftarrow \Phi \cup \{\mathsf{Exists}(\mathsf{b}, \textbf{true}) | \mathsf{Exists}(\mathsf{b}, \textbf{true}) \in \bigcup_{1 \leq \chi \leq n} \Phi_\chi\}$
  return Φ

Algorithm 4 Ad Hoc Explainability: binary clause dataful refinement

procedure FILLINDATAFRAME(S_i^χ, φ)
  ▷ S_i^χ: *set of* Vac*(uity),* Viol*(ations), or* Sat*(isfiability) conditions*
  global DF
  if $S_i^\chi = \emptyset$ **or** $S_i^\chi = \mathsf{Vac}$ **then**
    DF$[(\chi, i), \varphi] \leftarrow 0$
  else if $\mathsf{Viol} \in S_i^\chi$ **then**
    DF$[(\chi, i), \varphi] \leftarrow -1$
  else
    DF$[(\chi, i), \varphi] \leftarrow 1$

function BINARYREFINE($\mathsf{A}, \mathsf{B}, \mathfrak{S}_1, \ldots, \mathfrak{S}_n; \mathtt{poly} = \mathbf{true}$)
  template $\leftarrow$ [ChainResponse, ChainPrecedence, Precedence, Response]
  shorthands $\leftarrow$ [cr, cp, p, r]
  dictionary $\leftarrow$ ZIP(template,shorthands)
  pairs $\leftarrow \{\langle \mathsf{A}, \mathsf{B} \rangle, \langle \mathsf{B}, \mathsf{A} \rangle\}$
  for all $1 \leq \chi \leq n$ **do**
    $\mathfrak{S}_\chi = \langle \mathbb{G}_\chi, \mathcal{L}_\chi \rangle$
    poly $\leftarrow$ **if** ($\mathtt{poly}$ **and** $\exists G_\psi \in \mathbb{G}_\chi . \mathsf{A}, \mathsf{B} \in N_{G_\psi}$) **then true else false**
    CHAINS(A,B,θ,poly;$\mathcal{L}_\chi$); CHAINS(B,A,θ,poly;$\mathcal{L}_\chi$) ▷ Algorithm 5
    RESPPREC(A,B,θ,poly;$\mathcal{L}_\chi$); RESPPREC(B,A,θ,poly;$\mathcal{L}_\chi$)
  for all $c_l(\mathsf{A'}, \mathbf{true}, \mathsf{B'}, \mathbf{true})$ s.t. $\langle \mathsf{A'}, \mathsf{B'} \rangle \in$ pairs **and** $c_l \in$ template **do**
    short $\leftarrow$ dictionary$[c_l]$
    $D_{\text{act}} \leftarrow \left\{ (\kappa(\varsigma^i_{j,k}), \chi) \;\middle|\; \varsigma^i_{j,k}, \langle \varsigma^i_{j,k}, _ \rangle \in \text{act}(\chi)^{\text{short}}_{\mathsf{A'},\mathsf{B'}}, 1 \leq \chi \leq n \right\}$
    if $D_{\text{act}} = \emptyset$ **then yield** $c_l(\mathsf{A'}, \mathbf{true}, \mathsf{B'}, \mathbf{true})$
    else
      $\mathcal{M} \leftarrow$DECISIONTREE(D_{act})
      if PURITY($\mathcal{M}$)$> 50\%$ **then**
        for all $\pi \in \mathcal{M}$ **do** ▷ *Refine by activations*
          for all $1 \leq \chi \leq n,\ \mathfrak{S}_\chi = \langle _, \mathcal{L}_\chi \rangle,\ \varsigma^i_{j,k}, \langle \varsigma^i_{j,k}, _ \rangle \in \text{act}(\chi)^{\text{short}}_{\mathsf{A'},\mathsf{B'}}$ **do**
            test $\leftarrow \pi(\kappa(\varsigma^i_{j,k}))$
            if $\varsigma^i_{j,k} \in \text{viol}(\chi)^{\text{short}}_{\mathsf{A'},\mathsf{B'}}$ **then**
              $S_i^\chi \leftarrow S_i^\chi \cup \{\text{test} \,?\, \mathsf{Viol} : \mathsf{Vac}\}$
            else
              $S_i^\chi \leftarrow S_i^\chi \cup \{\text{test} \,?\, \mathsf{Sat} : \mathsf{Vac}\}$
          for all $\sigma^i \in \mathfrak{S}_\chi$ **do** FILLINDATAFRAME($S_i^\chi, c_l(\mathsf{A'}, \pi, \mathsf{B'}, \mathbf{true})$)
      else
        $D_{\text{tgt}} \leftarrow \left\{ (\kappa(\varsigma^i_{h,k'}), \chi) \;\middle|\; \langle \varsigma^i_{j,k}, \varsigma^i_{h,k'} \rangle \in \text{act}(\chi)^{\text{short}}_{\mathsf{A'},\mathsf{B'}}, 1 \leq \chi \leq n \right\}$
        $\mathcal{M} \leftarrow$DECISIONTREE(D_{tgt})
        if PURITY($\mathcal{M}$)$> 50\%$ **then**
          for all $\pi \in \mathcal{M}$ **do** ▷ *Refine by targets*
            for all $1 \leq \chi \leq n,\ \mathfrak{S}_\chi = \langle _, \mathcal{L}_\chi \rangle,\ \varsigma^i_{j,k}, \langle \varsigma^i_{j,k}, _ \rangle \in \text{act}(\chi)^{\text{short}}_{\mathsf{A'},\mathsf{B'}}$ **do**
              test $\leftarrow \pi(\kappa(\varsigma^i_{j,k}))$
              if $\varsigma^i_{j,k} \in \text{viol}(\chi)^{\text{short}}_{\mathsf{A'},\mathsf{B'}}$ **then**
                $S_i^\chi \leftarrow S_i^\chi \cup \{\text{test} \,?\, \mathsf{Viol} : \mathsf{Vac}\}$
              else
                $S_i^\chi \leftarrow S_i^\chi \cup \{\text{test} \,?\, \mathsf{Sat} : \mathsf{Vac}\}$
            for all $\sigma^i \in \mathfrak{S}_\chi$ **do** FILLINDATAFRAME($S_i^\chi, c_l(\mathsf{A'}, \mathbf{true}, \mathsf{B'}, \pi)$)
        else yield $c_l(\mathsf{A'}, \mathbf{true}, \mathsf{B'}, \mathbf{true})$ ▷ *Backtracking to the dataless clause*

tence can be easily formulated as propositional formulas composing Exists, we focus on refining the clauses that cannot be characterized through unary refinement alone: Precedence, Response, ChainPrecedence, and ChainResponse. We also discard ChainSuccession (and Succession) as those are composite clauses derivable from the former by conjunction, thus being also derivable from a propositional formula. For the time being, we do not consider All$\|c_l$ template variants, thus considering clause satisfaction if there are no violations across all activations within a trace.

After recalling that this procedure is invoked for any pair of activity labels A and B appearing as frequent in at least two segmented logs (Algorithm 2, Line 17), we first collect the evidence for the fulfilment of the aforementioned dataless clauses (narrated in the next paragraph). As this enables the collection of all the activated (Line 21) and targeted (Line 35) constituents' payloads, we can then see if is possible to extract a propositional characterization of the data through a decision tree; we consider the data summarization process through propositionalization successful if the tree achieves a suitable amount of purity (Lines 25 and 37). After extracting each path in such a tree as a binary predicate π, we consider all the activations first within a single trace σ^i of a segmented log $\mathfrak{S}_\chi$: differently from the refinement phases performed in DML [14] and in event-based mining algorithms [27], we change the clause satisfaction according to the joint fulfilment of the predicate π to generate a correct representation of the embedding: activated conditions that are no more satisfying π will always lead to a non-activation of the clause, and otherwise we retain the violation/satisfaction condition (Lines 28-32). We then consider the trace as violating the refined clause if at least one constituent leads to its violation (Line 28); we consider the clause vacuously satisfied if was never activated before or if the addition of the new π activation condition leads to de-activation of the clause testing (Line 4), and we consider as the trace being globally satisfied by the clause otherwise (Line 8). We draw similar considerations for the refinement of the target conditions. If neither of these two attempts at refining is satisfactory, we backtrack to the dataless clause (Line 45).

Algorithm 5 extends the mechanism in Bolt2 [11] in EMeriTAte for efficiently mining the aforementioned clauses by changing old trace IDs collections for clause activation (sat), vioilation (viol), or non-activation (vac), to considering the constituents generating these conditions. This reconstructs both the payloads associated with the activated and targeted conditions, and the constituent led to the violation of the clause. We denote all the possible target conditions for each activation leading to clause satisfaction as pairs of constituents associated with the activation. Bolt2 (e.g., Line 4 and Line 30) did not consider different overlapping constituents referring to the same MTS variable to avoid redundant and obvious correlations. This codebase can replicate this behaviour by setting the `poly` parameter to **false**. To recognize those, we now use activity label taxonomies and set the `poly` parameter to **true**: for constituents expressing distinct DT associated with the same MTS variable, we are interested in establishing temporal correlations between activated and targeted constituents if and only if the targeted (or activated) constituent terminates before the occurrence of the forthcoming activated (or targeted) one while always referring to the same variable. Bolt2 only considered pointwise non-polyadic events with no associated durative information, thus proving to be inadequate to support these new features.

Algorithm 5 Polyadic Mining for (Chain)Response and (Chain)Precedence.

procedure CHAINS($\mathsf{A}, \mathsf{B}, \theta, \texttt{poly}; \mathcal{L}_y$)
  for all $\sigma^i \in \mathcal{L}_y$ **do**
    for all $\varsigma^i_{j,k}$ **s.t.** $\lambda(\varsigma^i_{j,k}) = \mathsf{A}$ **do**
      $\mathsf{span} \leftarrow$ **if** $\texttt{poly}$ **then** $\pi(\varsigma^i_{j,k})$ **else** 1
      if not $\exists k.\lambda(\varsigma^i_{j+\mathsf{span},k}) = \mathsf{B}$ **then**{ act$(y)^{\mathsf{cr}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$; viol$(y)^{\mathsf{cr}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$}
      else act$(y)^{\mathsf{cr}}_{\mathsf{A},\mathsf{B}}$.add$(\langle\varsigma^i_{j,k}, \varsigma^i_{j+\mathsf{span},k}\rangle)$ **for all** k **s.t.** $\lambda(\varsigma^i_{j+\mathsf{span},k}) = \mathsf{B}$.
      if $j > 1$ **then**
        act$(y)^{\mathsf{cp}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$;
        if not $\exists h, k'.\lambda(\varsigma^i_{h,k'}) = \mathsf{B}$ **and** $h + (\texttt{poly}\,\textbf{?}\,\delta(\varsigma^i_{h,k'})\,\textbf{:}\,1) \leq j$ **then**
          viol$(y)^{\mathsf{cp}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$
        else act$(y)^{\mathsf{cr}}_{\mathsf{A},\mathsf{B}}$.add$(\langle\varsigma^i_{j,k}, \varsigma^i_{h,k'}\rangle)$ **for all** h, k' **s.t.** $\lambda(\varsigma^i_{h,k'}) = \mathsf{B}$ **and** $h + (\texttt{poly}\,\textbf{?}\,\delta(\varsigma^i_{h,k'})\,\textbf{:}\,1) \leq j$

procedure RESPPREC($\mathsf{A}, \mathsf{B}, \theta, \texttt{poly}; \mathcal{L}_y$)
  for all $\sigma^i \in \mathcal{L}_y$ **do**
    if not $\exists \varsigma^i_{h,k}.\lambda(\varsigma^i_{h,k}) = \mathsf{A}$ **then**
      return ▷ *Not reporting vacuous satisfaction explicitly*
    else if not $\exists \varsigma^i_{h,k}.\lambda(\varsigma^i_{h,k}) = \mathsf{B}$ **then**
      for all $\varsigma^i_{j,k}$ s.t. $\lambda(\varsigma^i_{j,k}) = \mathsf{A}$ **do**
        act$(y)^{\mathsf{r}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$; viol$(y)^{\mathsf{r}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$; act$(y)^{\mathsf{p}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$;
    else
      for all $\sigma^i \in \mathcal{L}_y$ s.t. $\exists \varsigma^i_{j,k}.\ \lambda(\varsigma^i_{j,k}) = \mathsf{A} \vee \lambda(\varsigma^i_{j,k}) = \mathsf{B}$ **do**
        if not $\exists h, k'.\ \lambda(\varsigma^i_{h,k'}) = \mathsf{B}$ **then** ▷ *Only* A*s*
          act$(y)^{\mathsf{r}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$; viol$(y)^{\mathsf{r}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$;
          act$(y)^{\mathsf{p}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$; viol$(y)^{\mathsf{p}}_{\mathsf{B},\mathsf{A}}$.add$(\varsigma^i_{j,k})$;
        else if not $\exists h, k'.\ \lambda(\varsigma^i_{h,k'}) = \mathsf{A}$ **then** ▷ *Only* B*s*
          act$(y)^{\mathsf{p}}_{\mathsf{B},\mathsf{A}}$.add$(\varsigma^i_{j,k})$; viol$(y)^{\mathsf{p}}_{\mathsf{A},\mathsf{B}}$.add$(\langle\texttt{NULL}, \varsigma^i_{j,k}\rangle)$
        else if $\lambda(\varsigma^i_{j,k}) = \mathsf{A}$ **then** ▷ *Both occur*
          act$(y)^{\mathsf{p}}_{\mathsf{A},\mathsf{B}}$.add$(\sigma^i_{j,k})$;
          if $\exists \varsigma^i_{j+h,k}, \varsigma^i_{j,k'}.\ h \geq 0$ **and** $\lambda(\varsigma^i_{j+h,k}) = \mathsf{B}$ **and** $\lambda(\varsigma^i_{j,k'}) = \mathsf{A}$ **then**
            viol$(y)^{\mathsf{p}}_{\mathsf{B},\mathsf{A}}$.add$(\varsigma^i_{j,k})$
          if not $\exists h, k'.\ \lambda(\varsigma^i_{h,k'}) = \mathsf{B}$ **and** $j + (\texttt{poly}\,\textbf{?}\,\pi(\varsigma^i_{j,k})\,\textbf{:}\,1) \leq h$ **then**
            viol$(y)^{\mathsf{r}}_{\mathsf{A},\mathsf{B}}$.add$(\varsigma^i_{j,k})$
          else act$(y)^{\mathsf{r}}_{\mathsf{A},\mathsf{B}}$.add$(\langle\varsigma^i_{j,k}, \varsigma^i_{h,k'}\rangle)$ **for all** h, k' **s.t.** $\lambda(\varsigma^i_{h,k'}) = \mathsf{B}$ **and** $j + (\texttt{poly}\,\textbf{?}\,\pi(\varsigma^i_{j,k})\,\textbf{:}\,1) \leq h$

5. Empirical Results

We exploit some time series datasets made available through the `sktime` library [29] while considering the Dyskinetic event dataset from our previous contribution [16]. Italy Power Demand [26] provides a dataset of univariate and equal-length time series, where the classification label is used to differentiate energy consumption patterns in the October-March period vs the April-September ones. The Basic Motions dataset[9] considers multivariate time series with 6 dimensions and distinguishing different motion types, walking, resting, running, and badminton. This dataset collects the x, y, and z axis information from wrist-watch accelerometer and a gyroscope; this dataset has been used to compare this motion dataset with the one discussed in our previous work while remarking the inherent more difficult nature of solving this other clinical problem. The Dyskinetic dataset [16] attempts at categorize Dyskinetic/Off events in terms of different drug assumption patterns as well as motor sensors information. Differently from the previous set of experiments [16], we now consider for both our algorithms and the competing MTSC approaches the full dataset also considering the active principles assumption intakes estimated using a rough approximation from current literature [6]. We also use the OSULeaf [21] dataset to remark how a simple image classification problem can be also represented as a time series classification problem: the univariate series were obtained by color image segmentation and boundary extraction (in the anti-clockwise direction) from digitized leaf images of six classes: Acer Circinatum, Acer Glabrum, Acer Macrophyllum, Acer Negundo, Quercus Garryana, and Quercus Kelloggii.

We consider such datasets for the following reasons: despite EMeriTAte was explicitly designed to capture trend correlations across different dimensions, we also use univariate datasets such as Italy Power Demand and OsuLeaf to show that this approach can be also applied to simpler dataset. Furthermore, we consider another multivariate dataset also considering different motor sensors to showcase the peculiar difference between the problem posed by the identification of the Dyskinetic Events from simply detecting different movement patterns. We provide some general statistics for these in Table 3.

Table 3. Dataset statistics

Dataset	$\lvert D\rvert$	avg_T Segments	#Dimensions	$avg_{\mathfrak{E}}\lvert\mathfrak{E}\rvert$	$\lvert\Sigma\rvert$	#Classes
Italy Power Demand	1096	1	1	24	13	2
Basic Motions	80	1	6	100	72	4
OsuLeaf	442	1	1	427	12	6
Dyskinetic Events	2	23	30	44	436	2

Our benchmarks were conducted on a Dell Precision mobile workstation 5760 running Ubuntu 22.04. The specifications of the machine include an Intel® Xeon(R) W-11955M CPU @ 2.60GHz × 16, 64GB DDR4 3200MHz RAM, with 500 GB of free disk space.

[9] `http://timeseriesclassification.com/description.php?Dataset=BasicMotions`

5.1. Run-time efficiency of EMeriTAte vs EMeriTAte+DF

Fig. 2 refers to the running times of both our proposed MTSC algorithms, from both our previous contribution and from the current paper. The first three phases refer to the A Priori explainability segment, while the last two refer to the Ad Hoc one. In both scenarios, we consider $\theta = 0$ for maximising the DECLARE recall, and consider in both circumstances a sensitivity parameter of $\varepsilon = 10^{-4}$. We considered the time required to fully process each dataset.

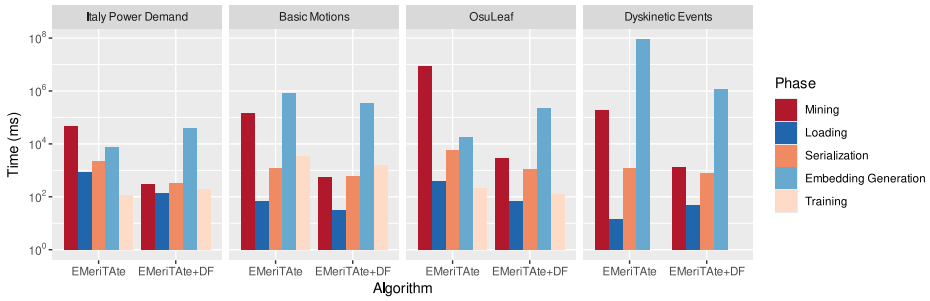


Fig. 2. Fine-Grained Mining+Learning times for the training phase of both EMeriTAte and EMeriTAte+DF.

Although the loading times of the datasets are comparable, we observe that the loading phase in this novel solution, which is different from the former, also includes a time series indexing phase. This then leads to a significant improvement in time minimisation, also ascribable to discarding some redundant DT activity labels. This is also witnessed in the serialization phase, which has slightly decreased despite the addition of catch22 payloads. Running times for the mining phase in EMeriTAte+DF remark that this algorithm is heavily dominated by the number of traces and segments being available as, in these situations, the former version of the algorithm was heavily hampered by conducting the mining and the embedding phases in two different times, thus requiring to load the data and index it several times. The newly proposed approach might have a more significant overhead in some situations. In fact, generating a dataful representation of the clauses requires accessing the constituents' payloads several times, thus increasing the number of times we access KnoBAB [11] for retrieving log information. Notwithstanding the former, this leads to a global running time decrease (Fig. 3).

5.2. Comparing competing MTSC approaches

We used the Python `sktime` library to ensure a uniform implementation for all competing approaches discussed in Section 2.3, for which we used the default hyperparameters setup from `sktime`. As all of the competing classifiers do not support classification of time series of different lengths, and given that these classifiers do not support time series varying their classification outcome in time, for the sole Dyskinetic event dataset, we consider the maximal projections belonging to the same class and, between each of these, we

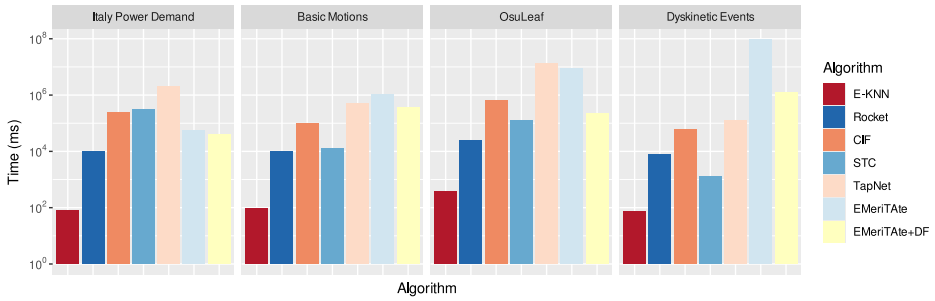


Fig. 3. Comparing the cumulative training times

consider multiple sliding windows of the same size. This overall builds up the number of time series to be classified by competing approaches.

Run-time efficiency. Fig. 3 compares the running time for all the MTSC approaches being trained over all the aforementioned datasets without splitting the data into training and testing datasets, thus using the same setting for the previous experiments. Competing approaches are not significantly impacted over the number of MTS dimensions as our approaches: please observe the tunning times for the Dyskinetic events where, despite the number of traces increased so to obtain multiple MTS of the same length, the competing approaches still provide a lesser running time if compared to ours. This is motivated by our solutions testing multiple possible correlations across dimension and activity labels, while other solutions such as Rocket and TapNet, attempt to capture cross-dimensional correlations through convolutions, thus flattening up the values via dimensionality reduction. Similar considerations can also be carried out by the other classifiers, which have not been specifically designed to consider correlations across dimensions: Euclidean distance for MTS only considers dimensionality-wise similarity, CIF considers variations within one fixed time and length sliding window while considering variations only within one single dimension, and STC considers shapelets only occurring within one single dimension. On the other hand, TapNet is heavily dominated by the number of traces being considered at training time and, across all the competitors, is the one always requiring a considerable amount on training time. EMeriTAte solution overtakes EMeriTAte+DF over the Dyskinetic dataset, while the other dataset consistently shows similar running times.

Classification outcomes. Fig. 4 provides the accuracy, precision, recall, and F1 score results across all the aforementioned datasets and competing approaches when testing the solutions with a 70%-30% split between training and testing dataset using stratified k-fold sampling. For both EMeriTAte and EMeriTAte+DF, we consider training the Decision Trees over the trace embeddings with a maximum depth of 5, so to minimise the chance of generating models overfitting the data. We run 10 experiments, for which we report the average score and the distance between the greatest and the lowest value obtained ($\pm$). When considering datasets with more than two classes, we use macro measures.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
E-KNN	96.47 ± 1.22	97.36 ± 1.19	95.58 ± 1.82	96.45 ± 1.24
Rocket [18]	97.02 ± 1.06	96.70 ± 1.72	97.39 ± 1.21	97.04 ± 1.05
CIF [32]	96.69 ± 1.82	96.73 ± 1.75	96.67 ± 3.03	96.69 ± 1.87
STC [23]	96.81 ± 1.37	96.81 ± 1.70	96.85 ± 3.03	96.81 ± 1.42
TapNet [43]	96.20 ± ε	94.76 ± 1.62	97.88 ± 0.91	96.28 ± ε
EMeriTAte [16]	99.59 ± ε	99.73 ± ε	99.45 ± 0.01	99.59 ± ε
EMeriTAte+DF	96.23 ± 1.52	96.24 ± 1.51	96.23 ± 1.52	96.23 ± 1.52

(a) Italy Power Demand

Algorithm	Accuracy (%)	M. Precision (%)	M. Recall (%)	M. F1 (%)
E-KNN	62.50 ± 8.33	53.09 ± 3.79	62.50 ± 8.33	53.91 ± 7.15
Rocket [18]	99.58 ± 2.08	99.64 ± 1.79	99.58 ± 2.08	99.58 ± 2.10
CIF [32]	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
STC [23]	99.58 ± 2.08	99.64 ± 1.79	99.58 ± 2.08	99.58 ± 2.10
TapNet [43]	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
EMeriTAte [16]	91.66 ± 8.33	92.69 ± 7.81	91.66 ± 8.33	91.51 ± 8.57
EMeriTAte+DF	98.75 ± 2.08	98.92 ± 1.78	98.75 ± 2.08	98.74 ± 2.10

(b) Basic Motions

Algorithm	Accuracy (%)	M. Precision (%)	M. Recall (%)	M. F1 (%)
E-KNN	63.46 ± 6.02	65.43 ± 5.79	62.59 ± 6.53	62.98 ± 6.54
Rocket [18]	96.32 ± 2.26	96.77 ± 1.98	95.98 ± 2.41	96.25 ± 2.24
CIF [32]	84.44 ± 3.01	86.55 ± 3.25	82.38 ± 4.94	83.29 ± 4.55
STC [23]	92.86 ± 3.01	93.35 ± 3.23	91.72 ± 3.78	92.25 ± 3.56
TapNet [43]	82.71 ± 1.50	88.29 ± 1.06	78.56 ± 5.49	78.59 ± 6.65
EMeriTAte[16]	79.62 ± 8.27	83.72 ± 7.78	80.13 ± 8.50	80.39 ± 8.56
EMeriTAte+DF	99.70 ± 0.01	99.63 ± 0.01	99.70 ± ε	99.65 ± ε

(c) OsuLeaf

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
E-KNN	12.14 ± 10.71	10.06 ± 18.18	12.86 ± 28.57	11.21 ± 22.22
Rocket [18]	12.14 ± 10.71	7.80 ± 18.18	10.00 ± 28.57	8.65 ± 22.22
CIF [32]	12.14 ± 10.71	4.58 ± 16.67	5.71 ± 21.43	5.08 ± 18.75
STC [23]	25.71 ± 17.86	14.44 ± 22.22	12.86 ± 28.57	13.02 ± 25.00
TapNet [43]	15.00 ± 10.71	15.19 ± 18.18	20.0 ± 28.57	17.09 ± 22.22
EMeriTAte [16]	98.57 ± 7.14	100.0 ± 0.00	97.14 ± 14.28	98.33 ± 8.33
EMeriTAte+DF	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00

(d) Dyskinetic Events

Fig. 4. Training Results over the datasets of interest. Macro metrics are used for datasets containing more than 2 classes. Numbers in blue (red) remark the best (worst) results

Concerning the Dyskinetic dataset, we kept different training/testing splits from the ones in our previous paper.

We observe that, for three datasets out of four, EMeriTAte+DF outperforms our previous implementation. The major score distance between these two solutions can be found on the OsuLeaf dataset: this can be explained as the main difference between the two solutions resides in the usage of the data refinement, while the discard of DT patterns seems not to have interfered with the classification outcome. For the Italy Power Demand dataset, we see that our first solution outperforms the current implementation: given the above, this can be ascribed to the presence of several fluctuations events which were not discarded in our previous solution.

By comparing our solutions to the competing approaches, no competitor can constantly outperform the others across all the datasets and, when EMeriTAte+DF did not achieve maximum scores, it still achieved $\approx 99\%$ of accuracy, macro precision, macro recall, and F1 score. E-KNN scores low values for datasets such as the former and OsuLeaf, thus remarking the impracticality of merely considering time series similarity as a valid pathway for classification outcomes for real-world datasets.

While comparing the results over the Dyskinetic dataset, it is clear that the best approaches were the ones considering silhouette (STC) and attention ones (TapNet) while still scoring metrics way below 50%. This can be motivated as follows: either the process of splitting the time series lost some important correlation information or truncated some patterns, or these solutions still cannot account for extracting the best behavioural predictors for the dataset of interest. After analysing the models extracted from EMeriTAte+DF, we can clearly see that the classification always refers to correlations across different dimensions. Given the explanation being extracted, this then remarks that the dataset of interest is heavily characterized by correlations across dimensions that could not be adequately captured by the pre-existing solutions.

6. Conclusion and Future Works

This paper provides the first algorithm towards a generalization of existing algorithms in the literature for classifications of temporal behaviours: in fact, we use a purely event-driven technique to classify multivariate numerical temporal sequences. By doing so, we have shown how it is possible to appropriately discretize a numerical representation, maintaining as much of the information in the data as possible. To appropriately characterize the numerical data, we have also deemed it necessary to further generalize the representation of temporal data to events, prescribing that multiple distinct events of different durations can occur at a given instant of time. From the generalization of the deviance mining algorithms, we obtain a new classifier that can be used to classify numerical time series with good results in precision, recall, and accuracy. Subsequent studies will also establish this mechanism's validity in characterizing event-based data.

Future works will investigate whether considering additional events not currently mined in the indexed DT phase will make EMeriTAte+DF reach the same precision, recall, and accuracy results from our previous contribution, EMeriTAte, without tampering the running time of the subsequent *ad hoc* phase. This will be supported by applying the experiments on novel datasets, while investigating whether the current solution is suitable to address the aforementioned challenge. Despite preliminary results remark that the

non-negligible running time of this latter phase might be due to the considerable presence of events to ascertain, considering larger amounts of data will necessarily require further improving over the presently proposed algorithms. As both our solutions are implemented using KnoBAB [11] as a temporal-driven database, future works will also require to extend it so to move from a in-memory data representation to a one using secondary memory. We will also consider the possibility of applying the general-to-specific mining approach that we previously investigated over Bolt2 for mining DT patterns, so to achieve mining efficiency without resorting to discarding noisy or redundant events.

Experiments over the datasets from Table 3 lead dataful specifications describing the classes showing only refinements over the activation conditions, but not on the target ones. We will also test the possibility of other datasets also requiring the definition of correlation conditions between activated and targeted events, thus requiring the definition of comparisons between activated and targeted constituents' payloads. Despite the possibility of doing so though weighted oblique trees [42], this decision will come at the cost of significantly increasing computational overhead over the refinement phase. As a byproduct of this, further work needs also to be done to further improve over the fitting of oblique decision trees.

Last, will also explore the possibility of time series forecasting from the mined and learned model. Given the dualism between polyadic traces and MTS and given the possibility of generating dataful traces out of conjunctive models [9], it should then be required to unpack the catch22 values being generated to reconstruct actual time series values. Thus, future works will need to address this challenge and verify its possibility. After achieving this, a further extension will be required to generate traces out of declarative models expressed as arbitrary propositions. We will then need to compare such a solution with state-of-the-art approaches in both machine learning [34,41,33] and statistical [40,28] literature.

Acknowledgments. SDD was supported by the National Institute for Health and Care Research (NIHR) Newcastle Biomedical Research Centre based at The Newcastle upon Tyne Hospitals NHS Foundation Trust, Newcastle University and the Cumbria, Northumberland and Tyne and Wear (CNTW) NHS Foundation Trust. The research was also supported by NIHR Newcastle Clinical Research Facility (CRF) Infrastructure funding. SDD was also supported by the Innovative Medicines Initiative 2 Joint Undertaking (IMI2 JU) project IDEA-FAST - Grant Agreement 853981. SDD was supported by the UK Research and Innovation (UKRI) Engineering and Physical Sciences Research Council (EPSRC) (Grant Ref: EP/X031012/1 and Grant Ref: EP/X036146/1).

References

1. van der Aalst, W.M.P.: Discovering Directly-Follows Complete Petri Nets from Event Data, pp. 539–558. Springer Nature Switzerland, Cham (2022), doi.org/10.1007/978-3-031-15629-8_29
2. Acampora, G., Vitiello, A., Di Stefano, B., van der Aalst, W., Gunther, C., Verbeek, E.: IEEE 1849: The XES Standard: The Second IEEE Standard Sponsored by IEEE Computational Intelligence Society. IEEE Comp. Int. Mag. 12(2), 4–8 (2017), doi.org/10.1109/MCI.2017.2670420
3. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM 26(11), 832–843 (Nov 1983), doi.org/10.1016/B978-1-4832-1447-4.50033-X

4. Andrzejewski, W., Bębel, B., Boiński, P., Wrembel, R.: On tuning parameters guiding similarity computations in a data deduplication pipeline for customers records: Experience from a r&d project. Information Systems 121, 102323 (2024), https://www.sciencedirect.com/science/article/pii/S030643792300159X
5. Anselma, L., Bottrighi, A., Montani, S., Terenziani, P.: Extending BCDM to cope with proposals and evaluations of updates. IEEE Transactions on Knowledge and Data Engineering 25(3), 556–570 (2013), doi.org/10.1109/TKDE.2011.170
6. Arav, Y., Zohar, A.: Model-based optimization of controlled release formulation of levodopa for Parkinson's disease. Scientific Reports 13(1), 15869 (Sep 2023), doi.org/10.1038/s41598-023-42878-5
7. Avolio, M., Fuduli, A., Vocaturo, E., Zumpano, E.: On detection of diabetic retinopathy via multiple instance learning. In: Proceedings of the International Database Engineered Applications Symposium Conference, IDEAS 2023, Heraklion, Crete, Greece, May 5-7, 2023. pp. 170–176. ACM (2023), doi.org/10.1145/3589462.3589490
8. Bergami, G.: On efficiently equi-joining graphs. In: Proceedings of the 25th International Database Engineering & Applications Symposium. p. 222–231. IDEAS '21, Association for Computing Machinery, New York, NY, USA (2021), doi.org/10.1145/3472163.3472269
9. Bergami, G.: Fast synthetic data-aware log generation for temporal declarative models. In: Hartig, O., Yoshida, Y. (eds.) Proceedings of the 6th Joint Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), Seattle, WA, USA, 18 June 2023. pp. 7:1–7:9. ACM (2023), https://doi.org/10.1145/3594778.3594881
10. Bergami, G.: DECLARE*d*: A Polytime LTL_f Fragment. Logics 2(2), 79–111 (2024), doi.org/10.3390/logics2020004
11. Bergami, G., Appleby, S., Morgan, G.: Specification mining over temporal data. Computers 12(9) (2023), doi.org/10.3390/computers12090185
12. Bergami, G., Bertini, F., Montesi, D.: Hierarchical embedding for DAG reachability queries. In: Proceedings of the 24th Symposium on International Database Engineering & Applications. IDEAS '20, Association for Computing Machinery, New York, NY, USA (2020), doi.org/10.1145/3410566.3410583
13. Bergami, G., Fox, O.R.: How explainable really is AI? benchmarking explainable AI. Logics 3(3) (2025), doi.org/10.3390/logics3030009
14. Bergami, G., Francescomarino, C.D., Ghidini, C., Maggi, F.M., Puura, J.: Exploring business process deviance with sequential and declarative patterns. CoRR abs/2111.12454 (2021), doi.org/10.48550/arXiv.2111.12454
15. Bergami, G., Maggi, F.M., Montali, M., Peñaloza, R.: Probabilistic trace alignment. In: 2021 3rd International Conference on Process Mining (ICPM). pp. 9–16 (2021), doi.org/10.1109/ICPM53251.2021.9576856
16. Bergami, G., Packer, E., Scott, K., Din, S.D.: Predicting dyskinetic events through verified multivariate time series classification. In: Chbeir, R., Ilarri, S., Manolopoulos, Y., Revesz, P.Z., Bernardino, J., Leung, C.K. (eds.) Database Engineered Applications - 28th International Symposium, IDEAS 2024, Bayonne, France, August 26-29, 2024, Proceedings. Lecture Notes in Computer Science, vol. 15511, pp. 49–62. Springer (2024), doi.org/10.1007/978-3-031-83472-1_4
17. De Giacomo, G., Maggi, F.M., Marrella, A., Patrizi, F.: On the Disruptive Effectiveness of Automated Planning for LTL_f-Based Trace Alignment. In: AAAI'17. AAAI press (2017), doi.org/10.1609/aaai.v31i1.11020
18. Dempster, A., Petitjean, F., Webb, G.I.: ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min. Knowl. Discov. 34(5), 1454–1495 (2020), doi.org/10.1007/s10618-020-00701-z

19. Donzé, A.: On signal temporal logic. In: Runtime Verification - 4th International Conference, RV 2013, Rennes, France, September 24-27, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8174, pp. 382–383. Springer (2013), doi.org/10.1007/978-3-642-40787-1_27
20. Färber, M.: Denotational semantics and a fast interpreter for jq. CoRR abs/2302.10576 (2023), https://doi.org/10.48550/arXiv.2302.10576
21. Gandhi, A.: Content-Based Image Retrieval: Plant Species Identification. Master's thesis, Oregon State University (2002), https://web.engr.oregonstate.edu/~tgd/leaves/acer-quercus.pdf
22. Gellér, Z., Kurbalija, V., Ivanović, M.: FAP: A time series analysis and mining framework for scientific and practical applications. Comput. Sci. Inf. Syst. Forthcoming (2025)
23. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Mining and Knowledge Discovery 28(4), 851–881 (Jul 2014), doi.org/10.1007/s10618-013-0322-1
24. Huo, X., Hao, K., Chen, L., song Tang, X., Wang, T., Cai, X.: A dynamic soft sensor of industrial fuzzy time series with propositional linear temporal logic. Expert Systems with Applications 201, 117176 (2022), doi.org/10.1016/j.eswa.2022.117176
25. Inoue, T., Kubota, K., Ikami, T., Egami, Y., Nagai, H., Kashikawa, T., Kimura, K., Matsuda, Y.: Clustering method for time-series images using quantum-inspired digital annealer technology. Communications Engineering 3(1), 10 (Jan 2024), https://doi.org/10.1038/s44172-023-00158-0
26. Keogh, E., Wei, L., Xi, X., Lonardi, S., Shieh, J., Sirowy, S.: Intelligent icons: Integrating lite-weight data mining and visualization into GUI operating systems. In: Sixth International Conference on Data Mining (ICDM'06). pp. 912–916 (2006), doi.org/10.1109/ICDM.2006.90
27. Leno, V., Dumas, M., Maggi, F.M., La Rosa, M., Polyvyanyy, A.: Automated discovery of declarative process models with correlated data conditions. Information Systems 89, 101482 (2020), doi.org/10.1016/j.is.2019.101482
28. Livera, A.M.D., Hyndman, R.J., Snyder, R.D.: Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association 106(496), 1513–1527 (2011), doi.org/10.1198/jasa.2011.tm09771
29. Löning, M., Bagnall, A.J., Ganesh, S., Kazakov, V., Lines, J., Király, F.J.: sktime: A unified interface for machine learning with time series. CoRR abs/1909.07872 (2019), doi.org/10.48550/arXiv.1909.07872
30. Lubba, C.H., Sethi, S.S., Knaute, P., Schultz, S.R., Fulcher, B.D., Jones, N.S.: catch22: CAnonical Time-series CHaracteristics - selected through highly comparative time-series analysis. Data Min. Knowl. Discov. 33(6), 1821–1852 (2019), doi.org/10.1007/S10618-019-00647-X
31. Wes McKinney: Data Structures for Statistical Computing in Python. In: Stéfan van der Walt, Jarrod Millman (eds.) Proceedings of the 9th Python in Science Conference. pp. 56 – 61 (2010)
32. Middlehurst, M., Large, J., Bagnall, A.: The canonical interval forest (CIF) classifier for time series classification. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 188–195 (Dec 2020), doi.org/10.1109/BigData50022.2020.9378424
33. Olivares, K.G., Challú, C., Garza, A., Canseco, M.M., Dubrawski, A.: NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022 (2022), https://github.com/Nixtla/neuralforecast
34. Pecev, P., Rackovic, M.: LTR-MDTS structure - a structure for multiple dependent time series prediction. Comput. Sci. Inf. Syst. 14(2), 467–490 (2017), doi.org/10.2298/CSIS150815004P
35. Petermann, A., Micale, G., Bergami, G., Pulvirenti, A., Rahm, E.: Mining and ranking of generalized multi-dimensional frequent subgraphs. In: Twelfth International Conference on Digital

Information Management, ICDIM 2017, Fukuoka, Japan, September 12-14, 2017. pp. 236–245. IEEE (2017), doi.org/10.1109/ICDIM.2017.8244685
36. Pešić, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: EDOC. pp. 287–300. IEEE Computer Society (2007), doi.org/10.1109/EDOC.2007.14
37. Rost, C., Gómez, K., Täschner, M., Fritzsche, P., Schons, L., Christ, L., Adameit, T., Junghanns, M., Rahm, E.: Distributed temporal graph analytics with GRADOOP. VLDB J. 31(2), 375–401 (2022), doi.org/10.1007/s00778-021-00667-4
38. Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A.: The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery 35(2), 401–449 (Mar 2021), doi.org/10.1007/s10618-020-00727-3
39. Seshia, S.A., Sadigh, D., Sastry, S.S.: Toward verified artificial intelligence. Commun. ACM 65(7), 46–55 (Jun 2022), doi.org/10.1145/3503914
40. Swain, S., Nandi, S., Patel, P.: Development of an ARIMA model for monthly rainfall forecasting over Khordha District, Odisha, India. In: Sa, P.K., Bakshi, S., Hatzilygeroudis, I.K., Sahoo, M.N. (eds.) Recent Findings in Intelligent Computing Techniques. pp. 325–331. Springer Singapore, Singapore (2018), doi.org/10.1007/978-981-10-8636-6_34
41. Tang, H., Zhang, C., Jin, M., Yu, Q., Wang, Z., Jin, X., Zhang, Y., Du, M.: Time series forecasting with llms: Understanding and enhancing model capabilities. SIGKDD Explor. Newsl. 26(2), 109–118 (Jan 2025), doi.org/10.1145/3715073.3715083
42. Yang, B.B., Shen, S.Q., Gao, W.: Weighted oblique decision trees 33(01), 5621–5627, doi.org/10.1609/aaai.v33i01.33015621
43. Zhang, X., Gao, Y., Lin, J., Lu, C.: TapNet: Multivariate time series classification with attentional prototypical network 34(4), 6845–6852 (2020), doi.org/10.1609/aaai.v34i04.6165

Giacomo Bergami, at the time of the writing, is a Lecturer (Assistant Professor) in Game Technology at Newcastle University, and is the PI for the LOGic for Data Science (LogDS) group. We advocate that Logic should be used as a main driving force for Data Science to obtain Verified and Explainable Artificial Intelligence. He received the Bachelor's and Master's Degree in Computer Science with full marks (both with a final passing grade of 110/110 with Merit). From the 1st of November 2014 to the 1st of November 2017, my PhD dissertation at Bologna University, Italy, CSE Dept. was supervised by Prof. Danilo Montesi: the research focused on two novel graph operators: graph joins (for combining data) and graph nesting (for structural aggregations), contextualised within a clinical data integration setting. Before obtaining my lectureship position, I gained international experience by working abroad, both as a Visiting PhD Student (UniversitÃ¤t Leipzig, Germany) and as a PostDoc Researcher (University of Florida, USA).

Kirsty Scott is a Postdoctoral Research Associate at Newcastle University since 2023, where she works in the Brain and Movement Team. She got the PhD in Mechanical Engineering at the University of Sheffield.

Emma Packer is a PhD student at Newcastle University. Emma's PhD explores how medication influences everyday walking in people with Parkinson's. Emma has an undergraduate degree in Biomedical Sciences, and a masters degree in Neuroscience.

Silvia Del Din is a bioengineer by background and a Reader (Associate Professor) in Digital Healthcare for translational applications at Newcastle University. She has attracted >£2.5 Million in funding as PI and Co-I of various digital health projects (e.g. from MRC, EPSRC, IMI-EU (Mobilise-D, IDEA-FAST)). She leads the wearable/digital health technology team within the Brain and Movement Research Group where she has contributed to build the digital health theme. Her research aims at innovating digital healthcare and translating it into clinical practice. Her expertise focuses on enhancing the use of digital health technology together with innovative data analysis techniques to enable remote monitoring and facilitate clinical management in ageing and neurodegenerative disorders (e.g. Parkinson's disease). She has published 100 full papers. She is part of the Editorial Board of the IEEE Open Journal of Engineering in Medicine and Biology and an Associate Editor for the IEEE Journal of Biomedical and Health Informatics. She serves as a member on national and international scientific committees (e.g., Parkinson's UK College of Experts, ISPGR Scientific Content Committee); since 2022 she is the elected Chair of the Early Career Researchers Working Group of the BioMedEng Association Council.

Received: March 03, 2025; Accepted: September 15, 2025.

Computer Science and Information Systems 23(1):475–512 https://doi.org/10.2298/CSIS250401080B

Siren Federate: Bridging Document, Relational, and Graph Models for Exploratory Graph Analysis*

Georgeta Bordea[1], Stéphane Campinas[2], Matteo Catena[2], and Renaud Delbru[2]

[1] L3i, La Rochelle University – La Rochelle, France
georgeta.bordea@univ-lr.fr
https://l3i.univ-larochelle.fr/
[2] Siren – Galway, Ireland
{stephane.campinas,matteo.catena, renaud.delbru}@siren.io
https://siren.io/

Abstract. Investigative workflows require interactive exploratory analysis on large heterogeneous knowledge graphs. Current databases show limitations in enabling such task. This paper discusses the architecture of Siren Federate, a system that efficiently supports exploratory graph analysis by bridging document-oriented, relational and graph models. Technical contributions include distributed join algorithms, adaptive query planning, query plan folding, semantic caching, and semi-join decomposition for path query. Semi-join decomposition addresses the exponential growth of intermediate results in path-based queries. Experiments show that Siren Federate exhibits low latency and scales well with the amount of data, the number of users, and the number of computing nodes.

Keywords: Exploratory Graph Analysis, Knowledge Graph, Database and Information System Architecture, Distributed Join Algorithms, Document-oriented Database.

1. Introduction

Investigative Intelligence encompasses domains such as law enforcement, financial compliance, cyber-threat analysis, and investigative journalism, in which professionals uncover hidden patterns and relationships by analyzing diverse and interconnected data sources [1, 3,4,22,25]. This analysis enables the detection of emerging threats, verification of claims, and identification of concealed connections, contributing to transparency, accountability, and informed decision-making in contexts involving corruption, misinformation, and complex criminal activities. These scenarios require integrating structured records, semi-structured logs, unstructured text, and increasingly, multi-modal content such as images and videos. A central challenge lies in enabling rapid, flexible, and iterative analysis over massive volumes of heterogeneous data.

Knowledge Graphs (KGs) [21] provide a unifying abstraction for integrating such diverse data into a single graph representation, where vertices represent entities and edges their relationships. This unified view enables analysts to "connect the dots" across modalities and data silos. However, scaling exploratory graph analysis to billions of entities – while maintaining interactive responsiveness – remains an open challenge.

* This manuscript is an extended version of the paper "Siren Federate: Bridging the Gap between Document and Relational Data Systems for Efficient Exploratory Graph Analysis" [8] which has appeared in the proceedings of the 28th International Symposium on Database Engineered Applications (IDEAS 2024).

Current database technologies present significant limitations for investigative workflows. Native graph databases excel at localized traversals but struggle with global analytical workloads and scalability. Relational databases provide robust structured data operations but lack flexibility for heterogeneous data and graph processing. Both approaches face a common challenge with path enumeration, where intermediate result explosion undermines performance at scale.

To address these challenges, we present *Siren Federate*, a system that integrates relational and graph querying capabilities into Elasticsearch, a distributed Information Retrieval (IR) system [17] written in Java and based on the Lucene search library [19]. By leveraging Elasticsearch's advanced text search, multi-dimensional search, vector search, and horizontal scalability, Siren Federate extends these strengths by incorporating distributed join algorithms, adaptive query planning, and semantic caching strategies. The result is a system that supports iterative exploration of large, multi-modal knowledge graphs, providing the multi-hop expansions and relational filtering essential for investigative workflows, all within interactive, sub-second to second response times.

This manuscript describes how Siren Federate bridges document-oriented, relational and graph models. The architecture presented here has been refined through a decade of R&D and production deployments, including installations that span hundreds of nodes and process petabyte-scale datasets. Rather than describing a theoretical system or research prototype, we share insights and optimizations derived from addressing real-world investigative intelligence challenges at scale.

We first outline the investigative intelligence domain and practical requirements in Sec. 2. We then present an overview of related work in Sec. 3, which examines general limitations of current database technologies and specific approaches to path query processing. Next, Sec. 4 discusses how to bridge the document, relational, and graph models to enable exploratory graph analysis, while Sec. 5 presents Siren Federate's architecture, focusing on its distributed join algorithms, adaptive query optimization, query plan folding, and semantic caching. In Sec. 6, we introduce a novel *Semi-Join Decomposition* technique that addresses the exponential growth of intermediate results in path-based queries. Finally, in Sec. 7, we evaluate Siren Federate's performance using large synthetic datasets and the LDBC Financial Benchmark [51]. These experiments demonstrate our system's capability to support complex investigative scenarios while maintaining responsiveness suitable for real-time exploration. By unifying IR, relational, and graph querying, Siren Federate offers a foundation on which investigative intelligence systems can efficiently handle increasingly complex and diverse data.

2. Investigative Intelligence Background

This section provides essential background on investigative intelligence workflows and their requirements for database systems. We first outline the domain and its distinctive data exploration patterns, then identify the specific technical capabilities required to support these workflows.

2.1. Domain Overview

Investigative workflows in law enforcement and cybersecurity increasingly rely on large knowledge graphs to detect threats by connecting data involving suspects, organizations,

financial transactions, and network events. Similar requirements arise in investigative journalism and financial compliance. Journalists must collate and verify evidence from data leaks, government archives, or user-generated content; compliance teams need to trace suspicious transaction patterns across multiple accounts to flag potential money laundering or fraud. Malicious actors exploit the increasing volume and complexity of data to blend in and operate undetected. Therefore, analysts require the capability to iteratively explore complex chains of relationships hidden within massive volumes of heterogeneous data, traversing across textual, numeric, relational, and multimedia modalities.

Knowledge Graphs (KGs) [21] serve as the foundation for these workflows by integrating diverse data sources — structured data (e.g., relational tables), semi-structured data (e.g., JSON, XML), and unstructured data (e.g., text, images, multimedia) — into a unified graph representation. This consolidated view enables analysts to "connect the dots" across data types, facilitating the identification of threats, cross-border criminal activity, or hidden corruption within complex corporate structures. KGs also enhance transparency by visualizing relationships, providing reasoning pathways, and ensuring traceable provenance – critical components in building and validating investigative narratives.

Investigations rarely proceed as single, linear queries. Instead, they involve exploratory, iterative analysis aimed at discovering hidden patterns and generating new leads [31]. Analysts typically begin with limited information – such as a suspicious keyword in contractual documents or an unusual shipping reference – and progressively expand their exploration across structured data, full-text documents, and multi-modal features to uncover additional evidence and refine their hypotheses. The investigative system must guide users through iterative searching, filtering, and data drilling, enabling rapid identification of entities of interest. Graph analytics capabilities, such as path-finding, centrality, and community detection, further assist in discovering relevant subgraphs and patterns.

Siren's platform [7] addresses these investigative needs by combining multiple data interaction paradigms – including search, analytic dashboards, set-to-set navigation, and graph visualization – into a unified exploration model. For example, in a Signals Intelligence scenario, individuals, cellphone data, call records, text messages, and network cells form a complex, interconnected graph. Set-to-set navigation [44] (a type of relational faceted navigation) enables analysts to easily navigate among these interrelated datasets, as applying filters to one set dynamically updates all related entities, supporting iterative exploration. An analyst may start with a textual clue (e.g., a suspicious message), then iteratively expand towards linked phone records, geolocations, or associated media. Graph visualization further helps in understanding the interconnected datasets, identifying clusters, and answering targeted questions such as *"Which individuals own these phones? Are they connected to the same network cells? Are they part of a coordinated group?"*.

2.2. Requirements for Investigative Intelligence Systems

To support investigative workflows, exploratory graph analysis system must handle diverse workloads that combine multiple querying paradigms:

1. Information Retrieval workloads, including: (a) Standard and advanced full-text searches over textual documents (such as social media posts, open web data, forensic reports, etc.), including keyword-based search, fuzzy matching, wildcard queries, and phonetic similarity to account for misspellings or alternate spellings. (b) Relevance ranking and

highlighting for prioritizing results and identifying key passages. (c) Semantic and vector-based searching across multi-modal data. (d) Multi-dimensional search, including geo-spatial (e.g., GPS coordinates, polygon-based region), temporal (e.g., event histories, timestamps), and numeric ranges (e.g., IP intervals, financial amounts).
2. Relational and graph database workloads [5], such as: (a) OLTP-style queries for rapid, localized retrieval of nodes and edges (e.g., fetch all immediate connections of a particular entity). (b) OLAP-style analytical queries that aggregate data over large graph segments, such as frequency distributions or statistical summarizations over extensive graph regions. (c) Neighborhood queries and localized traversals within limited hops (degrees of separation). (d) Path-finding queries and traversals across long graph paths (e.g., shortest paths). (e) Global graph analytical operations such as pattern matching, keyword-based subgraph searching, community detection, centrality analyses.
3. Hybrid exploratory workloads, involving combinations of IR and relational-graph queries, such as: (a) Multi-hop filtered expansions constrained by textual relevance, semantic similarity, or attribute conditions. (b) Interactive drill-down and faceted exploration, integrating relational joins and full-text or semantic filtering to iteratively narrow down datasets based on user-selected criteria. (c) Cross-modal correlation, where vector searches on one data modality dynamically filter relational expansions on another modality (such as numeric financial records or geospatial locations).

These workloads require flexible data modeling and efficient query execution across heterogeneous data. Additionally, investigative systems must ensure fast response time, as slow interactions can disrupt cognitive flow [33]. Supporting iterative, multi-step queries over large knowledge graphs while preserving sub-second to second responsiveness remains a critical challenge. Moreover, analysts increasingly require multi-modal querying across embeddings derived from text, images, or audio. Despite their importance, such capabilities are underrepresented in KG systems and remain an active research area [27].

Given these requirements, it is important to examine how current database technologies address these challenges and where they fall short. In the next section, we review related work covering both general limitations of existing database technologies and specific approaches to path query processing, which will provide context for our approach.

3. Related Work

This section reviews existing database technologies and query processing approaches relevant to the challenges of exploratory graph analysis in investigative intelligence. We first examine general limitations of current database technologies for handling heterogeneous knowledge graphs at scale, then provide a focused analysis of specific techniques for path query evaluation.

3.1. Limitations of Current Database Technologies

Despite advances, current graph and relational database solutions often fall short in meeting the requirements of investigative workflows [5].

Native graph databases excel at localized traversals but may not be effective for global analytical workloads. [49,59] noted challenges in scaling to very large graphs and handling

complex pattern-matching queries. [37] highlight limitations in current graph query languages and call for richer query paradigms combining structured and IR-style search. Many graph databases provide limited built-in support for global graph analytics, typically offering only basic functions while requiring external libraries for advanced algorithms [59]. They also often lack robust semantic similarity search capabilities [63], although systems like Neo4j have recently added vector indexing.

Relational databases handle structured data and global analytics effectively, but they lack flexibility in managing heterogeneous data and exhibit limited native support for graph and advanced text processing [12]. When used for graph workloads, they produce "huge redundant intermediate data" during join operations [57], with multi-hop queries degrading by "several orders of magnitude" compared to native graph approaches.

Multi-model databases attempt to integrate multiple data models (key-value, document, graph), but originating from specific data paradigms frequently leads to inefficiencies in complex workloads [66]. Cross-model query optimization presents significant challenges, particularly for hybrid queries spanning data models. Achieving optimal performance across relational, graph, and full-text search remains difficult due to challenges in query processing, schema design, and indexing [35]. Finally, polyglot architectures – combining specialized systems like text search engines, graph databases, and vector stores – add operational complexity, fragment user workflows, and create challenges in maintaining data consistency and coordinated queries [28, 42].

As a result, analysts face operational complexity, fragmented user experiences, and missed investigative insights when queries cannot seamlessly span data modalities. To the best of our knowledge, no single system fully integrates advanced IR over multi-modal data with scalable relational joins and graph analytics. In particular, joins are fundamental to robust graph analysis [64]. We argue that combining the strengths of IR, relational, and graph paradigms within a unified framework is a balanced solution to these challenges.

This motivated our adoption of Elasticsearch, which offers a flexible data model, horizontal scalability, and advanced IR capabilities – including full-text search, multidimensional indexing, and vector similarity. However, Elasticsearch lacks native support for query-time joins across shards, requiring indexing-time preprocessing and often resorting to denormalization.[3] These constraints only adequately supports simple hierarchical relationships, and are inadequate for complex graph data. Siren Federate addresses these challenges by integrating scalable, query-time distributed joins and graph capabilities into Elasticsearch. Sec. 4 and 5 detail our integration approach and system architecture that enable these capabilities.

3.2. Path Query Techniques and Limitations

Path queries represent a specific challenge within the broader limitations of current database technologies. For example, an investigator might ask to *"Find all communication chains up to length four linking person A to person B through calls and transactions"*. Such queries require exploring many possible paths, often causing an explosion of intermediate results and underscoring the complexity of path enumeration. Existing approaches to path enumeration fall into two main categories: one rooted in graph traversal, through variants

[3] `https://www.elastic.co/guide/en/elasticsearch/reference/8.17/joining-queries.html`

of the breadth-first search (BFS), and another grounded in relational algebra, using join operators to express multi-hop relationships.

Graph Traversal (BFS-Based) Approaches BFS and its derivatives are the most intuitive methods for computing shortest paths [39]. These algorithms proceed in iterative "waves", expanding the frontier at each step. To enumerate all shortest paths between a source and target node, BFS is often combined with a backtracking phase over a shortest-path direct acyclic graph (DAG) built during traversal [62]. Following the graph model defined in Sec. 4, a typical BFS-based workflow involves: (1) Starting from a source node u, retrieving immediate neighbors based on a field-matching condition; (2) Appending these neighbors into a frontier queue while maintaining a visited set; (3) Iteratively expanding each node in the frontier, maintaining parent pointers to enable backtracking for path reconstruction; (4) Continuing until the target node v is reached, or the maximum path length L is reached. Even after finding the target vertex, the algorithm must continue exploring any remaining frontier nodes at the current depth to ensure all shortest paths of equal minimal length are identified. As traversal proceeds in a dense graph, each level may yield an exponential number of partial paths to track. These intermediate structures must be maintained until the algorithm determines which paths are truly minimal. Even with compact representations like predecessor DAGs, the memory footprint of intermediate state grows rapidly.

Native graph databases like Neo4j[4] and Memgraph[5] implement built-in functions to enumerate all minimal-length paths between two nodes. Internally, these engines rely on BFS-style traversal, maintaining a shortest-path DAG and exploring it to produce all valid path permutations. [38] highlight the critical issue of exponential intermediate path sets in graph database queries, proposing compact path multiset representations to manage intermediate result explosion. [47] similarly demonstrate the combinatorial explosion of candidate paths in hop-constrained path enumeration, proposing efficient polynomial-delay algorithms to mitigate this issue. [18] extend BFS approaches to billion-scale distributed graphs by combining BFS and DFS searches with aggressive pruning methods to control intermediate result sizes. [29] propose an FPGA-based batching strategy using DFS principles to curb BFS-related exponential memory overhead effectively.

Although recent approaches offer meaningful improvements through indexing, pruning, and hardware optimizations, BFS-based traversal remains fundamentally limited by exponential growth in intermediate state and runtime, particularly in dense graphs with high branching factors. The Resonance Algorithm [32], while innovative in its wave propagation technique, also suffers from maintaining extensive intermediate path states, offering only limited mitigation of path explosion.

Relational Algebra (Join-Based) Approaches An alternative class of methods models path traversal as a sequence of relational joins $D_1 \bowtie \cdots \bowtie D_{l+1}$ where each join captures an edge in the traversal path. Formally, given a path length $l \leq L$, the query would be structured as $D_1 \bowtie \cdots \bowtie D_{l+1}$ where each D_i represents an index participating in the

[4] `https://neo4j.com/docs/graph-data-science/current/algorithms/pathfinding/`

[5] `https://memgraph.com/docs/advanced-algorithms/deep-path-traversal`

query. Typically, the indices involved ($D_1, \ldots, D_{l+1}$) and their order are determined by analyzing the underlying ontology or schema that defines relationships between document types (see Sec. 4). Executing such inner-join chains explicitly enumerates all possible paths connecting the source node $u \in D_1$ and the target node $v \in D_{l+1}$.

This approach benefits from mature query optimization infrastructure, data locality, and bulk-processing efficiencies inherent in relational algebra. For instance, [23] explored using standard relational operations to perform path traversal. Evaluating a shortest-path query typically involves issuing one join chain per tested path length l, terminating when a match is found or the maximum length L is reached. However, relational joins suffer from intermediate result explosion due to Cartesian product semantics. At each join step, the engine computes all combinations of matching documents between consecutive indices, resulting in exponential growth in the number of partial paths. Specifically, for a query of length l, intermediate results at step i can reach $O(b^i)$, where b is the average out-degree [53]. Although many of these paths are ultimately discarded, they are nonetheless materialized and carried forward, incurring significant costs.

[65] address this explosion through cost-based query planning specifically for recursive property paths, while [60] propose factorized representations to efficiently enumerate join results, significantly reducing redundancy. [40] introduce optimization techniques to prune redundant paths early and strategically reuse intermediate results in navigational graph queries. [2] further establishes algebraic foundations, highlighting theoretical limitations of unrestricted recursive queries. It is important to note that while worst-case optimal join algorithms, such as those by [41], tightly bound intermediate result sizes in conjunctive pattern matching, they do not inherently address recursive or iterative path enumeration challenges.

Graph-oriented relational frameworks such as [16, 46, 68] adopt a hybrid approach by implementing graph operators within relational database systems, leveraging existing SQL engines and optimizers. Although this approach enables BFS-style graph expansions using database primitives, it does not alleviate the fundamental issue of exponential path enumeration. PathEnum [58] constructs a lightweight query-time index based on vertex distances to prune invalid edges before enumeration, significantly outperforming previous methods by reducing the search space, yet still potentially materializing a substantial number of invalid partial results.

Partial-Path Explosion Both classes of approaches face the same fundamental challenge: they materialize a large set of intermediate path candidates, only to later discard many of them after determining they do not meet final criteria. This materialization overhead becomes prohibitive in large-scale graphs, motivating our exploration of semi-join decomposition as an alternative approach. These specific limitations directly motivate our semi-join decomposition technique introduced in Sec.6.

4. Bridging the Document, Relational, and Graph Models

Knowledge graphs are powerful constructs for representing real-world entities and their relations via vertices and edges. Fig. 1 shows a graph with three people (Alice, Bob, and Charlie), where Alice and Bob are relatives. Each person owns a cellphone contract associated with a phone number. On a certain date, Alice's phone called Bob for about 10

minutes. On another occasion, Bob's phone texted Charlie about a sport match. These entities are represented as vertices in the graph, while edges depict the relationships between them (e.g., "being relatives", "making a call", etc.).

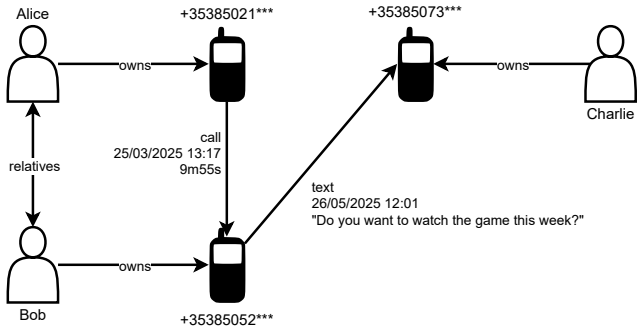


Fig. 1. An example graph

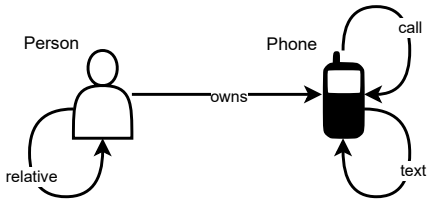


Fig. 2. The related ontology

More formally, a knowledge graph can be seen as a directed graph $G(V, E)$ with $V = \{v_1, \ldots, v_{|V|}\}$ being the set of nodes or vertices (representing entities), and E being the set of edges (representing relations). An edge $e_{vw} \in E$ is a directed link from a source node v to a target node w. The types of entities and relation that can occur in a knowledge graph are described by its *ontology*. For example, Fig. 2 shows that our graph can contain two types of entities – people and phones. In terms of relations, the ontology shows that people can be relatives to each other and can own phones, which can call and text.

In a document-oriented store, the most common approach for modeling a knowledge graph is to represent its vertices as documents [5]. A graph model is mapped to one or more document index D, where vertices are mapped to documents stored within those indices. More specifically, each document $d \in D$ may be mapped to a vertex $v \in V$. For instance, Alice, Bob, and Charlie are mapped to documents from a People index, while their phones are mapped to documents from a Phones index (see Fig. 3). Abusing the

notation, we can interchangeably refer to the vertex v as the document d that represents it, and vice versa.

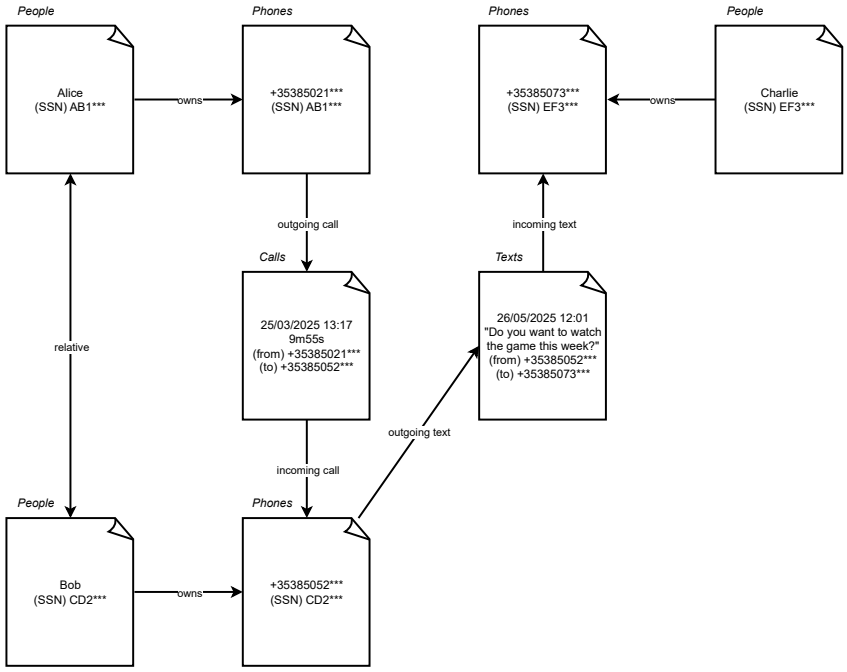


Fig. 3. The example graph in a document-oriented store

Different approaches exist for modeling edges. Simple edges without attributes can be derived dynamically at query time: an edge e_{vw} from vertex v to vertex w exists if values from a specific field of the document v share common elements with values from another specific field of the document w. For instance, we know that Alice owns a certain phone number because the document representing the phone contract includes the same social security number as the document representing Alice (see Fig. 3). More formally, for two indices A and B, consider fields s and t in their respective documents. We denote the values of field s in a document $v \in A$ as $v[s]$, and similarly $w[t]$ denotes the values of field t in document $w \in B$. An edge e_{vw} exists if and only if there is an intersection in the values of these fields, i.e., $v[s] = w[t]$, implying at least one common value. [6]

Beyond this basic graph model, Siren Federate can also accommodate property graph models, where edges themselves contain properties and are treated as first-class concepts [6]. In a property graph model, an edge is not merely a connection between vertices but a

[6] Under this mapping, edges can be defined through various conditions, not limited to simple equality. The graph model can accommodate more complex conditions, including range-based conditions, multi-dimensional conditions, pattern matching, etc. For clarity, we primarily use equalities in our examples.

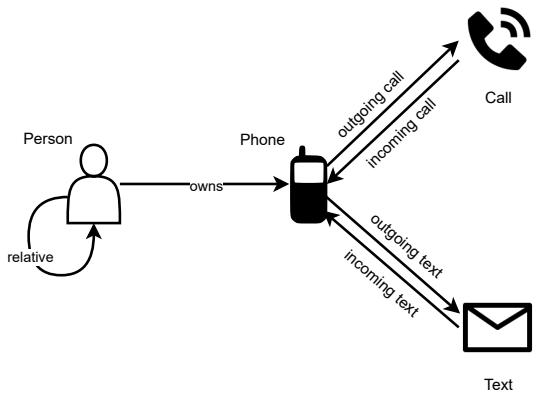


Fig. 4. The ontology after reifications

record with its own attributes (e.g., weight, timestamp, type). To represent such edges in a document-oriented store, we use a dedicated index E for each relation type, where each document $e \in E$ corresponds to an edge with properties. Alongside its own attribute, this edge document contains fields that reference both the source and target vertices. For instance, we may index a document with Alice's and Bob's phone numbers, a date, and a time duration to represent the call between the two (see Fig. 3). This approach is reflected in the ontology by the introduction of an additional entity in place of the original relation, and associating the new entity with the original ones through new relations, in a process similar to reification [36]. For instance, the "call relation" could be replaced by a "call entity" connected to the phones via new "outgoing call" and "incoming call" relations as shown in Fig. 4.

IR systems provide flexible data modeling and advanced search capabilities that enable searching the content of documents that encode a graph. However, they lack the relational join operations necessary for exploratory graph analysis. For instance, suppose we want to find all vertices adjacent to the vertex associated with phone number "+353 85052***". From the ontology, we know that we must join the documents for the "person", "call", and "text" entities with the document associated with that phone number – an operation not supported by traditional IR systems. Siren Federate bridges this gap by allowing joins within the document-oriented model, thus supporting the analysis of knowledge graphs.

Several works attempt to bridge document-oriented and relational models by mapping the first into the second [10, 24, 52, 61]. However, these approaches are limited to what relational databases propose and miss optimizations offered by IR systems. Siren Federate takes the opposite approach, mapping from a relational data model to a document-centric data model to fully leverage what IR systems offer: in the relational model, a join combines rows from multiple tables into a new table; in the document-oriented model, queries are applied to documents in an index and return matching documents.

Siren Federate expresses a join operation $\bowtie$ within this model as the process of finding documents from an index (the parent set) that are related to documents from another

index (the child set) according to specific conditions. Siren Federate implements (a) the *semi-join* $\ltimes$ for filtering the parent set's documents based on the child set's documents; and (b) the *inner-join* $\bowtie$ for extending the parent set's documents with fields from the matching child set's documents. [7] These operators forms the foundation for implementing graph operations over our document-oriented distributed system. Specifically, an inner-join $A \bowtie_{v[s]=w[t]} B$ explicitly enumerates edges e_{vw} by identifying pairs of vertices connected according to join condition equality. A semi-join $A \ltimes_{v[s]=w[t]} B$ efficiently determines which vertices $v \in A$ have outgoing edges towards vertices $w \in B$ by some edge e_{vw} without enumerating all vertex pairs explicitly. Abusing the notation, in the following we will just write $A \bowtie B$ (resp. $A \ltimes B$) when the join keys are clear from the context.

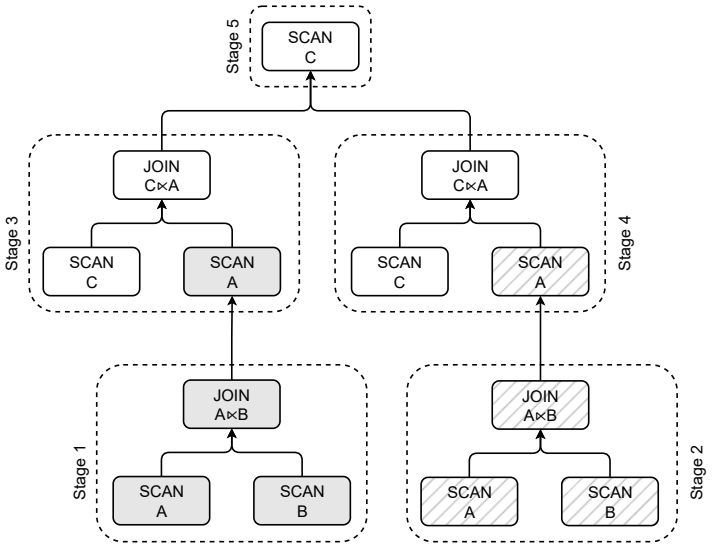


Fig. 5. A staged logical query plan

The logical steps for Siren Federate to join two document sets from indices A and B are highlighted in gray in Fig. 5. Two of the steps involve a SCAN operation, searching over the parent set A and child set B to retrieve subsets of documents to be joined. These documents may need to be exchanged across the computing cluster according to one of the different strategies described in Sec. 5.3. The parent and child subsets are then locally joined on the cluster's nodes by a JOIN operation, using data structures like hash tables, inverted indices, or k-d trees. The join results are tuples (in the relational sense) representing documents from the parent index that have fulfilled the join conditions. These tuples are then used by another SCAN operation to filter the parent index, retrieving the parent documents that meet the join conditions. This model also supports multi-join operations, with multiple child sets joined with the parent set. This is represented using a non-binary

[7] The Siren Federate domain-specific language for joins is documented at `https://docs.siren.io/siren-federate-user-guide/37/siren-federate/query-dsl.html`

tree structure, where each SCAN operation can be associated with more than one child JOIN operation (see Fig. 5).

Siren Federate follows a *late materialization* approach, scanning only fields from parent documents to evaluate a join operation and avoid manipulating entire documents. Each document is associated with a global ID (see Sec. 5.1), to uniquely identify it across the system. Tuples produced by the join operation include this ID, rather than the entire document content. Upstream operations can use this ID to materialize necessary fields. This strategy is used for various operations, such as filtering, sorting, aggregating, and retrieving document content. These operations are delegated to the underlying Elasticsearch engine, which is optimized for handling such tasks efficiently.

Given the document-centric model, tuples produced by the join must be grouped and sorted by the global document ID, as the join may produce scattered tuples about the same parent document, for example in the case of many-to-many relationships. This enables the parent SCAN operation to efficiently merge the join output based on ordered document IDs, which align with the natural order of the underlying log-structured storage (see Sec. 5.1). We employ efficient exchange strategies for optimizing the grouping and sorting operations (see Sec. 5.3).

Siren Federate uses this logical model to integrate relational joins into the document-oriented model. This representation drives the architecture design and runtime behavior of Siren Federate. For instance, the adaptive query planner uses it to stage the query plan execution. The semantic information embedded within this model is used by the semantic caching, but also for folding the query plan. Finally, this model retains the search engine's capabilities to efficiently execute filters, sorting, and aggregations.

5. Siren Federate Architecture

This section introduces the core architectural components of Siren Federate, shown in Fig. 6. Siren Federate acts as the compute layer of an investigative system, leveraging the distributed computing and storage architecture of Elasticsearch for scalability. The application layer of the investigative system relies on Siren Federate's relational and analytical capabilities via its search API.

The distributed IR system consists of a cluster of computing nodes. Each node plays a different role in the cluster: coordinator nodes are responsible for planning the execution of a request received from the search API, while data nodes are responsible for storing data and executing operations dictated by the coordinator's query plan. This architecture ensures sub-second to seconds response time at scale, as computational load is distributed across data nodes, which can independently process the log-structured data storage [43] to produce results (Sec. 5.1). Data nodes execute scan and join operations using a columnar data processing model (Sec. 5.2) and different join algorithms (Sec. 5.3). The query plan defined by the coordinator is divided into multiple stages (Sec. 5.4). Redundant operations of the query plan are folded to avoid unnecessary computation (Sec. 5.5). The logical query plan is processed iteratively, stage-by-stage, interleaving its physical planning with its execution. At each iteration, a cost-based query optimizer checks the semantic cache to reuse existing join results (Sec. 5.6) or selects the most efficient join algorithm.

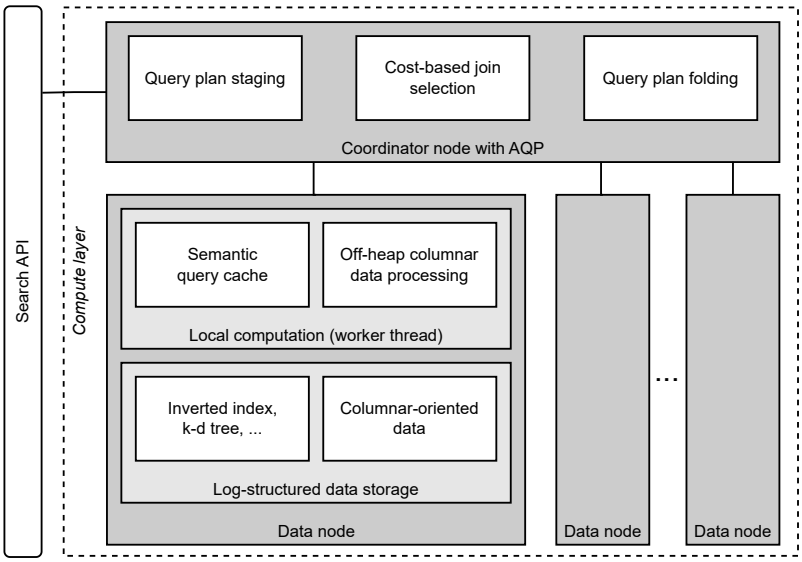


Fig. 6. Siren Federate's architecture

5.1. Log-Structured Distributed Data Store

Siren Federate leverages Elasticsearch's distributed data store, which horizontally partitions data across nodes using document sharding. An index is partitioned into shards, and each document is routed to a shard. A shard is a Lucene index [19], based on a log-structured model [43], and composed of one or more index segments. The log-structured model adopts an append-only update strategy and consists in creating a file-based data structure called index segment. Segments are immutable and get merged over time or when a size threshold is reached. This append-only model allows for (1) implementing a lightweight read-lock mechanism to guarantee data consistency during the execution of distributed joins, supporting concurrent query execution with real-time data updates; and (2) dynamically generating a global document IDs by combining shard and segment IDs with the document's insertion order, thanks to the immutability of segments. This global ID enables the quick location of a document's physical position in the cluster, and late materialization as explained in Sec. 4.

5.2. Columnar In-Memory Processing

Siren Federate stores data for intermediate join computation into off-heap main memory[8] using a columnar layout and leverages compression algorithms optimized for specific data

[8] Off-heap memory is allocated outside the heap space managed by the Java Virtual Machine (JVM). The JVM uses the heap space to dynamically allocate memory for Java objects at runtime. This memory is periodically reclaimed by the JVM's garbage collector (GC), and these GC cycles can delay application threads, introducing latency. By using off-heap memory for critical, potentially large data structures, an application can reduce GC-related delays and achieve better performance, at the cost of requiring manual memory management.

types. During join operations, only necessary fields, such as join key fields and global document IDs, are processed. The data exhibits a tabular structure, with tuples corresponding to documents and columns to their fields. There are two approaches for processing tabular data: row-at-a-time and column-at-a-time.

The row-at-a-time approach reads whole tuples even if only a few columns are needed, leading to CPU cache misses and negatively impacting the performance. Following best practices from [26], Siren Federate adopted the column-at-a-time approach, improving the query performance by a factor of 2 compared to the row-at-a-time implementation.[9]

The column-at-a-time approach uses a batch-processing pipeline. Each batch stores a fixed number of tuples, stored in a columnar fashion. The size of a batch is optimized to fit within the CPU cache line to avoid cache misses. A profiling tool[10] showed an increase of the CPU cache usage with the column-at-a-time approach: Siren Federate ver. 27.5 increased "cache-references" by 25% compared to ver. 22.6 that uses the row-at-a-time approach, demonstrating enhanced CPU cache utilization.

5.3. Distributed Join Algorithms

Siren Federate implements join techniques that leverage the intrinsic data structures of the underlying IR system to ensure scalability and high performance. An example with two distributed indices *A* and *B* is shown in Fig. 7. Both indices are partitioned into three shards, whose data needs to be exchanged across the computing nodes in order to be joined. The available join strategies are

Broadcast Hash Join Data from the child index is forwarded to all computing nodes hosting shards of the parent index (see Fig. 7, left). Local hash tables, created from the received data, are probed while scanning the parent index's columnar storage. Worker threads process segments in parallel, and local hash tables are shared across these threads.
Broadcast Index Join This strategy utilizes Lucene's inverted indexes (akin to burst tries [20]) for binary values, and Bkd-trees [50] for numerical values. Data are exchanged like the broadcast hash join (see again Fig. 7, left), but the child set data is used for index lookups over the parent set, eliminating exhaustive scans of the columnar storage. Worker threads process segments and probe the index with the received data. This is effective for graph expansion or path finding tasks, where the objective is to incrementally expand relationships from a group of records. Empirical assessments in Sec. 7 show competitive performance when joining thousands of records with larger relations (in the billions).
Partitioned Hash Join Inspired by [56], it leverages the columnar storage to scan data from the parent and child indices, partitioning data across computing nodes, and creating localized hash tables for each partition (see Fig. 7, middle). This method employs morsel-driven parallelism and involves a two-step partitioning to create fixed-sized work units: an initial node partitioning at the scan level (sender side) and a second partitioning at the join level (receiver side). This method achieves better parallelism and reduced memory and network overheads compared to strategies like the broadcast hash join. In Fig. 7 only three computing nodes are shown for the sake of space. However, this join strategy can leverage all cluster nodes regardless the number of shards. Empirical evaluations in Sec. 7 indicates horizontal (relative to the number of data nodes) scalability.

[9] https://info.siren.io/content/siren-benchmark-whitepaper
[10] https://github.com/async-profiler/async-profiler

Routing Join Similar to the broadcast hash join, it leverages the document sharding to reduce network traffic. It reuses the sharding routing function of the parent index to partition and exchange the child set's tuples to the corresponding parent set's shards [9] (see Fig. 7, right). Each worker thread employs either a hash table-based strategy (like the broadcast hash join) or an inverted index-based strategy (like the broadcast index join) to compute the results. Preliminary experiments (not presented in this work) indicate a 30% reduction in response times compared to the broadcast hash join strategy.

These strategies optimize specific scenarios. The role of the query planner (Sec 5.4) is to select the most cost-effective join strategy by considering factors such as shard topology and set cardinality to optimize the cluster's utilization.

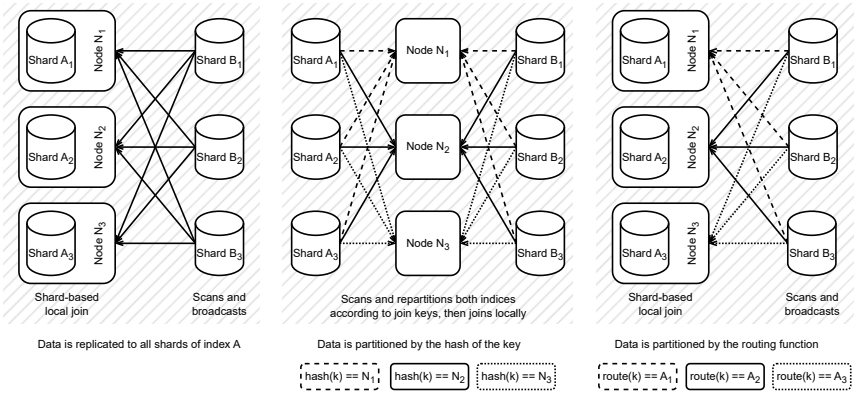


Fig. 7. Federate's distributed join algorithms: (left) Broadcast Hash/Index Join, (middle) Partitioned Hash Join, (right) Routing Join. Arrows represent data exchange between computing nodes

5.4. Adaptive Query Planner

Accurate join cardinality estimation is crucial for planning the most effective join algorithm. This is challenging with complex query plans with deeply nested joins, common in investigative scenarios. Traditional static methods, based on histograms and cardinality estimation formulas, often yield inaccurate estimations due to assumptions like attribute independence and distribution uniformity, resulting in sub-optimal selection of join algorithms. These inaccuracies are exacerbated as the complexity of the query plan increases [30]. Index-based join sampling, while more accurate, is computationally expensive, especially in distributed systems where it requires data shuffling across the network, and also suffers from inaccuracies with long sequence of joins.

To address this, Siren Federate implements an adaptive query planner (AQP) that interleaves planning and execution via stages [14]. This approach collects runtime statistics during execution, allowing more accurate cardinality estimation compared to static meth-

ods, especially for long sequence of joins. This enables to dynamically adjust the query plan based on real-time feedback. AQP operates in several key phases:

Logical Plan Generation The planner generates a logical query plan divided into stages. Each corresponds to a materialization point where an intermediate result is fully created before proceeding further. Typically, it comprises a logical join and two logical scans.
Physical Optimization The planner gathers statistical information, computes costs for various join strategies, and selects the optimal one. This is repeated for each stage, leveraging runtime cardinality estimates from already computed nested joins (stages).
Execution The physical sub-graph of each stage is executed, materializing intermediate results before proceeding.
Parallelization The query plan enables parallel execution of independent stages. Independent stages are executed concurrently, while dependent stages must wait for predecessors to complete.

To illustrate how AQP works, consider a dataset with three indices. An index A of documents representing cellphones, with fields containing the phone number, the operator, reference to the person who subscribed the contract, etc. An index B of documents representing the online activity of a person such as social media posts, with fields like person's identity and textual content. An index C of documents representing each call detail record (CDR) with fields such as time, duration, completion status, source and destination numbers of a call [55]. Imagine we want to find all CDRs related to phones used by people involved in suspicious online activities. AQP would generate a logical query plan in stages (logical plan generation) as shown in Fig. 5. Assume filters (e.g., keyword matching or vector search) applied to B identifies crime-related posts, then the set of "phones used by people involved in suspicious online activities" is the result of $A \ltimes B$ (Stages 1 and 2). The set of CDRs where these phones are the caller is returned by $C \ltimes A$ using the CDR's caller as the join key (Stage 3). Similarly, the set of CDRs where these phones are the callee is returned by $C \ltimes A$ using the CDR's callee as the join key (Stage 4). The disjunction of these two sets produces the desired results (Stage 5). Different join strategies may be used depending on the statistical information gathered from the previous stages (physical optimization and execution). Since Stages 1 and 2 are independent, they can be executed in parallel before moving to Stages 3 and 4 (parallelization).

5.5. Query Plan Folding

User queries often contain redundant operations that negatively impact query processing, such as repeated searches or joins. Redundancies commonly occur when investigating related entities through various graph topologies, boolean expressions, or batched requests targeting the same entities with diverse filters or aggregations. Redundant operations affect also SQL query processing [34, 54].

To address this challenge, Siren Federate adopts a query plan folding that uses the semantic definition of query operators to detect and merge redundant operators across one or more logical query plans. The semantic definition of an operator captures its logical meaning, structure, dependencies, and state of the data tables it involves [13].

Siren Federate handles not only the folding of selection and scan operations [54], but also of join operations. The folding strategy consolidates redundant operations into a unified shared operator. In the previous AQP example, the operators from Stage 1 and its

subsequent SCAN A are folded with those from Stage 2 and SCAN A (highlighted by a hatched background in Fig. 5) since they represent the same join. However, Stage 3 and 4 are not folded as they have different join conditions: the CDR's caller number is used in Stage 3, while the CDR's callee number is used in Stage 4. Similarly, SCAN C is not folded between the two stages because it scans different fields.

5.6. Semantic Caching

In exploratory graph analysis, iterative analysis often results in recurrent execution of the same join operations. By caching these, the system can optimize subsequent related queries, reducing the latency and computational load.

Semi-joins are well-suited for caching compared to other join types, as their outputs can be represented as sets of document IDs. Exploiting this, Siren Federate employs semantic caching, relying on the semantic definition of query operators (Sec. 5.5). Compared to conventional caching methods which operate at the query syntactic level [45], semantic caching [13] indexes cache entries according to query operator semantics, guarantying data consistency and resilience to changes in the underlying data, even when query operators depend on multiple data sources derived from descendant query operators. Unlike traditional caching strategies that focus on raw results which can lead to large memory overhead, the semantic caching strategy uses compact bitset representations to efficiently encode semi-join outputs. This reduces memory consumption and increases the potential amount of cached operations, enhancing the overall system efficiency.

To illustrate this, take the example from Sec. 5.4. In a subsequent iteration, the investigator wants to identify, from a new index D, people owning phones involved in previously found CDRs. This adds a new join $D \ltimes C$, with C being the results of the previous iteration. The AQP generates a logical plan including the subtree from Fig. 5. With semantic caching, the results of the subtree can be reused, meaning that only the additional join $D \ltimes C$ must be computed.

This strategy benefits graph analytics, particularly path finding algorithms which can be represented as sequences of semi-joins. Semantic caching of semi-joins reduces redundant computations, minimizing the number of operations and associated I/O, and resulting in a more efficient execution of graph queries (see Sec. 6).

5.7. Principles for a competent graph analytics system

In [64], ten Wolde et al. identify eight core features necessary for a competent graph analytics system. Siren Federate leverages and extends the capabilities of Elasticsearch to meet these principles:

Fast Scans on Elements with Schema Siren Federate uses Elasticsearch's dynamic schema capabilities and column-oriented storage for fast attribute scanning. Dynamic schema-awareness offers flexibility in handling knowledge graph variability and optimizes query processing by adapting to the data structure.

Skippable Compressed Columnar Storage Elasticsearch supports fast columnar scans with data skipping via pushed-down predicates, combining columnar storage ordered by document IDs with an inverted index or k-d trees.

Vectorized or Data-Centric Execution Siren Federate adopts column-at-a-time (vectorized) processing for its data pipelines during scan and join operations, and an in-memory vector format with compression and data skipping.
Morsel-Driven Multi-Core Siren Federate uses morsel-driven parallelism during scan and join operations to distribute constant-sized work units (morsels) across worker threads, reducing load imbalance and optimizing CPU cache usage. Dynamic index segment splitting is critical for better parallelism during scans of large segments, and reducing query execution latency.
State-of-the-Art Query Optimization Siren Federate's AQP splits query plans into stages and employs runtime estimation to avoid the overhead of traditional table sampling, ensuring efficient query execution.
Bulk APIs/Algebras Elasticsearch's boolean algebra operates as a bulk API for predicate evaluation by enabling manipulation of document sets efficiently. Siren Federate integrates relational algebra that also functions as a bulk API.
Out-of-Core Buffer Manager While performing in-memory join operations, Siren Federate leverages Elasticsearch's ability to handle out-of-core data sizes efficiently during scan. Frequently accessed columns are cached at the operating system level, ensuring scan operations in RAM.
Explicit Control over Memory Locality Siren Federate uses off-heap memory management to reduce garbage collection overhead when handling gigabytes of data in memory for short durations, and optimizes memory locality through columnar storage, morsel-driven parallelism, and effective data partitioning.

To further enhance performance in a distributed environment, Siren Federate adopts three additional core features:

Data Locality Leveraging data locality minimizes data movement across the network, a common bottleneck as it requires additional intermediate serialization and copy of the data. Siren Federate performs late materialization of documents using global IDs to quickly locate documents in the cluster and co-locates data by reusing existing data routing coming from document sharding.
Data Exchange Effective data exchange operators exploit the physical distribution and structure of the storage to maximize memory locality without data reorganization. Preserving the implicit order, even partially, of materialized tuples during scan and join operations improves the performance of the exchange operator, which must group and sort tuples based on the document ID as explained in Sec. 4. Radix partitioning [67] is a highly efficient method for clustering tuples from a range of documents together, improving document sorting.
Caching Implementing compact caching strategies for intermediate results allows reuse across queries and users, reducing redundant computations and improving overall system efficiency, especially in incremental exploratory scenarios.

6. Semi-Join Decomposition for Path Queries

Investigative intelligence workflows frequently require querying complex chains of relationships in large-scale knowledge graphs. As discussed in Sec. 2.2, supporting these workflows requires scalable mechanisms for path-finding queries and traversals. Analysts

must answer fundamental questions such as "Are these two suspects indirectly connected?" or "Which entities connect them?", often traversing long graph paths. In such scenarios, analysts must inspect the actual paths – not just to check reachability or compute distances, but to inspect each intermediate entity or event along the way. Path enumeration therefore emerges as a central operation in investigative workflows, yet it poses significant scalability challenges in distributed systems.

The enumeration of all shortest paths between two entities – known as the single-pair all-shortest-paths problem – is computationally challenging due to the rapid combinatorial expansion of potential solutions. Traditional methods struggle with this complexity, as intermediate results generated during enumeration (such as expanding join tables or maintaining traversal predecessors) grow rapidly and can quickly exhaust memory resources.

This combinatorial explosion is a recurring limitation in existing path enumeration techniques, as detailed in Sec. 3.2. To address this, we introduce Semi-Join Decomposition (SJD), a technique designed for scalable, memory-efficient path enumeration.[11] SJD decomposes deep path queries into a sequence of semi-joins that iteratively prune unreachable candidates, significantly reducing intermediate result sizes. Although SJD applies broadly to general path queries, we illustrate its benefits through the all-shortest-paths use case, which shows the practical and computational challenges that SJD addresses.

In the following subsections, we formalize the reachability and path query problems (Sec. 6.1), present the SJD method in detail (Sec. 6.2), and describe its integration into Siren Federate's distributed architecture (Sec. 6.4).

6.1. Path Queries and the Path Enumeration Challenge

At the core of graph analysis lies the path enumeration problem – identifying and retrieving all paths that meet specified criteria. This becomes particularly challenging as the number of potential paths between nodes can grow exponentially with graph size and connectivity. This exponential growth of intermediate results is a fundamental challenge that traditional approaches struggle to address, as discussed in Sec.3.2.

Reachability and shortest-path problems highlight these enumeration challenges. In graph theory, the reachability problem consists in determining whether a node $v \in V$ can be reached from another node $u \in V$, i.e., if there exists at least one path p of finite length between the two nodes. A path p is an ordered list of edges $[e_{uw_1}, e_{w_1w_2}, \ldots, e_{w_nv}]$. Equivalently, a path p can be represented as an ordered list of nodes $[u, w_1, w_2, \ldots, w_n, v]$ where $e_{w_iw_j} \in E$ if nodes w_i and w_j appear consecutively in the list (i.e., $j = i + 1$).

The single-pair shortest path problem extends reachability by finding all paths between nodes $u, v \in V$ having the minimum possible length in terms of edge count. This corresponds to the "ALL SHORTEST" semantics in modern graph query languages such as GQL [15]. Solutions to the shortest path problem also solve reachability, as any shortest path confirms that one node is reachable from the other.

For practical efficiency, path queries are often bounded with a maximum length L. This constrains the reachability problem to test if v can be reached from u following a path with at most L edges, and limits the shortest path problem to paths whose length is at most L. However, even with these constraints, the computational complexity remains challenging in large graphs as it grows with the number of vertices and edges [11].

[11] The SJD technique was first described and patented in [48].

The difficulty of path enumeration becomes apparent when we consider the growth pattern of potential paths. Let b denote the average branching factor (out-degree) of nodes in a graph. For each additional hop in a path query, the number of potential paths from a node multiplies by approximately b. Thus, for a path of length L, there could be up to $O(b^L)$ distinct paths to consider – an exponential growth that quickly overwhelms naive approaches [53]. While determining the mere existence of a path or computing a single shortest path can be done efficiently, enumerating all paths meeting specific criteria becomes exponentially more difficult as path length increases.

Beyond shortest paths, many path queries involve bounded path enumeration with constraints. These might include finding all paths between nodes that are at most L hops long, or paths that satisfy specific conditions on intermediate nodes or edges (e.g., "only through legal entities in a given country"). Even with such constraints, the potential number of paths to enumerate remains exponentially large in densely connected graphs.

6.2. Semi-Join Decomposition

To overcome the limitations of traditional BFS and join-based approaches, we introduce a technique called Semi-Join Decomposition (SJD). This query planning technique avoids intermediate path materialization by leveraging semi-joins, decomposing complex chains of inner-join into multiple simpler queries, significantly mitigating the exponential explosion of intermediate data during multi-hop graph traversal.

The central idea of SJD is that many intermediate paths generated by traditional methods are ultimately discarded – not because they do not match join conditions at each hop, but because they fail to reach the target and do not lead to complete paths. Rather than fully constructing these paths only to prune them later, SJD incrementally eliminates candidates using a series of semi-joins by propagating reachability constraints.

Core Idea The essence of semi-join decomposition lies in replacing a chain of inner-joins used for a single-pair shortest path query with multiple queries, each structured as sequences of semi-joins. Consider the original inner-join chain for a path of length l, i.e., $D_1 \bowtie D_2 \bowtie \cdots \bowtie D_{l+1}$. This is decomposed into $l+1$ queries q_k (for $k = 1, \ldots, l+1$), each consisting of a sequence of semi-joins and targeting a specific "layer" of vertices along the desired path. Each decomposition query q_k returns a set of documents R_{q_k} that correspond to vertices appearing at position k in at least one path from the source u to the target v. In other words, each document $w \in R_{q_k}$ belongs to at least a path $p = [u = w_1, \ldots, v = w_{l+1}]$ and appears precisely at position k.

For example, given a path of length 3 across indices $\{D_1, D_2, D_3, D_4\}$, the inner-join sequence $D_1 \bowtie D_2 \bowtie D_3 \bowtie D_4$ can be decomposed into four semi-join queries: (1) $q_1 = D_1 \ltimes D_2 \ltimes D_3 \ltimes D_4$; (2) $q_2 = (D_2 \ltimes D_1) \wedge (D_2 \ltimes D_3 \ltimes D_4)$; (3) $q_3 = (D_3 \ltimes D_2 \ltimes D_1) \wedge (D_3 \ltimes D_4)$; (4) $q_4 = D_4 \ltimes D_3 \ltimes D_2 \ltimes D_1$.

For each tested path length l, the semi-join decompositions q_k can be executed sequentially or in parallel. However, one query is better used as an initial reachability test, and any decomposition query q_k can be used for this. For instance, the midpoint $q_{\lceil \frac{l+1}{2} \rceil}$ could be tested using a bisecting approach, and q_{l+1} can be used to exploit the cached results of the reachability test for $l - 1$. If this initial query returns an empty result set, there is no path of the given length, thus preventing unnecessary computation of subsequent queries. Otherwise, the remaining queries are executed and their result sets stored.

This decomposition aggressively prunes irrelevant paths early because semi-joins do not enumerate all possible combinations explicitly. Instead, they filter vertices by simply checking for the existence of matches, limiting intermediate results to a size linear in the number of vertices rather than exponential with respect to branching factors.

Path Materialization Once the relevant documents at each layer have been identified via semi-joins, full path enumeration proceeds via a guided depth-first search (DFS), constrained to documents in $\{R_{q_1}, \ldots, R_{q_{l+1}}\}$: (1) Starting from the source node, adjacent nodes are retrieved from the index using the appropriate semi-join result sets as lookup tables. (2) Iteratively, subsequent vertices at each path position are identified via in-memory hash tables keyed by relevant field values. Each R_{q_k} can be retrieved from the index and stored in a separate hash table with source fields as keys, enabling rapid lookups.

This materialization approach supports streaming delivery, allowing paths to be generated and consumed incrementally without requiring full materialization in memory. The approach also offers flexibility in how it can materialize results, supporting complete path tuples (capturing the entire path from source to destination), individual edges, or triples (similar to RDF's subject-predicate-object model).

Remark (Connection to BFS) Classic BFS-based approaches also organize nodes into "layers" but typically maintain a layered DAG of parent pointers. In our method, the sets $R_{q_1}, \ldots, R_{q_{l+1}}$ serve an analogous function, capturing exactly those vertices that appear at position k in some valid path.

6.3. Complexity and Caching Benefits

The size complexity of each semi-join result set R_{q_k} is bounded linearly by the number of vertices $|V|$, rather than growing exponentially with the branching factor. Consequently, the combined complexity of all decomposed results remains linear with respect to $|V|$, a significant improvement over traditional sequence of joins.

However, SJD requires $l + 1$ queries each involving l semi-joins, i.e., $O(l^2)$ total operations versus $O(l)$ inner-joins in a naive chain. In practice, SJD remains favorable because each semi-join is significantly cheaper than a full inner join: (1) it produces a smaller result set and exchanges less data over the network; (2) it operates faster due to smaller memory footprints and optimized local join algorithms; (3) it is efficiently cached using compact bitsets, as detailed in Sec. 5.6, and can be reused by subsequent decompositions.

Caching dramatically reduces repeated semi-join computations, as consecutive decompositions often share numerous intermediate results, allowing significant reuse of previously computed results.

Theorem 1 *For each subsequent decomposition q_i, all semi-join operations except one are identical to those performed by the previous decomposition q_{i-1}, benefiting directly from cached results.*

Proof. By induction: (Base case) Let $q_1 = (D_1 \ltimes \cdots \ltimes D_{l+1})$ *and* $q_2 = (D_2 \ltimes D_1) \wedge (D_2 \ltimes \cdots \ltimes D_{l+1})$. $(D_2 \ltimes \cdots \ltimes D_{l+1})$ *from* q_2 *is a sub-sequence of* $(D_1 \ltimes \cdots \ltimes D_{l+1})$ *from* q_1*, hence it can be processed by reusing the cached results. Only* $(D_2 \ltimes D_1)$ *must be computed. (Step case) Let* $q_i = (D_i \ltimes \cdots \ltimes D_1) \wedge (D_i \ltimes \cdots \ltimes D_{l+1})$ *and* $q_{i+1} = (D_{i+1} \ltimes \cdots \ltimes D_1) \wedge (D_{i+1} \ltimes \cdots \ltimes D_{l+1})$. $(D_{i+1} \ltimes \cdots \ltimes D_{l+1})$ *from* q_{i+1} *is a sub-sequence*

of $(D_i \ltimes \cdots \ltimes D_{l+1})$ *from* q_i*, therefore it can be processed by reusing the cached results.* $(D_{i+1} \ltimes \cdots \ltimes D_1)$ *from* q_{i+1} *contains sub-sequence* $(D_i \ltimes \cdots \ltimes D_1)$ *from* q_i*, therefore only the outermost join from* $(D_{i+1} \ltimes (D_i \ltimes \cdots \ltimes D_{l+1}))$ *must be computed while the rest can be processed by reusing the cached results.*

For instance, in the example decomposition $q_1 = D_1 \ltimes D_2 \ltimes D_3 \ltimes D_4$, computation $D_2 \ltimes D_3 \ltimes D_4$ is reused by subsequent query q_2 without additional processing. Even though we conceptually have $O(l^2)$ semi-joins, most are skipped by cache hits, significantly reducing the computational overhead. After caching, at most $2l$ semi-join operations need execution for a single path of fixed length l, thus achieving linear complexity.

When testing all path lengths from 1 up to L to find the minimal length at which the target is reachable, the difference between naive chains of joins and SJD is even more pronounced. The chains of joins will execute l inner-joins for each length $l = 1, 2, \ldots, L$ which amounts to a total of $O(L^2)$. In comparison, SJD will execute one semi-join query with l semi-joins for each length l, reusing cached results almost entirely, until the reachability test is positive. Empirically, the reachability test for each new length often requires computing at most two semi-joins, while reusing the cached result sets of the other semi joins that were computed at the previous iterations. Therefore, the number of semi joins to perform the reachability tests until a path length of l is found is $2 \cdot (l - 1)$ semi-joins. Once the reachability test is positive, the $(l - 1)$ remaining semi-join queries are executed by computing only one semi-join operation (reusing the cached results for the others) for each query. This leads to an overall cost of about $3 \cdot (L - 1)$ semi-joins, i.e., linear in L. Although the exact break-even point varies, once L grows past 5 or 6, SJD plus caching can require fewer total joins than the naive chains of inner-joins method.

Overall, SJD's complexity in terms of semi-join operations scales linearly with the path length l once caching is factored in, while intermediate data remain linear in $|V|$. By separating "reachability pruning" (through semi-joins) from "path materialization" (through a final DFS), SJD provides a scalable strategy for iterative path queries that avoids prohibitive computational or memory overhead.

6.4. Integration with Siren Federate

The semi-join decomposition approach integrates seamlessly with Siren Federate's architecture and optimization strategies. This integration enhances Siren Federate's ability to handle complex graph queries efficiently by leveraging several key features:

Adaptive Query Planning (AQP) Siren Federate employs an adaptive query planner (Sec. 5.4) that iteratively optimizes query execution plans based on runtime statistics. Semi-join decomposition queries naturally fit this model, as each query step provides valuable intermediate statistical information used to choose the most efficient semi-join strategy (Sec. 5.3).

Late Materialization and Columnar Storage Consistent with Siren Federate's late materialization strategy (Sec. 5.1), semi-join decomposition primarily manipulates compact document identifiers rather than entire document contents. Combined with Siren Federate's columnar storage and vectorized data processing (Sec. 5.2), semi-join operations access only minimal field subsets required for join computations, significantly lowering I/O and memory overhead.

Semantic Caching The effectiveness of semi-join decomposition is substantially enhanced by Siren Federate's semantic caching mechanism (Sec. 5.6). Semi-join results are compactly represented as bitsets, which can be cached efficiently. Since multi-hop investigative queries typically involve iterative and incremental expansions, cached semi-join results dramatically reduce redundant computations across query iterations.

Join Exchange Operators and Data Locality Semi-join decomposition exploits Siren Federate's optimized data exchange operators (Sec. 5.3). These operators efficiently handle data transfers between cluster nodes, taking advantage of existing data locality and document-sharding mechanisms to minimize network traffic in highly distributed environments.

In Sec. 7, we evaluate semi-join decomposition on complex path queries, showing that it outperforms naive join approaches as data scale increases, while maintaining interactive response times suitable for investigative workflows.

7. Evaluation

In this section, we present an experimental evaluation of Siren Federate that focuses on two complementary aspects of the system: (1) scalability with large data volumes in a distributed environment; and (2) performance on complex graph querying patterns. Our evaluation methodology reflects the system's primary design goals – enabling interactive exploration of large-scale heterogeneous knowledge graphs with sub-second to second response times.

Comprehensive cross-system comparisons present practical challenges: configuring and optimizing multiple systems for fair comparison requires significant resources, particularly when considering commercial systems with potential benchmarking restrictions. We have therefore focused on providing an objective assessment of our system's performance characteristics and the impact of key architectural decisions. This approach aligns with our primary contribution of sharing practical architectural insights gained through long-term system development in an industrial setting.

Rather than comparing against multiple systems with different architectural foundations, we focus on Siren Federate's performance across critical dimensions aligned with investigative intelligence requirements: data scale (billions of documents), query complexity (from simple relation traversals to complex path finding), and concurrent usage patterns. This approach validates the effectiveness of the architectural choices and optimization techniques described in previous sections. Future work could explore more targeted comparative evaluations of specific components such as the adaptive query planner and semi-join decomposition under varying workloads and data distributions.

In Sec. 7.1, we evaluate the system's performance and scalability using a synthetic dataset representing cell phone location records – a common data source in investigative scenarios. In Sec. 7.2, we assess the system's graph querying capabilities using the LDBC Financial Benchmark (Finbench), analyzing how our adaptive query planning and semi-join decomposition techniques perform on standardized graph patterns. Finally, Sec. 7.3 draws insights from real-world deployments, demonstrating how these techniques translate into operational impact in production environments.

7.1. Large Scale Benchmark

This section demonstrates Siren Federate's scalability and performance under realistic investigative intelligence workloads. We evaluate the system's ability to process semi-joins efficiently across three dimensions that reflect real-world deployment scenarios: computing cluster size, concurrent user load, and data volume. We focus on semi-joins as they are more suitable than inner-joins for large datasets and form the foundation of exploratory graph analysis tasks like set-to-set navigation, graph expansion, and pathfinding. Our experimental design uses a synthetic dataset modeling cellular location tracking, representing a common scenario where analysts must correlate location data across time periods and geographic boundaries.

Dataset. We use a synthetic dataset[12] with 15.6 billion documents tracking the positional information of cell phones to mimic scenarios where analysts monitor phone calls. This dataset covers 100 days, with 156 million documents per day, 6.5 million unique phone identifiers, and 2,400 positions per phone. One Elasticsearch index per day is created, with 8 primary shards with no replicas. The total size of the data is 2.7 TB.

Setup. The benchmark tests use a varying number of machines with 16 CPUs, a local NVME drive for Elasticsearch data, a gp2 drive for the OS, and a 10 Gbps network link. Elasticsearch is configured with 30 GB heap memory, and Siren Federate with 16 GB off-heap memory.

Experiments. We evaluate Siren Federate with varying cluster sizes (12, 18, and 36 nodes), data volumes, and concurrency loads (1, 5, and 10 users). The system is setup to serve the maximum number of concurrent users. We use the following queries with different complexity: **Q1** joins phone numbers in a given area on one day with those in another area on another day (78 million documents per set); **Q2** is similar to Q1 but over a week (546 million documents per set); **Q3** , given a phone number, finds other phones at the same location over 90 days (14 billion documents filtered with 2,160 documents); **Q4** finds phones at the same location on two different days (156 million documents each); **Q5** is similar to Q4 but over a week (more than 1 billion documents per set); **Q6** is similar to Q4 but over two weeks (more than 2 billion documents per set).

Queries use the *partitioned hash join*, except Q3, which uses the *broadcast index join* due to the small cardinality of its child set. We measure the execution time for a randomly-selected query with a fixed number of concurrent users, bypassing query caches. The benchmark runs until at least 100 measurements per query are produced, reporting the 90th percentile processing time (P90).

Results. Fig. 8 reports P90 for queries Q1 to Q3. With one concurrent user, Q1 joins 156 million documents in subsecond time. P90 does not decrease as the number of nodes increases because the amount of data joined is too small. The latency of the join phase represents an insignificant part of the response time, while the scan phase is tied to a limited number of shards and cannot be further distributed.

Q2 joins over 1 billion documents (×7 more data than Q1), with P90 increasing by at most ×2.5, indicating a better usage of the computing resources. However, response time

[12] `https://gist.github.com/scampi/07e7bd556fe016a5cba6c092c3f418fb`

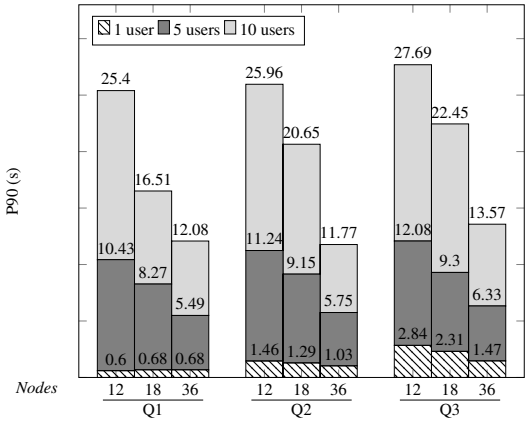


Fig. 8. Query times for queries Q1 to Q3 with varying number of nodes and users

does not decrease as expected with more nodes. In fact, increasing the number of nodes by ×3 only decreases P90 by 30%. This suggests that Q1 and Q2 latency is dominated by fixed overhead during the query planning and pipeline execution. This requires further analysis, and we will improve our monitoring solutions to assist this and similar investigations.

Q3, which filters 14 billion documents using the *broadcast index join* strategy, shows P90 decreasing by ×2 as nodes increases by ×3. This strategy avoids partitioning and shuffling such a large parent set while still distributing load across the full cluster as the number of nodes increases.

With 5 concurrent users, P90 of Q1 increases by ×12.5 – on average across cluster's configurations – when compared to response times with 1 user; Q2 increases by ×6.8; Q3 by ×4.2. With 10 concurrent users, the increase of P90 is by at most ×2.3 w.r.t. 5 users. Differently than moving from 1 to 5 concurrent users, doubling the number of users doubles the response times in this case. This significant increase in latency is particularly evident for Q1, and it may underline as previously noted an overhead in query planning and execution for short queries. Such overhead may impact scaling, and needs to be investigated in future work. Nonetheless, these results show that Siren Federate scales well with the number of users and nodes, achieving subsecond to second response times over large datasets.

For Q4, Q5, and Q6, the aim is to further assess the ability of the system to scale with the amount of data, joining parent and child sets containing hundreds of millions to billions of documents each. With 36 nodes, the reported P90 is 1.92, 5.00, and 7.58 seconds for Q4, Q5, and Q6 respectively. We can observe a sub-linear scaling factor with the size of the join operation. Between Q4 and Q5 the size of the join operation increases by ×7, but P90 only increases by 2.2 seconds. Between Q5 and Q6, the size of the join operation increases by ×2 but P90 only increases by 1.5 seconds. These results confirm that Siren Federate scales well with the amount of data.

7.2. LDBC Finbench

This section evaluates Siren Federate's graph querying capabilities using standardized, complex query patterns derived from the LDBC Financial Benchmark (Finbench). These experiments assess our query planning strategies and semi-join decomposition (SJD) technique on realistic graph traversal workloads.Using an industry-standard benchmark provides objective measurements of our system's ability to handle diverse graph query patterns – from multi-hop traversals to shortest path computations – that are representative of investigative workflows in financial compliance and fraud detection scenarios.

Our evaluation focuses on two key aspects of graph query performance: (1) comparing the efficiency of adaptive versus static query planning approaches across varying complexity levels and resource configurations; and (2) demonstrating the practical benefits of semi-join decomposition for path queries compared to traditional sequences of inner joins. For path queries, while theoretical comparisons to BFS-based approaches are provided in Sec.3.2, our experimental evaluation compares SJD against the sequence of inner joins approach implemented within our system. This comparison allows us to measure the performance improvements of our contribution while maintaining implementation feasibility within our research scope.

These measurements validate that our architectural decisions enable performant graph analytics on document-oriented data stores, bridging the gap between rich IR capabilities and complex graph traversals. The results provide clear evidence of SJD's practical benefits, while establishing a foundation for future comparative evaluations against additional graph processing techniques.

Dataset. We use the LDBC Financial Benchmark dataset [51], denoted as *Finbench*. This benchmark represents financial industry data and query patterns. Its schema consists of 5 entity and 13 edge types, e.g., persons and accounts along with money transfers. This benchmark[13] provides pre-generated datasets of different sizes; we used the dataset with a scale factor of 10, containing 138,305,500 triples which is in terms of documents indexed 5,376,981 entity instances and 26,633,151 relations.

Setup. Since the system's performance in a distributed environment has already been assessed in Sec. 7.1, this section focuses on evaluating the efficiency of local processing and optimization techniques for graph queries. Each entity and edge type from the dataset is indexed into its own index – as described with the reification approach in Section 4 – in a single-node Elasticsearch cluster with index replication disabled. The computing instance is equipped with 64GB of main memory, while we conduct tests with varying numbers of CPUs: 8, 16, 32, and 64.

Experiments. We use a subset of the *complex-read queries* from the Finbench *transaction workload*; these queries require matching graph patterns of varying complexity. To process them, we express the queries in GQL [15], an emerging ISO-standard declarative query language for property graphs that supports expressive pattern matching and enables complex graph-oriented queries in a concise, human-readable form. Siren Federate can interpret GQL queries and internally translate them into Federate query plans for processing. Noticeably, queries TCR1, TCR2, TCR7, TCR9, TCR11, and TCR12 require

[13] `https://ldbcouncil.org/benchmarks/finbench/`

retrieving some entities' attributes, whereas queries TCR3 and TCR5 involve computing certain paths (see Appendix A.1). Due to current system limitations, we were unable to process some other queries from the dataset. Specifically: (a) TCR4 requires support for cycles; (b) TCR6 requires support for conditions on the number of connected documents; (c) TCR8 requires the graph engine to handle complex patterns, such as edges defined as the disjunction of different labels or referencing values from other nodes in a WHERE clause; and (d) TCR10 requires computing the Jaccard similarity between two sets. These features are not currently available, but we plan to introduce them in future work.

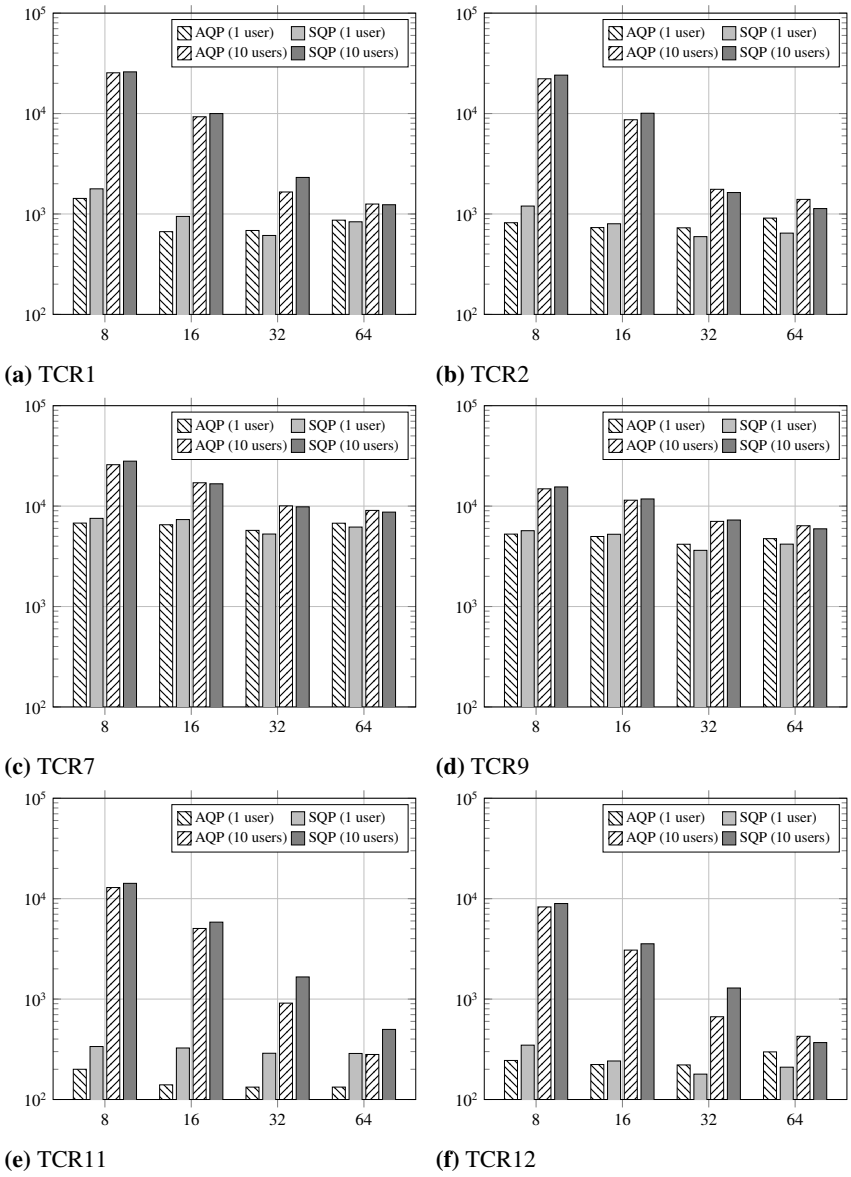
We evaluate Siren Federate by simulating a single user or 10 concurrent users issuing these queries. Additionally, we experiment with query planning using a static or adaptive planner to assess which approach is the most efficient. For each join in the query, the static query planner (SQP) selects the most appropriate join strategy prior to the query processing and leveraging simple simple statistics estimation. On the contrary, the adaptive query planner (AQP) interleaves query planning and processing to collect and leverage more accurate runtime statistics, as described in Sec. 5.4.

Results. The results for the *Finbench* benchmark are summarized by Fig. 9 and Fig. 10, while the raw results are reported in tabular format in Appendix A.2. Results show that Siren Federate processes complex graph queries in seconds, never exceeding the one-minute threshold, even with 10 concurrent users. In some cases, the system achieves sub-second response times (e.g., TCR1, TCR2, TCR11, and TCR12 with a single user). In the following, we first focus on queries TCR1, TCR2, TCR7, TCR9, TCR11, and TCR12 whose results are shown in Fig. 9.

All queries except TCR11 require selecting fields from multiple entities, necessitating a sequence of one or more inner joins. While we plan to introduce broadcast-based inner-join strategies in the near future, Siren Federate currently offers only the partitioned hash-join strategy for processing inner joins. This choice is justified by the fact that distributed inner joins often require exchanging a significant amount of data, making a partitioned exchange approach more suitable than broadcast-based ones in most cases. Consequently, AQP and SPQ exhibit similar performance on most queries, as they generate identical query plans.

TCR11 selects fields from a single entity, allowing its execution to be performed using only semi-joins, for which Siren Federate offers multiple join strategies. For this query, AQP achieves better response times than SPQ – 50% faster on average in the single-user scenario – as it takes into account statistics of intermediate join results to better select the appropriate join strategies. This optimization is particularly critical for TCR11, as it requires matching a graph pattern that can expand from a length of 1 up to 10, requiring a minimum of 4 joins and up to a maximum of 22 joins to be processed. When handling such a high number of joins, planning errors can easily accumulate, leading to significant latency. AQP mitigates this by leveraging finer-grained estimations, resulting in more efficient execution plans.

As the number of CPUs increases, we observe that query latency decreases for both AQP and SQP. However, in single-user experiments, SQP is often slightly faster than AQP due to its higher parallelization. In fact, SQP determines strategies for all joins upfront, without considering intermediate. This enables greater operation-level parallelism across the physical execution. To better understand, consider the query plan in Fig. 5. With SQP, the leftmost SCAN A in Stage 1 can execute concurrently with the SCAN C in Stage

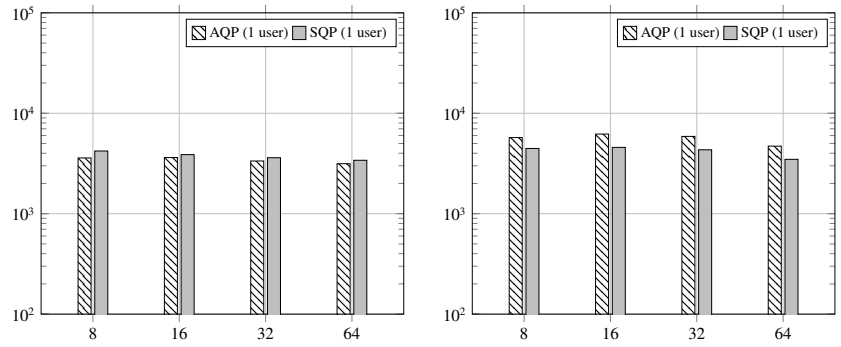


(a) TCR1 **(b)** TCR2

(c) TCR7 **(d)** TCR9

(e) TCR11 **(f)** TCR12

Fig. 9. Query times (P90) in ms for LDBC queries. The *y*-axis is in logarithmic scale



(a) TCR3 **(b)** TCR5

Fig. 10. P90 processing times (in ms) for queries TCR3 and TCR5, for a single users. The *y*-axis is in logarithmic scale

3, even though Stage 3 depends on the results of Stage 1. This is possible because SQP pre-schedules all operations, allowing scans to pre-fetch data before than the joins which consume their results are ready to execute. In contrast, our current AQP implementation processes stages sequentially. For instance, AQP must complete the entire Stage 1 (including the join and all associated scans) before initiating any operations in Stage 3 when executing the query in Fig. 5. The better parallelism provided by SQP can improve performance for single queries, though AQP more efficient join selection often compensates under concurrent workloads. In fact, the cost of the query plan remains similar (or better) with AQP. Consequently, under concurrent workload, the system can utilize underused resources across all running queries. In other words, while AQP may not saturate the CPU for a single query, under concurrent load it enjoys effectively the same (or better) throughput because the total workload is distributed more evenly. These findings suggest that further tuning of the adaptive planner – particularly a mechanism to increase or decrease task parallelism based on the current load – would improve the single-user performance of AQP without sacrificing its advantages under higher concurrency.

In the single-user run, results show a slight increase in latency for queries TCR1, TCR2, TCR7, TCR9, and TCR12 when moving from 32 to 64 CPUs. By contrast, this does not appear in the multi-user scenario. This suggests that once the workload saturates roughly 32 CPUs, adding more threads can actually incur extra overhead (e.g., for scheduling and coordinating join tasks) without delivering further speedup. Specifically, when the dataset is relatively small, distributing its partitions (morsels) across numerous threads yields diminishing returns – each morsel becomes so small that parallelization overhead outweighs any gains. Under concurrent workloads, however, the higher core count is still well utilized because multiple queries can keep all cores busy.

Finally, we discuss the results for TCR3 and TCR5, which are shown in Fig. 10. Both queries require finding specific paths and are processed using the semi-join decomposition (SJD) technique from Sec. 6. In the single-user run, Siren Federate returns results for both

queries in a matter of seconds. Notably, this is thanks to SJD, as an implementation using the chains of inner-joins was unable to complete the queries. Such naive implementation was failing in over half of the runs due to the excessive memory consumption caused by the combinatorial growth of intermediate results. This highlights the benefits of SJD that can be explained by (1) the presence of a reachability test to prevent unnecessary computation while expanding a path; (2) the availability of different semi-joins strategies to handle different complexities in terms of data to be joined; and (3) the fact that, at the time of writing, neither query planners support the ability to push JOIN operations to left SCANs to reduce the amount of data to process.

As with the previous queries, we observe that AQP and SQP exhibit similar performance for TCR3, whereas SQP is significantly faster than AQP for TCR5. Examining the query plan execution logs reveals that SQP spawned a higher number of parallel tasks in certain phases, despite both planners having comparable per-task costs. This parallelization advantage impacts TCR5 because it executes joins that are more computationally intensive than the joins in TCR3. This mirrors our earlier observations that the lower task-parallelism of AQP can increase latency in single-user scenarios, whereas SQP tends to distribute the same workload across more threads, thereby completing faster when system resources are not contended by multiple concurrent workload.

Unfortunately, we cannot comment on the results for 10 concurrent users, as the system was unable to process them due to memory constraints. We believe this limitation is specific to the experimental setting and will be mitigated with additional main memory, either on a single computing instance or across multiple instances.

Our evaluation on the LDBC Finbench benchmark demonstrates the effectiveness of semi-join decomposition and adaptive query planning for complex graph queries. Building on these results, future work could explore: (1) testing with larger-scale Finbench datasets to assess SJD scaling properties across different graph structures; (2) comparative analysis with BFS-based traversal to better understand the performance trade-offs between approaches; and (3) exploring how semi-join decomposition could be applied to a broader range of graph queries, with the goal of developing a comprehensive cost model that would enable the query planner to automatically select the optimal strategy for each query type.

7.3. Real-World Deployment Insights

To validate our system's practical applicability, we examined, among others, a real-world deployment at Apollo.io[14] involving a 350-node Elasticsearch cluster processing nearly half a petabyte of data with multiple concurrent users. The deployment demonstrated Siren Federate's ability to handle intricate data relationships and high-concurrency scenarios, demonstrating the effectiveness of our distributed join and query optimization techniques. Empirical results showed a dramatic reduction in query response time from an average of 7 seconds to sub-second, while significantly improving cluster stability and reducing query failures. These insights complement our quantitative benchmarks by providing empirical validation of Siren Federate's architectural design, specifically its approach to bridging document-oriented and relational data models in large-scale distributed environments.

[14] https://siren.io/case-study-transforming-enterprise-search-at-apollo-with-siren-federate/

8. Conclusion

This paper presented Siren Federate, a system that enables efficient exploratory analysis of large-scale knowledge graphs by bridging document-oriented, relational, and graph data models. Our architecture addresses challenges faced by investigative intelligence workflows that require both advanced search capabilities and complex graph operations at scale.

Key contributions include integrating relational join operations within the document-oriented model, leveraging IR system capabilities, and implementing distributed join algorithms optimized for IR systems. We also introduced adaptive query planning for accurate runtime cardinality estimation, query plan folding to reduce redundant computations, and semantic caching to enhance iterative query performance. Columnar in-memory processing and Elasticsearch's log-structured distributed architecture were also highlighted.

An important contribution is a approach to path queries through Semi-Join Decomposition (SJD). SJD reduces the combinatorial explosion of intermediate results by breaking down path queries into semi-joins. This approach minimizes memory usage and computational overhead, making it more scalable and efficient for large graphs. While SJD is applicable to general multi-hop path queries, it is particularly effective in the context of all-shortest-paths problems, offering a practical solution to the challenges faced by traditional methods. SJD's integration with Siren Federate's adaptive query planner and semantic caching further enhances its effectiveness for exploratory graph analysis.

We validated the effectiveness of our architecture and techniques through two evaluation scenarios. First, using a synthetic dataset containing billions of documents, we demonstrated the system's capability to sustain sub-second to second response times. Second, leveraging the LDBC Financial Benchmark, we assessed the system's proficiency in efficiently handling complex graph queries in a matter of seconds. Additionally, we presented insights from a real-world deployment involving 350 nodes, highlighting the practical applicability and robustness of the system architecture under operational conditions.

In conclusion, our architecture retains the search and relevance ranking capabilities of the IR system while introducing efficient relational operations and graph analytics at scale. This demonstrates how a combination of document and relational system features can enhance scalability and analytical capabilities for exploratory analysis of large knowledge graphs. Looking forward, our research will focus on further enhancing parallel processing across data segments, extending the suite of available graph analytic techniques, and refining adaptive optimization methods to continually improve performance and usability. Additionally, we plan to further develop Semi-Join Decomposition by exploring broader applicability to complex graph patterns, and more rigorous theoretical analysis. Future work will also involve comparative benchmarking of SJD against alternative graph querying approaches to clearly delineate its strengths and boundaries of effectiveness.

Acknowledgments. The first author was supported by ANR through the CPJ project no. ANR-22-CPJ2-0107-01. We would like to thank Martin Anseaume, George Apaaboah, Issac Garcia, David Homes, Johnny Hujol, Flavio Pompermaier, and all past members of our engineering team for their significant contributions to the development of Siren Federate. Their efforts were essential to the realization of the system described in this work. We used ChatGPT to improve the writing in some parts of this otherwise original manuscript. All AI-generated text has been carefully reviewed to ensure accuracy and correctness.

References

1. Akhgar, B., Bayerl, P.S., Sampson, F. (eds.): Open Source Intelligence Investigation. Springer (2016)
2. Angles, R.: Path-based algebraic foundations of graph query languages. In: Proc. EDBT (2025)
3. Atzenbeck, C., Ozgul, F., Hicks, D.L.: Linking and organising information in law enforcement investigations. In: Proc. IV (2009)
4. Balalau, O., Barret, N., Ebel, S., Galizzi, T., Manolescu, I., Mohanty, M.: Graph lenses over any data: the ConnectionLens experience. In: Proc. ICDEW (2024)
5. Besta, M., Gerstenberger, R., Peter, E., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., Hoefler, T.: Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries. ACM Comput. Surv. 56(2) (2023)
6. Bonifati, A., Fletcher, G., Voigt, H., Yakovets, N., Jagadish, H.V.: Querying Graphs. Morgan & Claypool Publishers (2018)
7. Campinas, S., Catena, M., Delbru, R.: Relational search and its application to investigative intelligence scenarios. In: Proc. IIR (2023)
8. Campinas, S., Catena, M., Delbru, R.: Siren Federate: Bridging the gap between document and relational data systems for efficient exploratory graph analysis. In: Proc. IDEAS (2024)
9. Campinas, S., Catena, M., Delbru, R.: Method to reduce network communication for distributed join on shared-nothing databases. International (PCT) Application PCT/EP2023/059801 (2023)
10. Chasseur, C., Li, Y., Patel, J.M.: Enabling JSON document stores in relational systems. In: Proc. WebDB (2013)
11. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Third Edition. The MIT Press, 3rd edn. (2009)
12. Davoudian, A., Chen, L., Liu, M.: A survey on NoSQL stores. ACM Comput. Surv. 51(2) (2018)
13. Delbru, R., Campinas, S.: Semantic caching of semi-join operators in shared-nothing and log-structured databases. u.s. Patent Application 17/597923 (2020)
14. Deshpande, A., Ives, Z., Raman, V.: Adaptive query processing. FnT Databases 1(1) (2007)
15. Deutsch, A., Francis, N., Green, A., Hare, K., Li, B., Libkin, L., Lindaaker, T., Marsault, V., Martens, W., Michels, J., et al.: Graph pattern matching in GQL and SQL/PGQ. In: Proc. SIGMOD/PODS (2022)
16. Gao, J., Jin, R., Zhou, J., Yu, J.X., Jiang, X., Wang, T.: Relational approach for shortest path discovery over large graphs. arXiv preprint arXiv:1201.0232 (2011)
17. Gormley, C., Tong, Z.: Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine. O'Reilly (2015)
18. Hao, K., Yuan, L., Zhang, W.: Distributed hop-constrained s-t simple path enumeration at billion scale. Proc. VLDB Endowment 15(2) (2021)
19. Hatcher, E., Gospodnetic, O.: Lucene in Action (In Action Series). Manning (2004)
20. Heinz, S., Zobel, J., Williams, H.E.: Burst tries: A fast, efficient data structure for string keys. ACM Trans. Inf. Syst. 20(2) (2002)
21. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. ACM Comput. Surv. 54(4) (2021)
22. Jamil, H.M., Rubaiat, S.Y.: Online digital investigative journalism using SociaLens. In: Proc. iiWAS (2025)
23. Jindal, A., Madden, S., Castellanos, M., Hsu, M.: Graph analytics using vertica relational database. In: Proc. Big Data (2015)
24. Karpathiotakis, M., Alagiannis, I., Ailamaki, A.: Fast queries over heterogeneous data through engine customization. Proc. VLDB Endowment 9(12) (2016)
25. Kejriwal, M., Szekely, P., Knoblock, C.: Investigative knowledge discovery for combating illicit activities. IEEE Intelligent Systems 33(1) (2018)

26. Kersten, T., Leis, V., Kemper, A., Neumann, T., Pavlo, A., Boncz, P.: Everything you always wanted to know about compiled and vectorized queries but were afraid to ask. Proc. VLDB Endowment 11(13) (2018)
27. Khan, A.: Knowledge graphs querying. ACM SIGMOD Record 52(2) (2023)
28. Kiehn, F., Schmidt, M., Glake, D., Panse, F., Wingerath, W., Wollmer, B., Poppinga, M., Ritter, N.: Polyglot data management: state of the art & open challenges. Proc. VLDB Endowment 15(12) (2022)
29. Lai, Z., Peng, Y., Yang, S., Lin, X., Zhang, W.: PEFP: Efficient k-hop constrained s-t simple path enumeration on fpga. In: Proc. ICDE (2021)
30. Leis, V., Radke, B., Gubichev, A., Kemper, A., Neumann, T.: Cardinality estimation done right: Index-based join sampling. In: Proc. CIDR (2017)
31. Lissandrini, M., Mottin, D., Hose, K., Pedersen, T.: Knowledge graph exploration systems: are we lost? In: Proc. CIDR (2022)
32. Liu, Y., Lin, Q., Hong, B., Peng, Y., Hjerpe, D., Liu, X.: Resonance algorithm: an intuitive algorithm to find all shortest paths between two nodes. Complex & Intelligent Systems 9(4) (2023)
33. Liu, Z., Heer, J.: The effects of interactive latency on exploratory visual analysis. IEEE Trans. Vis. Comput. Graph. 20(12) (2014)
34. Lu, A., Fang, Z.: SQL2FPGA: Automatic acceleration of SQL query processing on modern CPU-FPGA platforms. In: Proc. FCCM (2023)
35. Lu, J., Holubová, I.: Multi-model databases: A new journey to handle the variety of data. ACM Comput. Surv. 52(3) (2019)
36. Manola, F., Miller, E., McBride, B.: RDF primer (2004), `https://www.w3.org/TR/rdf-primer/`
37. Manolescu, I., Mohanty, M.: Full-power graph querying: state of the art and challenges. In: Proc. VLDB (2023)
38. Martens, W., Niewerth, M., Popp, T., Rojas, C., Vansummeren, S., Vrgoč, D.: Representing paths in graph database pattern matching. Proc. VLDB Endowment 16(7) (2023)
39. Moore, E.: The Shortest Path Through a Maze. Technical publications, Bell Telephone System. (1959)
40. Mulder, T., Fletcher, G., Yakovets, N.: Optimizing navigational graph queries. The VLDB Journal 34 (2025)
41. Ngo, H.Q., Porat, E., Ré, C., Rudra, A.: Worst-case optimal join algorithms. Journal of the ACM 65(3) (2018)
42. Noel, S., Bodeau, D.J., McQuaid, R.: Big-data graph knowledge bases for cyber resilience. In: Proc. NATO IST-153 (2017)
43. O'Neil, P., Cheng, E., Gawlick, D., O'Neil, E.: The log-structured merge-tree (LSM-tree). Acta Informatica 33(4) (1996)
44. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. In: Proc. ISWC (2006)
45. Papailiou, N., Tsoumakos, D., Karras, P., Koziris, N.: Graph-aware, workload-adaptive SPARQL query caching. In: Proc. SIGMOD (2015)
46. Paradies, M., Lehner, W., Bornhövd, C.: GRAPHITE: an extensible graph traversal framework for relational database management systems. In: Proc. SSDBM (2015)
47. Peng, Y., Lin, X., Zhang, Y., Zhang, W., Qin, L., Zhou, J.: Efficient hop-constrained s-t simple path enumeration. The VLDB Journal 30 (2021)
48. Pini, D., Tummarello, G., Delbru, R.: Optimization of database sequence of joins for reachability and shortest path determination. U.S. Patent 11720564 (2022)
49. Pokornỳ, J.: Graph databases: their power and limitations. In: Proc. IFIP (2015)
50. Procopiuc, O., Agarwal, P.K., Arge, L., Vitter, J.S.: Bkd-tree: A dynamic scalable kd-tree. In: Advances in Spatial and Temporal Databases (2003)

51. Qi, S., Lin, H., Guo, Z., Szárnyas, G., Tong, B., Zhou, Y., Yang, B., Zhang, J., Wang, Z., Shen, Y., et al.: The LDBC financial benchmark. arXiv preprint arXiv:2306.15975 (2023)
52. Roijackers, J., Fletcher, G.H.L.: On bridging relational and document-centric data stores. In: Big Data (2013)
53. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Press, 3rd edn. (2009)
54. Sahal, R., Khafagy, M., Omara, F.: Big data multi-query optimisation with Apache Flink. Int. J. Web Eng. Technol. 13 (2018)
55. Sammons, J.: The basics of digital forensics: the primer for getting started in digital forensics. Syngress (2014)
56. Schuh, S., Chen, X., Dittrich, J.: An experimental comparison of thirteen relational equi-joins in main memory. In: Proc. SIGMOD (2016)
57. Shen, S., Yao, Z., Shi, L., Wang, L., Lai, L., Tao, Q., Su, L., Chen, R., Yu, W., Chen, H., et al.: Bridging the gap between relational OLTP and graph-based OLAP. In: Proc. USENIX (2023)
58. Sun, S., Chen, Y., He, B., Hooi, B.: PathEnum: Towards real-time hop-constrained s-t path enumeration. In: Proc. SIGMOD/PODS (2021)
59. Träris, T., Balsacq, M.: Where the path leads: State of the art and challenges of graph database systems. informatikJournal 12 (2021)
60. Tziavelis, N., Gatterbauer, W., Riedewald, M.: Beyond equi-joins: Ranking, enumeration and factorization. Proc. VLDB Endowment 14(11) (2021)
61. Ágnes Vathy-Fogarassy, Hugyák, T.: Uniform data access platform for SQL and NoSQL database systems. Information Systems 69 (2017)
62. Vrgoč, D.: Evaluating regular path queries under the all-shortest paths semantics. arXiv preprint arXiv:2204.11137 (2022)
63. Vrgoc, D., Rojas, C., Angles, R., Arenas, M., Calisto, V., Farías, B., Ferrada, S., Heuer, T., Hogan, A., Navarro, G., et al.: MillenniumDB: A multi-modal, multi-model graph database. In: Proc. SIGMOD/PODS (2024)
64. ten Wolde, D., Singh, T., Szárnyas, G., Boncz, P.: DuckPGQ: Efficient property graph queries in an analytical RDBMS. In: Proc. CIDR (2023)
65. Yakovets, N., Godfrey, P., Gryz, J.: Query planning for evaluating SPARQL property paths. In: Proc. SIGMOD/PODS (2016)
66. Zhang, C., Lu, J., Xu, P., Chen, Y.: UniBench: a benchmark for multi-model database management systems. In: Proc. TPCTC (2019)
67. Zhang, Z., Deshmukh, H., Patel, J.M.: Data partitioning for in-memory systems: Myths, challenges, and opportunities. In: Proc. CIDR (2019)
68. Zhao, K., Yu, J.X.: All-in-one: Graph processing in RDBMSs revisited. In: Proc. SIGMOD-/PODS (2017)

A. LDBC Finbench

This section reports the queries and benchmark results discussed in the evaluation Section 7.2.

A.1. GQL Queries

LDBC introduces several workloads for measuring different aspects of a graph engine. We express some of those workload as GQL queries that Siren Federate can interpret and internally translate into query plans for processing. We here lists the queries, for the complex translation workload, that we used in our experiments.

These queries are parameterized with a START and END variables to represent date ranges, and a PERSON_ID and ACCOUNT_ID variables to identify people and accounts. WHERE clauses – used by GQL to filter entities that match a node pattern – are here expressed using the Lucene query syntax. Finally, SELECT statements are amended to return raw data points since aggregation operators such as SUM are not supported yet at the time of writing.

```
SELECT other.id, medium.mediumType, medium.id
FROM "ldbc-finbench"
MATCH (medium:Medium WHERE "isBlocked:true")
      ->(:MediumSignInAccount)
      ->(other:Account)
      (
        (:Account)
        <-(:AccountTransferAccount WHERE "createTime:{START TO END}")
        <-(:Account)
      ){1,3}
      (account:Account WHERE "id:ACCOUNT_ID")
```

Listing 1.1. TCR1

```
SELECT other.id, l.loanAmount, l.balance
FROM "ldbc-finbench"
MATCH (l:Loan)
      ->(:LoanDepositAccount WHERE "createTime:{START TO END}")
      ->(other:Account)
      (
        (:Account)
        ->(:AccountTransferAccount WHERE "createTime:{START TO END}")
        ->(:Account)
      ){1,3}
      (:Account)<-(:PersonOwnAccount)<-(person:Person WHERE "id:PERSON_ID")
```

Listing 1.2. TCR2

```
SELECT trace
FROM "ldbc-finbench"
MATCH trace = ALL SHORTEST (src:Account WHERE "id:ACCOUNT_ID")
      (
        (:Account)
        ->(:AccountTransferAccount WHERE "createTime:{START TO END}")
        ->(:Account)
      ){1,10}
      (dst:Account WHERE "id:ACCOUNT_ID")
```

Listing 1.3. TCR3

```
SELECT trace
FROM "ldbc-finbench"
MATCH trace =
      (
        (:Account)
        <-(:AccountTransferAccount WHERE "createTime:{START TO END}")
        <-(:Account)
      ) {1,3}
      (account:Account)<-(:PersonOwnAccount)<-(:Person WHERE "id:PERSON_ID")
```

Listing 1.4. TCR5

```
SELECT src.id, dst.id, edge1.amount, edge2.amount
FROM "ldbc-finbench"
MATCH (src:Account)
      ->(edge1:AccountTransferAccount WHERE "amount:{0 TO *} AND createTime:{START
     TO END}")
```

```
      ->(mid:Account WHERE "id:PERSON_ID")
      ->(edge2:AccountTransferAccount WHERE "amount:{0 TO *} AND createTime:{START
    TO END}")
      ->(dst:Account)
```

Listing 1.5. TCR7

```
SELECT edge1.amount, edge2.amount, edge3.amount, edge4.amount
FROM "ldbc-finbench"
MATCH (up:Account)
      ->(edge3:AccountTransferAccount WHERE "amount:{0 TO *} AND createTime:{START
    TO END}")
      ->(mid:Account WHERE "id:PERSON_ID")
      ->(edge4:AccountTransferAccount WHERE "amount:{0 TO *} AND createTime:{START
    TO END}")
      ->(down:Account)
      (mid)<-(edge1:LoanDepositAccount WHERE "amount:{0 TO *} AND createTime:{START
    TO END}")
      (mid)<-(edge2:AccountRepayLoan WHERE "amount:{0 TO *} AND createTime:{START TO
     END}")
```

Listing 1.6. TCR9

```
SELECT l.loanAmount
FROM "ldbc-finbench"
MATCH (l:Loan)<-(:PersonApplyLoan)<-(p2:Person)
      (
        (:Person)<-(g:PersonGuaranteePerson WHERE "createTime:{START TO END}")
        <-(:Person)
      ) {1,10}
      (p1:Person WHERE "id:PERSON_ID")
```

Listing 1.7. TCR11

```
SELECT compAcc.id, edge2.amount
FROM "ldbc-finbench"
MATCH (company:Company)
          ->(:CompanyOwnAccount)
          ->(compAcc:Account)
          <-(edge2:AccountTransferAccount WHERE "createTime:{START TO END}")
          <-(pAcc:Account)
          <-(:PersonOwnAccount)
          <-(person:Person WHERE "id:PERSON_ID")
```

Listing 1.8. TCR12

A.2. Raw Benchmark Results

Table 1. Query times (P90) in ms for LDBC queries.

Planner	Users	CPUs	TCR1	TCR2	TCR3	TCR5	TCR7	TCR9	TCR11	TCR12
Adaptive	1	8	1,429	818	3,590	5,725	6,775	5,263	200	245
		16	667	732	3,623	6,205	6,521	4,983	140	223
		32	685	728	3,348	5,884	5,732	4,171	133	221
		64	868	910	3,146	4,706	6,757	4,743	133	298
	10	8	25,454	22,194	N/A	N/A	25,839	14,864	12,905	8,276
		16	9,288	8,676			17,074	11,452	5,051	3,075
		32	1,656	1,761			10,058	7,047	912	667
		64	1,256	1,397			9,062	6,375	281	426
Static	1	8	1,779	1,199	4,208	4,464	7,552	5,691	337	348
		16	946	799	3,874	4,573	7,345	5,244	326	242
		32	611	595	3,609	4,329	5,271	3,628	289	179
		64	836	645	3,403	3,478	6,184	4,181	287	210
	10	8	26,001	24,127	N/A	N/A	28,030	15,521	14,210	8,943
		16	9,994	10,072			16,678	11,783	5,838	3,555
		32	2,312	1,635			9,838	7,271	1,661	1,290
		64	1,236	1,132			8,713	5,941	499	369

Georgeta Bordea holds a Junior Professorship in Natural Language Processing (NLP) for Digital Humanities at the L3i Laboratory (Informatics, Image, and Interaction) of La Rochelle University. She is particularly interested in narrative understanding and the application of NLP techniques in social science and economics, as well as automating the extraction, translation, and alignment of scientific terms across different domains and languages, leveraging large language models.

Stéphane Campinas completed his Ph.D. at the University of Galway, where he focused on Information Retrieval for Linked Data, sparking a lasting interest in the field. Following his doctoral studies, he joined Siren, where he contributed to the development of Siren Federate. He is currently one of the lead architects of the system.

Matteo Catena holds a Ph.D. in Computer Science from the Gran Sasso Science Institute of L'Aquila, where his research focused on efficient Information Retrieval. During his doctoral studies, he was a research associate at the High Performance Computing Lab of ISTI-CNR in Pisa, where he later continued as a postdoctoral researcher. Following his academic work, he contributed to search technologies in two industry roles before joining Siren, where he is currently a researcher on the Siren Federate team.

Renaud Delbru holds a Ph.D. in Computer Science from the University of Galway, where his research focused on information retrieval techniques for semantic web data. Follow-

ing his doctoral studies, he continued as a postdoctoral researcher at the Insight Research Center for Data Analytics (formerly known as Digital Enterprise Research Institute). He co-founded Siren in 2010, where he currently serves as Chief Scientific Officer, leading research on distributed search systems, the integration of knowledge graphs with modern AI technologies, and the application of graph analytics to investigative intelligence scenarios.

Received: April 01, 2025; Accepted: July 12, 2025.

Computer Science and Information Systems 23(1):513–534 https://doi.org/10.2298/CSIS250226013B

Using Genetic Programming as a Feature Selector and Classifier to Implement Bankruptcy Prediction Models*

Ángel Beade[1], José Santos[2], and Manuel Rodríguez[1]

[1] Business Department, University of A Coruña,
Campus de Elviña, s/n 15071 A Coruña, Spain
{a.beade, manuel.rodriguez.lopez}@udc.es
[2] CITIC (Centre for Information and Communications Technology Research), Department of Computer Science and Information Technologies, University of A Coruña, Campus de Elviña, s/n 15071 A Coruña, Spain
jose.santos@udc.es

Abstract. Genetic Programming (GP) was used as a feature selector and classifier to implement bankruptcy prediction models for medium-sized companies. Two sets of input variables were used for the prediction models: one using a large number of exclusively financial variables and the other incorporating variables from the economic environment, which allows analyzing the capability of the latter to improve performance. Two strategies were defined for GP as a feature selector, based on the statistical relevance of the selected features in the GP process, with a novel proposal based on a progressive reduction of the set of selected variables and with the aim of minimizing the risk of eliminating relevant features. An analysis is performed of the improvement obtained with feature selection with both GP-based methods in comparison with the use of complete sets of variables and using GP as a classifier. With the selected variables, we also compared GP as a classifier with respect to other standard classifiers, using automatic parameter adjustment with AutoWeka for these classifiers. The best results are obtained with the synergy of using GP as a feature selector and as a classifier, with the advantage of the direct interpretability that GP provides in the application.

Keywords: Genetic programming, feature selection, bankruptcy prediction models.

1. Introduction

The prediction of insolvency or bankruptcy of a company was one of the traditional research topics in financial economics. In the last two decades, Machine Learning (ML) models have emerged as a better alternative to traditional prediction models based on statistical models [1, 50].

The bankruptcy prediction problem is treated, with the information available in the financial statements of companies, as a classification problem. In the ML field, there are a large number of classification techniques applied to bankruptcy prediction, including ensembles of classifiers [32, 35, 46] and deep learning methods [29, 30, 47]. However, our previous work [6–8] has focused on the use of Genetic Programming (GP) [25, 39].

*This paper is an extension of INISTA 2024 Conference.

The main reason is that GP, by evolving "programs" (usually using a tree-based representation), allows for straightforward and easier interpretability (at least readability) compared to classification techniques that are considered black boxes. This is very important in this field from the point of view of the end user of the prediction models, as well as of the economic and political institutions that are beginning to demand interpretability and explainability of the prediction results [14–16]. Moreover, GP does not impose restrictions on the explanatory variables of the prediction models (contrary to classical statistical methods). Finally, the complexity of the evolved programs can be easily regulated. For example, the maximum depth of the evolved classification trees, the maximum number of nodes in the evolved trees and the set of functions that can be used by the tree nodes are decisions aimed at facilitating interpretability while maintaining high prediction performance. Moreover, as shown in previous work [6], GP can obtain bankruptcy prediction models (BPMs) with high performance and stable predictive power over time, even considering a long post-training period that includes a changing economic environment (with a sharp recession and a strong recovery) [6].

Furthermore, the selection of explanatory variables for the BPMs is, after modeling techniques, a major issue in business failure research [48] and studies show that BPMs can be more effective with proper variable selection [46]. But variable selection is not an issue that can be addressed in isolation, as has been repeatedly pointed out since early work on business failure prediction, as variable selection and the classification method are intrinsically linked [1, 17, 51], and there is no consensus on which variables are relevant or how to perform variable selection [1, 3, 4, 19, 21, 26, 32]. One of the common ways to select independent variables in a model is to start from an initial set of available variables of high dimensionality and reduce this dimensionality by means of Feature Selection (FS) methods, FS that is also expected to improve model performance.

In our previous study [7], we analyzed the suitability of a method based on GP to address dimensionality reduction through feature selection. The FS method is based on the relative frequency of occurrence, in GP evolved solutions, of the input variables and their statistical significance. In this aforementioned study [7], FS by means of GP was compared with other widely used FS methods. The comparison was also performed using a set of 15 classification methods, also including GP as a classifier.

Previous works [34, 42] have also considered the frequency of occurrence of variables in the GP process as a basis for performing FS. Nevertheless, Neshatian and Zhang [34] only considered the best final evolved solutions, which is a very small subset where the optimized trees can be very similar. But, as Liu et al. [28] point out, even considering the best-evolved solutions with GP, "the features generated in the output individual may still contain redundant features". Liu et al. [28] addressed this problem by considering the best individuals generated during GP evolution instead of a single best individual, also in combination with a feature ranking. However, the bloat problem may appear in many individuals in the population, since a part of the tree that is irrelevant to the fitness of a solution may be maintained in several solutions in the population.

To address the problem and ensure a correct FS process, decreasing the possibility of selection of irrelevant variables due to the GP bloat problem, our previous proposal [7] considers a large number of independent GP runs to compute the frequency of variable occurrence in the GP process (logically with the disadvantage of increased

computational time). Moreover, unlike the previous works discussed [28, 34, 42], the FS process defined in [7] also considered the statistical relevance of variable occurrence, with the aim of ensuring a correct selection of the most significant variables. This FS process based on GP presented better results considering efficiency (i.e., classification improves when using FS compared to using all features) and also performance (i.e., for a particular classifier, the classification results obtained with the different FS methods are compared) with respect to other feature selection methods [7].

This work proposes an alternative to the one presented in [7], with a new proposal that aims to minimize the risk of information loss due to discarded variables. This second proposed methodology is a stepwise process that addresses a progressive reduction in the size of the set of explanatory variables by eliminating, at each stage, those variables clearly showing irrelevance, thus minimizing the risk of possible elimination of relevant variables. Since the work in [7] performed a detailed comparison of the GP-based FS with other widely used FS methods, the present work focuses on the comparison of the new alternative with the previous one.

In addition, to make the conclusions of the comparison more robust, two different initial sets of variables were used as possible explanatory variables of the GP-evolved BPMs (different BPMs with different prediction time horizons). The first set is defined with only financial variables of the company and the second set is an extended set incorporating, among others, variables of the economic environment. This use of two sets of variables also allows us to situate the results of GP-evolved BPMs in relation to the existing debate in the field between those authors who use and defend the sufficiency of using only explanatory variables constructed exclusively from accounting data [10, 12, 45], and those authors who incorporate non-financial variables [3, 4, 20, 40]. In this regard, our previous work presented at the INISTA 2024 conference [8], analyzed the comparison of performance prediction with both sets of explanatory variables. Thus, the present work extends that comparison when different feature selection methods are used.

Furthermore, apart from the selection of relevant explanatory variables provided by GP-based FS methods, to evaluate GP as a classification method, in this work a comprehensive comparison with other classification methods has been carried out. For this purpose, the Weka package called AutoWeka [24] was used, which provides automatic adjustment of the parameters of the other ML classifiers.

Therefore, the main objective of this paper is to analyze the GP-based FS methods, with emphasis on the novel proposal oriented to minimize the risk of eliminating possible relevant variables in its iterative FS procedure. For the analysis of the FS methods, different BPMs with different prediction horizons are evolved, also considering the two sets of possible explanatory variables mentioned above, where one includes non-financial variables. In addition, an analysis is performed comparing GP as a classifier to implement BPMs with respect to other ML classification methods, where the AutoWeka environment automates the parameterization of the classifiers, an analysis oriented to emphasize the link between the selection of variables and the classification methods.

The remainder of this article is structured as follows. Section 2 details the different methods used for the implementation of the BPMs, such as the explanation of the input variables in the two sets of variables considered, the dataset used to evolve the GP-based BPMs, as well as the FS methods considered, with emphasis on the new proposal of this work. Section 3 details the results in the comparison with the two FS strategies

considered, as well as the performance results when using both sets of possible input variables in the BPMs. Section 3 also details the performance comparison between GP as a classifier with respect to other ML classifiers. Finally, Section 4 summarizes the main conclusions of the paper and possible avenues for further research.

2. Methods

2.1. Input Variables for the Prediction Models

Two sets of input variables were used as inputs for the prediction models. The first set includes only financial variables of the companies (set A). This set includes 97 explanatory variables that refer to a wide range of aspects that are considered, a priori, relevant to business failure. Table 1 shows the categories (standard categorization in financial analysis) of these variables.

The explanatory variables of the BPMs included in set A were preferentially chosen based on two criteria: i) relevance shown in the bibliography and their presence in prediction models contrasted in previous work. Most of the variables correspond to financial ratios (understood as the quotient of two financial variables), as they are the traditional basis for financial analysis. ii) other variables collected from the annual accounts of the companies and which refer to different aspects that are rarely used (e.g., variations in ratios or variations in magnitudes) or novel in this field (e.g., degree of the balance sheet decomposition and variables referring to fraud and productivity)

It is important to emphasize the large number of explanatory variables considered, which is much higher than the number usually used in the literature dealing with insolvency prediction [3, 44]. According to systematic reviews, the trend in contemporary models is to use between 15 and 30 variables in the initial stage, especially when employing advanced machine learning and big data techniques. However, automatic variable selection processes tend to reduce the final model to 5-8 predominant ratios. This pattern appears in international contexts (US, Europe, India, etc.) and in various business sectors [1–3, 11, 27, 41].

The second set extends set A with four qualitative non-financial variables with company data and 11 quantitative variables with information on the economic environment (set B). The 4 qualitative variables correspond to the age of the company (3 categories considering the age in ranges 0-3 years, 4-9 years and more than 9 years), the auditor's qualification in the financial year (negative, not available or positive) and two variables related to financial stress (true/false/not available). The 11 variables that attempt to reflect the economic environment are defined considering aspects (in each financial year and referring to Spain) such as the gross operating surplus and gross mixed income, the number of bankrupt companies, the gross domestic product, the market prices, the employment level and interest rates. Note that, with the objective (not addressed here) of counteracting the prediction of failure and thus minimizing the risk of bankruptcy, the firm can act on all the variables in set A, while it cannot act on the variables of the economic environment and on most of the firm's qualitative variables, which are incorporated in set B.

Table 1. Sets A and B of input variables

Category	Number	Set
Changes in financial ratios	2	A, B
Contribution	2	A, B
Decomposition degree	3	A, B
Efficiency	11	A, B
Financial structure	14	A, B
Fraud	11	A, B
Growth	1	A, B
Interest expenses	5	A, B
Liquidity and solvency	18	A, B
Productivity	4	A, B
Profitability	12	A, B
Size	4	A, B
Turnover	7	A, B
Variations in magnitudes	3	A, B
Qualitative variables of the company	4	B
Economic environment variables	11	B

In the case of the quantitative variables in set A, since some of the company's observations (financial information of the company in a financial year) present an outlier in one of the numerous explanatory variables considered, a limiting of extreme values has been carried out. To this end, the variables of a company in a financial year with values above the 97.5% percentile or below the 2.5% percentile, corresponding to the distribution of a specific variable, are replaced with the respective reference percentile value. The objective is to eliminate values that are clearly outside the usual range of values and, thus, the difficulties that they entail for learning. This delimitation of values is common in ML [36, 38, 43] and was carried out in order not to eliminate information from companies, with the aim of including them in the training and test sets (next subsection).

In addition, as a last step, with the values of the variables bounded (of set A), a transformation is performed for each variable (according to a logistic distribution and considering the values in the training period detailed below), with the aim of improving the generalization capacity of the prediction models.

2.2. Dataset and Prediction Models

This paper considers the prediction of corporate insolvency in medium-sized Spanish companies. A population of 11,158 firms is considered (10,091 classified as non-failed, 1,067 classified as failed), with their accounting information from 2005 to 2019.

Both the accounting data (balance sheet and income statement) and other relevant data (date of incorporation, audit, etc.) have been obtained from the Iberian Balance Sheet Analysis System (SABI) database [52]. The concept of bankruptcy followed in this work refers to the legal declaration of suspension of payments, which is the definition most commonly considered in the field [27, 50].

A large set of different BMPs were considered to draw solid conclusions with the GP processes: first, 9 BPMs labeled M1, M2, … M9, named as "annual BPMs" as these predict the failure of a company 1, 2, … and 9 years in advance. Secondly, five "multiperiod BPMs" [9], which predict failure over a range of future years (instead of failure at a specific future point in time). For example, BPM M1-3 predicts the failure of a company in the interval between 1 and 3 years in the future.

The use of multi-period BPMs, although less common in the field of financial economics, is an interesting approach since, for example, a lender is surely more interested in estimating bankruptcy during the loan period (rather than at a specific point in the future), as discussed in [9]. Consequently, we have also incorporated this alternative, in addition to the classic annual BPMS, into the analysis. Therefore, this set of annual/multiperiod BPMs covers a wide range of prediction years corresponding to short-, medium- and long-term prediction horizons.

The training and test sets of all these prediction models include observations of companies in the period 2005-2007. An "observation" is defined as the financial data of a company in a particular financial year. Thus, for example with the observations of 2005, M9 predicts failure in 2014 and with the observations of 2007 it predicts failure in 2016. Table 2 shows the number of observations in the training and test sets for the different prediction models. Note that the data are highly imbalanced in this application (between failed and healthy companies). However, balanced training sets are used (Table 2), where the available number of failed observations is randomly divided between the training and test set observations. In the case of non-failed observations, those in the training set are selected from the much larger total number of non-failed observations, while the remainder are inserted into the test set. The use of balanced training sets avoid biased predictions in the machine learning process.

Various authors provide the sizes of the training sets used in different BPM studies for different techniques and different time horizons [1, 3, 5, 20, 22]. In particular, the work of du Jardin [20] analyzes studies referring to annual models with a time horizon of up to 5 years, work in which the number of companies used in the training set and the wide variety of sizes in that set can be observed. Therefore, although there is no standard for determining the optimal size of a BPM training set, the range used in this study for annual models fits perfectly, for example, with the sizes in the studies reviewed by du Jardin and Séverin [22]. They also fit with the size of class 1 (failures) in the study by Altman et al. [3], which ranges from 38 to 180 failed companies.

The three consecutive years (2005-2007) are considered for the training and test sets, with the number of observations specified in Table 2. This allows for a long post-learning period to check, for example, the stability of predictions over that period (as performed in [6] on the performance of genetic programming-evolved BPMs over that period, which is not the aim here). It should be noted that there is no definitive consensus on the number of exercises to be used in training a model, as performance depends both on the quality of the model fit and the dataset used. In machine learning studies, it is common to use between 3 and 5 years of financial data for the training period [13].

Table 2. BMPs considered and their training/test sets

	BPM	Number of observations Training set Failures | non-failures	 Test set Failures | non-failures
Annual BPMs	M1	41 | 41	41 | 22,289
	M2	89 | 89	88 | 22,241
	M3	113 | 113	112 | 22,217
	M4	123 | 123	124 | 22,207
	M5	141 | 141	141 | 22,189
	M6	170 | 170	170 | 22,160
	M7	160 | 160	161 | 22,170
	M8	116 | 116	115 | 22,214
	M9	61 | 61	61 | 22,269
Multiperiod BPMs	M1-3	242 | 242	242 | 22,088
	M1-6	677 | 677	676 | 21,653
	M1-9	1,013 | 1,013	1,014 | 21,317
	M4-6	434 | 434	435 | 21,896
	M7-9	337 | 337	337 | 21,993

2.3. Genetic Programming FS Methods

This paper addresses dimensionality reduction by means of FS using GP. The underlying idea is that, in the evolutionary process of GP, variables that are relevant will be maintained, over generations, in the population, as opposed to irrelevant variables that will be progressively eliminated by selection pressure. Therefore, GP has an advantage in feature selection over other ML FS techniques, as it is an intrinsic selection process. This is because the evolution of the final solutions involves an automatic selection of which are the relevant input variables in the optimized solutions.

With this idea, two feature selection methods were considered to automatically obtain the most relevant variables for BPMs. The first is detailed in [7, 8], which is summarized below, while the second incorporates a new proposal as a variant of the first.

First variant of Genetic Programming Feature Selection ($GPFS_1$). For FS, a set of GP independent runs is considered. The relative frequency of occurrence of each explanatory variable in a subset of these independent GP runs is used. The most frequent inputs are expected to be the most relevant variables for the prediction models.

For FS, a null hypothesis is considered in which the frequency of occurrence of each explanatory variable is due to chance, leading to a normal distribution of selection for

each feature. A statistical test (Kolmogorov-Smirnov test), considering the real distribution obtained for each variable, allows to reject (or not) the aforementioned null hypothesis, and the p-value obtained from it will also allow establishing a relevance ranking of each variable. The steps of the GPFS1 process can be summarized as:

i. For each of the BPMs (M1, M2, ...), a large experiment (1,000 independent runs of GP) is available with the totality of the input variables. The high number of runs (1,000) guarantees that, in independent experiments with that number of runs, a very high degree of agreement in the selected variables is obtained (data not detailed here).
ii. Based on the results of the experiment with the independent runs, a subset of GP runs (5%) is established, those that provide solutions considered to be the best for the application: those solutions that provide a classification (evaluated on the test set) with the highest AUC (Area Under the ROC Curve) value. In addition, these solutions are selected with a filtering, choosing among the 5% of solutions with the best sum of true positive rate (TPR) + true negative rate (TNR). Therefore, solutions that focus their performance on the extremes of the ROC curve are avoided, which is especially relevant given that the test sets are unbalanced.
iii. For each explanatory variable, its relative frequency and its p-value are calculated (the null hypothesis being that the relative frequency is due to chance), both referring to the solutions evolved in the subset of selected GP runs (in the whole evolutionary process). Finally, those variables with p-value<0.05 will be selected, i.e., those with statistically relevant results and not due to chance. That is, the number of selected features (p-value<0.05) is determined automatically.

Finally, with the variables determined as most relevant in each prediction model, a new experiment is performed (1,000 new GP runs), but now with the variables selected with $GPFS_1$, which will provide the best final prediction model (selecting the one that provides the best AUC over the test set, filtering again from those with the best TPR + TNR). That is, GP is first used as an FS method and, in a second independent step, it is used as a classification method to provide the final evolved BPMs.

Second variant of Genetic Programming Feature Selection ($GPFS_2$). One of the risks of FS is losing information about relevant variables. Feature selection is a non-monotonic problem, since it is difficult for the best subset of p features to include the best subset of q features, with $q<p$. With $GPFS_1$, dimensionality reduction is addressed in terms of the statistical significance of the relative frequency of each of the variables in the input set (p-value<0.05). However, another approach that tries to avoid the loss of possible relevant variables can be analyzed.

Thus, the second method proposed here is a stepwise process that addresses a progressive decrease in the size of the set of explanatory variables, removing, in each of the steps, those variables that show clear irrelevance (p-value>=0.3333). Therefore, this second method follows a similar philosophy to that applied in Guyon et al. [18] with its recursive feature elimination algorithm. The main difference is that $GPFS_2$ takes into account the statistical significance of the presence of features.

Therefore, the aim is to progressively decrease the number of input variables, with the consequent minimization of the risk of elimination of relevant variables. To this end, it is established that the variables selected should have a p-value<0.3333 (which should

minimize the risk of eliminating relevant variables). A p-value<0.05 or p-value<0.01 are usual values for determining that a variable is statistically relevant. A p-value<0.3333 implies that variables with a probability of being relevant >= 66.67% will be taken, a very lax criterion, compared to the standard 95-99%.

Three variants were considered with this alternative. These three variants differ in the number of steps considered in the selection of variables, where each step selects a subset of the previously selected features:

1. $GPFS_{21}$, in which variables are selected according to the p-value of the relative frequency of each variable in the subset of solutions of the independent GP runs (p-value<0.3333, i.e., same as $GPFS_1$, changing the threshold p-value).
2. $GPFS_{22}$, repeating the process in a second iterative process. That is, using as inputs the subset of variables selected with $GPFS_{21}$, a subsequent experiment is performed with 1,000 independent GP runs, calculating again the relative frequency of occurrence of each variable. Choosing again those with p-value<0.3333 will provide a more limited set of $GPFS_{22}$ selected variables.
3. $GPFS_{23}$, repeating the process in a third iterative process, providing a set of $GPFS_{23}$ selected variables.
   In other words, a repetitive cycle of the successive process of elimination of non-relevant variables with 3 iterations.

Some considerations regarding $GPFS_1$ and $GPFS_2$. The FS methods are traditionally classified as filter, wrapper and embedded methods [23]. Filter methods attempt to detect variables whose values differ sufficiently between different classes, but not within instances of the same class, and are therefore independent of the classification method. In wrapper methods, features are evaluated based on their performance in a specific modeling algorithm, making them more computationally expensive. Finally, in embedded methods, feature selection is performed as the learning method is applied, since selection is integrated into the learning method. Consequently, the first consideration is that $GPFS_1$ and $GPFS_2$ can be classified as embedded FS methods, since the FS process is embedded in the GP learning process.

Secondly, these GPFS methods are context-sensitive, since the relevance of a feature is measured in the presence of other variables. Finally, $GPFS_1$ and $GPFS_2$ base their approaches on the statistical significance of variable appearance (contrary to the works in [34] and [42]), as indicated in the Introduction section. Moreover, the consideration of a large set of evolved trees in $GPFS_1$ and $GPFS_2$ (to calculate the frequency of variable occurrence) diminishes the possible effect of bloat, as parts of a tree that do not affect fitness (bloat) may appear in an evolved tree, but are unlikely to appear in all evolved trees and in independent GP runs.

2.4. GP Environment

All GP processes were implemented with the HeuristicLab (HL) environment [49, 53]. Standard tree-based representations were considered, modeled in HL as a problem of symbolic classification.

HL allows to divide the training set into a fitness subset and a validation subset. The fitness subset drives the evolutionary process, since the fitness of solutions (MSE, Table 3) is determined by this subset. At the end of the evolutionary process, HL provides two final solutions: the best training solution, which corresponds to the solution with the

best fitness in the total training set, and the best validation solution, which corresponds to the best program in the validation subset. The objective of the validation subset is to discover solutions that may perform well when generalized to the test set, that is, to prevent over-fitting. Note that these 2 solutions are the ones taken into account in each GP run to finally select the one with the best AUC (on the test set) (Section 2.3).

The GP parameter tuning was performed with a simple sweep of the most relevant parameters, while others were set to the standard values considered in HL (e.g., Model Creator and Solution Creator). Table 3 summarizes the GP parameters (HL nomenclature).

Table 3. GP parameters

GP Parameter	Value/option
Fitness function	MSE (Mean Squared Error, between real and predicted values)
Solution Creator	Probabilistic Tree Creator
Tree Grammar	Arithmetic functions (+, -, *, /)
Maximum Depth	10 (maximum depth of the tree/program)
Maximum Length	100 (maximum number of tree nodes)
Generations	100
Population Size	1,500
Selector	Tournament - Window size: 8 (used in mutation and crossover)
Elites	Preserves the best solution between generations
Crossover	Subtree Swapping at the crossing point
Mutation	Multi Symbolic Expression Tree Manipulator (allows different types of mutations)
Mut. Probability	15%
Fitness and validation sets	Fitness subset: 100 % of the training set, Validation subset: 30 % of the training set
Model Creator	Accuracy Maximizing Thresholds (the solutions with the classification threshold that maximizes the percentage success in the training set are returned)

The parameter values were selected as those that provide solutions with the highest classification performances (AUC, considering the filtered solutions) when the BPMs are evaluated on their test sets. To obtain high-performance BPMs, basic arithmetic functions were sufficient, which also facilitates the interpretability (at least readability) of the prediction models. The fitness/validation split shown in Table 3 (the subsets of fitness and validation need not be disjoint) was also chosen experimentally with the same AUC evaluation and after considering other alternatives with disjoint subsets. This experimental tuning was performed considering the total set of BPMs, so the GP parameter set is common to all BPMs.

Table 4. AUC comparison with and without dimensionality reduction with the best FS method in each BPM and using financial variables from set A. The highest values in terms of maximum, minimum and average AUC of the 5 best solutions are highlighted in bold for each model

	With dimensionality reduction				Without dimensionality reduction		
BPM	Selected variables	Maximum AUC (test)	Minimum AUC (test)	Average AUC (test)	Maximum AUC (test)	Minimum AUC (test)	Average AUC (test)
M1	$GPFS_1$	**94.29%**	**93.62%**	**93.86%**	93.91%	93.19%	93.64%
M2	$GPFS_{22}$	**91.53%**	**90.88%**	**91.11%**	90.65%	90.22%	90.44%
M3	$GPFS_{23}$	85.60%	**85.49%**	**85.54%**	**85.64%**	84.79%	85.10%
M4	$GPFS_{23}$	**85.78%**	**85.51%**	**85.64%**	85.61%	84.89%	85.23%
M5	$GPFS_{23}$	**80.84%**	**80.31%**	**80.54%**	80.45%	79.56%	79.83%
M6	$GPFS_1$	**79.02%**	**78.60%**	**78.78%**	78.98%	78.21%	78.50%
M7	$GPFS_{21}$	**76.08%**	**74.26%**	**75.12%**	74.00%	73.71%	73.81%
M8	$GPFS_1$	**67.39%**	**66.77%**	**67.05%**	66.25%	65.18%	65.81%
M9	$GPFS_1$	**70.21%**	**68.27%**	**68.91%**	67.93%	67.37%	67.60%
M1-3	$GPFS_1$	**89.46%**	**89.21%**	**89.34%**	89.10%	88.67%	88.91%
M1-6	$GPFS_1$	**83.93%**	**83.74%**	**83.79%**	83.56%	83.29%	83.45%
M1-9	$GPFS_1$	**80.12%**	**79.90%**	**80.04%**	79.37%	79.23%	79.28%
M4-6	$GPFS_{23}$	**81.70%**	**81.45%**	**81.56%**	81.63%	81.34%	81.47%
M7-9	$GPFS_{23}$	**73.51%**	**72.73%**	**73.03%**	73.12%	72.44%	72.62%

3. Results

3.1. Results with Set A Using GP as a Classifier

First, we compare the different GP-based FS methods explained above. The comparison can be performed, for example, considering the best AUC (over the test set and with the filtered solutions) obtained with the evolved solutions. That is, the best-evolved solutions of the independent GP runs when GP is used as a classifier, after the previous step of using GP as an FS method. However, we considered the average AUC of the 5 best solutions, in order to determine if an FS method can obtain recurrently good solutions in the different independent GP runs considered in the FS process.

Table 4 shows the best FS methods in each of the BPMs (i.e., the method with the best average AUC) and the maximum/minimum/average AUC of the 5 best solutions of the best FS method. Table 4 also includes the same data before dimensionality reduction, i.e., the evolved BPMs can use the full set of input variables.

The highest values in terms of maximum, minimum and average AUC of the 5 best solutions are highlighted in bold for each model. As can be seen and as might be expected, dimensionality reduction improves the results with respect to the use of the total number of input variables in all aspects (maximum, minimum and average AUC), with the only exception of the maximum AUC of M3. Moreover, $GPFS_{1,}$ with its

dimensionality reduction based on p-value<0.05, shows better results than other alternatives ($GPFS_{21}$, $GPFS_{22}$ or $GPFS_{23}$), but not in all BPMs (7 out of 14 models), followed by $GPFS_{23}$.

Table 5. AUC comparison with and without dimensionality reduction with the best FS method in each BPM and using variables from set B. The highest values in terms of maximum, minimum and average AUC of the 5 best solutions are highlighted in bold for each model

	With dimensionality reduction				Without dimensionality reduction		
BPM	Selected variables	Maximum AUC (test)	Minimum AUC (test)	Average AUC (test)	Maximum AUC (test)	Minimum AUC (test)	Average AUC (test)
M1	$GPFS_{23}$	**95.77%**	**95.19%**	**95.44%**	95.25%	94.40%	94.69%
M2	$GPFS_{1}$	**91.21%**	**90.76%**	**90.94%**	90.64%	89.66%	90.11%
M3	$GPFS_{1}$	**85.38%**	**84.35%**	**84.86%**	84.20%	83.83%	83.97%
M4	$GPFS_{23}$	**86.17%**	**85.40%**	**85.72%**	85.85%	84.85%	85.24%
M5	$GPFS_{1}$	**80.54%**	**80.14%**	**80.29%**	79.67%	79.11%	79.42%
M6	$GPFS_{97+15}$	**78.59%**	**78.11%**	**78.33%**	78.32%	77.97%	78.21%
M7	$GPFS_{23}$	**74.64%**	**73.40%**	**73.85%**	72.79%	72.22%	72.55%
M8	$GPFS_{22}$	**73.37%**	**72.82%**	**72.96%**	70.99%	70.51%	70.73%
M9	$GPFS_{1}$	**72.78%**	**71.83%**	**72.12%**	71.28%	70.33%	70.71%
M1-3	$GPFS_{97+15}$	89.39%	**89.10%**	**89.21%**	**89.44%**	88.57%	88.86%
M1-6	$GPFS_{23}$	**84.04%**	**83.83%**	**83.91%**	84.03%	83.72%	83.85%
M1-9	$GPFS_{1}$	**80.11%**	**80.01%**	**80.07%**	79.70%	79.42%	79.51%
M4-6	$GPFS_{1}$	**82.03%**	**81.95%**	**81.97%**	81.55%	81.20%	81.30%
M7-9	$GPFS_{1}$	**75.70%**	**74.85%**	**75.16%**	74.42%	73.96%	74.12%

3.2. Results with Set B Using GP as a Classifier

Table 5 shows the same information as in the previous table when the explanatory variables correspond to set B. Now, the different GPFS alternatives can select among 112 explanatory variables (97 from set A plus 15 additional variables defining set B). An additional possibility was included in the analysis: on top of the previous dimensionality reduction (with set A), the feature set corresponding to the best reduction method applied to set A (FS method specified in Table 4), increased by the 15 variables that differentiate sets A and B, possibility labeled "$GPFS_{97+15}$".

Several comments can be made from the results shown in Table 5. First, dimensionality reduction again improves the results with respect to the use of the total number of variables, as can be seen with the best values obtained in the maximum, minimum and average AUC (highlighted in bold for each model). Only in M1-3 the maximum AUC is slightly lower with dimensionality reduction than with the total variables. Secondly, for set B, again dimensionality reduction based on p-value<0.05 ($GPFS_1$) shows better results, but not in a generalized way (7 of 14 BPMs).

Another aspect to comment on is that, in only BPMs M6 and M1-3 the best method is $GPFS_{97+15}$ (adding the 15 extra variables to the best set of variables selected from set A). That is, the extra 15 variables are not enough to obtain better results than those provided by the FS from set A (see maximum values in Table 4 for M6 and M1-3). On the other hand, the differences in AUC are not evident except in M8, with clearly higher AUC results using set B as the initial variables to be selected. The comparative results between Tables 4 and 5 are summarized in Table 6. Each cell in Table 6 specifies whether the best AUC values (over the test set, with respect to the maximum, minimum and average values) are obtained using set A or set B with the corresponding best evolved BPM (AUC values shown in Tables 4 and 5).

It is observed (Table 6) that the situation is one of total equilibrium when deciding on the best initial set of explanatory variables. For example, in the case of taking into account the BPMs evolved after dimensionalty reduction, considering the maximum and minimum AUC, in 7 of the 14 models, the dimensionality reduction applied on the set of financial variables of the company (set A) presents better results than when the reduction is applied on the total of variables of the company and of the economic environment (set B). Considering the average AUC, the use of set B of possible explanatory variables to be selected also presents better results (8 out of 14 cases).

The same analysis, considering the best-performing BPMs without dimensionality reduction, provides similar conclusions. For example, considering the maximum and average AUC values, in 8 out of 14 cases (maximum AUC) and 7 out of 14 cases (average AUC), set B offers better results in terms of AUC. Consequently, the dilemma about the use, in the initial set of input variables, of variables other than financial variables is not clearly resolved, as previously analyzed in [8].

Table 6. Comparison of results between using set A or B on the evolved BPMs. Each cell corresponds to the set that provides the best AUC results on the best-evolved BPM (values shown in Tables 4 and 5)

	With dimensionality reduction			Without dimensionality reduction		
BPM	Maximum AUC (test)	Minimum AUC (test)	Average AUC (test)	Maximum AUC (test)	Minimum AUC (test)	Average AUC (test)
M1	B	B	B	B	B	B
M2	A	A	A	A	A	A
M3	A	A	A	A	A	A
M4	B	A	B	B	A	B
M5	A	A	A	A	A	A
M6	A	A	A	A	A	A
M7	A	A	A	A	A	A
M8	B	B	B	B	B	B
M9	B	B	B	B	B	B
M1-3	A	A	A	B	A	A
M1-6	B	B	B	B	B	B
M1-9	A	B	B	B	B	B
M4-6	B	B	B	A	A	A
M7-9	B	B	B	B	B	B

3.3. Comparison of GP as a Classifier with other Classifiers Adjusted with AutoWeka

To check the capability of GP as a classifier, its results (with the best GPFS method), are compared with different classification methods. Our previous study in [7] performed a comparison of GP as a classifier, using $GPFS_1$ as the FS method (and other widely used FS methods), with a large set of standard classification methods (including ensembles of classifiers), with the standard values in the defining parameters of the classifiers and using the variables selected by $GPFS_1$. However, in order to make the conclusions from the analysis of GP as a classifier more robust, a more thorough comparison can be made with other ML classifiers where there is now an automatic tuning process for their defining parameters. For this purpose, the Weka package called AutoWeka [24] was used, which provides an automatic adjustment of the parameters in each ML method for correct comparison. Automated Machine Learning (AutoML) is the comprehensive automation of machine learning model building tasks, with AutoWeka being one of the leading software packages for its implementation. AutoML is widely used for classification and regression in sectors such as healthcare, finance, marketing and natural language processing. AutoML has also been used in predicting business failure [31, 33, 37].

AutoWeka is a free and open-source software package that automates algorithm selection and hyperparameter settings for each classification method in the learning process. Its main functions are: i) Algorithm self-selection: AutoWeka automatically evaluates a wide range of ML algorithms and selects the most suitable one for a specific data set; ii) Hyperparameter optimization: AutoWeka searches for the optimal hyperparameter settings of the selected algorithm to improve model performance (including FS); iii) Fully automated approach: No user intervention is required for algorithm selection or hyperparameter settings. AutoWeka configuration parameters are minimal, the most relevant being the time limit to perform the process.

To perform the comparison, AutoWeka is supplied with the same training/test sets and the same sets (A and B) of explanatory variables used with GP. The annual BPMs were considered in the comparison. For each of the annual BPMs, AutoWeka was run with different time limits (15, 30, 60, 120, 180 and 240 minutes, in a platform with 24 GB of RAM and Intel i7-7700 processor, with 240 minutes being the heuristic time limit to obtain the best results in a generalized way). For each temporal prediction horizon (1, 2, ...9 years prior to failure) the best result of the different AutoWeka runs (15, 30, ..., 240 minutes) is chosen. The metric used for comparison is again the AUC calculated on the test set. Note that AutoWeka also performs an FS process to determine the most relevant variables to be used in each classifier.

In the case of GP, HeuristicLab allows an automatic simplification and manual pruning of the evolved solutions (oriented to improve AUC over the test set, a process that can also eliminate bloat in the evolved trees). The results of the best-evolved BPMs before such processes are shown in Tables 4 and 5, corresponding to the maximum values of the 5 best GP-evolved solutions. Table 7 shows the AUC values after applying the HL simplification and pruning to those best annual BPMs obtained after dimensionality reduction. As an example, Figure 1 includes the evolved trees of M5 (medium-long term prediction), with the evolved tree (with the best AUC) before and after simplification and pruning. The evolved trees are shown in a hierarchical

representation provided by HL, corresponding to a mathematical expression that uses the basic arithmetic operations (specified in Table 3) on the nodes of the tree to process the input information of the selected explanatory variables. The input variables are labeled in HL as *log_X*, where *log* refers to the logistic transformation discussed in Section 2.1 and *X* refers to the index of the input variable (out of the 97 in set A where the best M5 model is obtained). Table 8 includes a brief definition of these input variables used in this prediction model. As can be seen, the BPM M5 uses a considerable number of explanatory variables. At this stage, it is not possible to know the relevance of each of them in the value estimated by the model, but it is possible to note (in a very general way) two main focuses of interest: the profitability (variables 13, 37, 43, 66, 82) and the indebtedness (variables 24, 27, 59, 69, 77), which are "a priori" consistent with the prediction horizon of the model.

Table 7 also shows, for each set of explanatory variables, the AUC corresponding to the best solution found by AutoWeka for each BPM (the best considering the different time limits). The meaning of the AutoWeka classifier acronyms is as follows [24]: MLP (Multilayer Perceptron), LWL (Locally Weigththted Learning), LMT (Logistic Model Tree), Ibk (Instance-based k-neighbors), SGD (Stochastic Gradient Descent) and SMO (Sequential Minimal Optimization - support vector classification).

Table 7. Best AUC values, evaluated on the test set of annual BPMs obtained with GP (after simplification and pruning) and those obtained with AutoWeka. GP (as a classifier) uses the best feature subset selected with the GPFS methods. The values in bold correspond to the best AUC values in each of the sets of possible explanatory variables (A and B). The values in filled gray correspond to the best AUC values in the comparison between sets A and B

	AUC (test set) - set A			AUC (test set) - set B		
BPM	GP (after simplification and pruning)	AutoWeka	AutoWeka Classifier	GP (after simplification and pruning)	AutoWeka	AutoWeka Classifier
M1	**95.02%**	91.80%	MLP	**95.94%**	91.30%	lazy-LWL
M2	**91.78%**	90.20%	lazy-LWL	**92.40%**	89.10%	RandomForest
M3	**86.60%**	84.80%	lazy-LWL	**85.64%**	83.50%	Bagging
M4	**86.44%**	83.50%	LMT	**87.30%**	83.60%	SimpleLogistic
M5	**82.10%**	80.10%	lazy-Ibk	81.30%	**81.60%**	RandomForest
M6	**80.02%**	75.40%	SGD	**78.95%**	76.60%	Bagging
M7	**76.18%**	72.80%	Bagging	**75.78%**	72.00%	NaiveBayes
M8	**71.75%**	63.30%	lazy-LWL	**74.91%**	63.20%	SMO
M9	**71.41%**	65.70%	SGD	**73.89%**	68.90%	MLP

It is observed that the models obtained with GP offer better performance than those obtained with other ML classifiers. Whether only financial variables (set A) or financial and non-financial variables are used (set B), for all the temporal horizons of the prediction models, the AUC evaluated on the test set of the models obtained with GP is superior to that obtained with AutoWeka, with considerable increases in AUC in some cases, as in M8 (8.45% AUC increase with set A and 11.71% with set B). The only exception of M5 with set B.

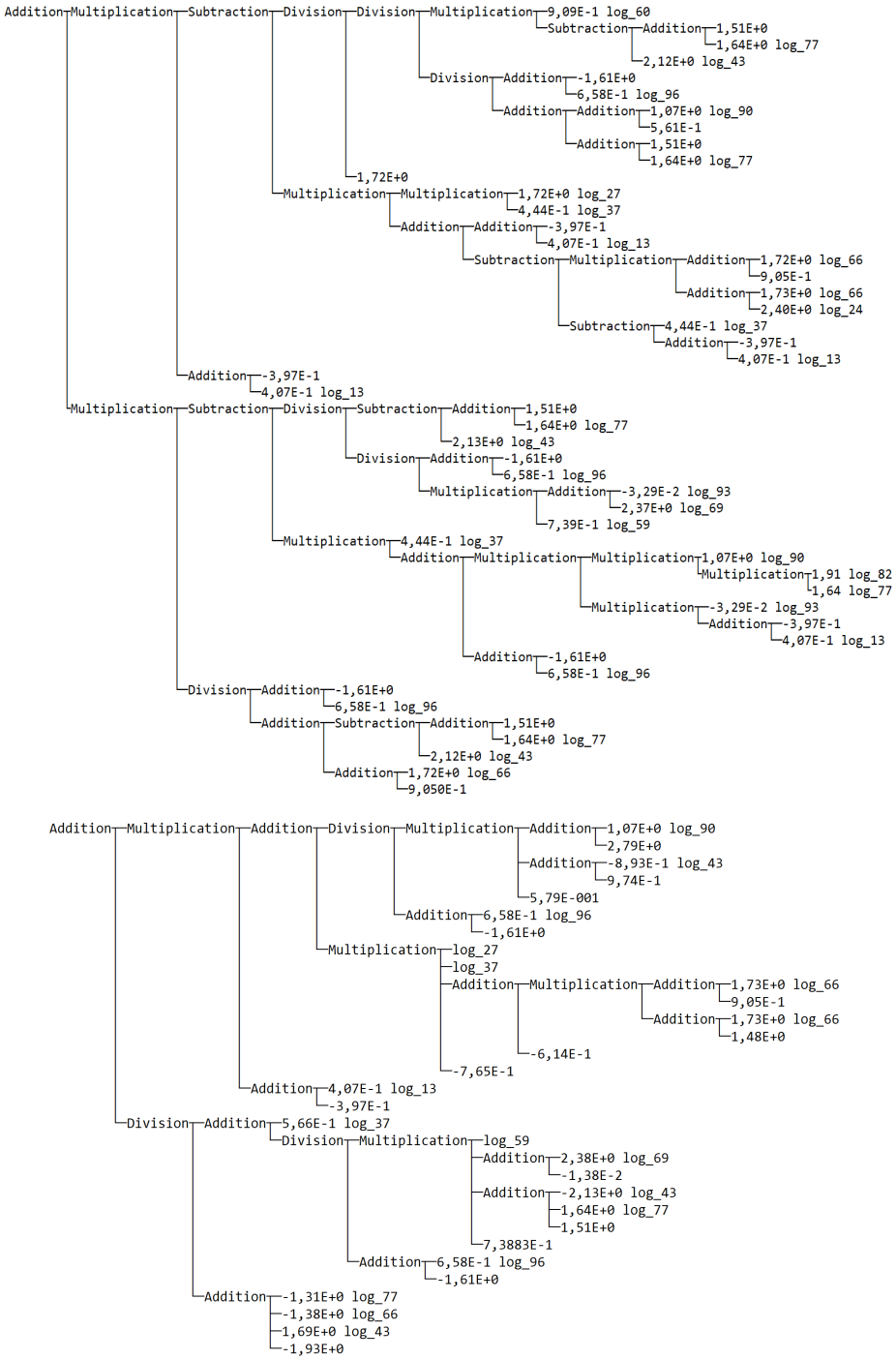


Fig. 1. Best evolved BPM M5, before (upper figure) and after (bottom figure) the simplification and pruning process

The comparison of computational time should also be taken into account. In this comparison, as a reference, each independent run of GP implies an average computation time of 76 seconds, using a model with an average training set (282 observations) and evaluating 150,000 solutions per independent run (population size=1,500, Generations=100, Table 3). That is, the 1,000 independent runs involve 1,267 minutes.

In the case of AutoWeka, as stated, 240 minutes was the heuristic time limit to obtain the best AUC values in a generalized way. For example, doubling this time to 480 minutes (representing an average of 7,500 configurations tested by AutoWeka), when using set A, the improvement in the percentage of misclassified instances in the training set is generalized in the nine BPMs (5 of them present an error of 0 in the classification of the training set). However, only one of the nine AUCs evaluated on the test set, with 480 minutes, improves (by less than 2%) on those shown in Table 7.

It is noteworthy that, even with the BPMs evolved before simplification and pruning, the AUC values (shown in Tables 4 and 5, maximum AUC) in practically all BPMs are better compared to those obtained by the classifiers selected by AutoWeka (the only exception is again M5 with set B). Consequently, the combination of the best FS method based on GP together with GP as a classifier improves the results of the other ML methods, even with the automatic adjustment provided by AutoWeka in those other ML methods.

Table 8. Variables of evolved BPM M5

Variable	Definition
13	EBITDA / Interest expenses
24	Total debt / Total assets
27	(Current liabilities - Cash) / Total assets
37	Net income / Shareholder funds
43	Operating ROA
59	Accounts payable / Total sales
60	Inventory / Total sales
66	Interest expense percent of EBITDA
69	Interest expense / Total debt
77	Change in financial debts
82	Variation *t* over *t-1* in purchases / Variation t over *t-1* in accounts payable
90	LN (Total assets) squared
93	Degree of assets decomposition
96	Number of employees

4. Conclusions and Future Work

In this paper, a novel feature selection approach based on genetic programming was proposed and defined, with an iterative process of eliminating irrelevant variables, as opposed to a previous approach that directly selects relevant variables based on the statistical significance of their appearance in the evolved GP solutions. The novel proposal maintains the statistical significance to select the variables, but with a more lax

criterion and with its repetition in an iterative process to reduce the possibility of removing possible relevant variables that could be discarded with a strict selection of variables.

The comparison of the results has been performed considering a large number of BPMs (annual and multiperiod) with different prediction horizons, as well as considering two sets (A and B) of explanatory variables in the BPMs. Several conclusions can be drawn from the experimentation with both FS strategies and both sets of explanatory variables:

i. First, the results show the usefulness of the proposed FS methods, since their use improves BPM performance in practically all cases with respect to the use of the whole set of input variables. It also shows that the proposed FS methods manage to automatically select the most relevant variables for each particular BPM, highlighting that the proposed FS methods automatically determine the number of explanatory variables to be taken into account. Moreover, this selection is performed in a context-sensitive manner (i.e., considering the relationship with the other selected variables).

ii. Secondly, the comparison of results, with both FS strategies and using GP as classifier, shows the robustness of the approach that directly selects the relevant variables based on their statistical significance ($GPFS_1$, variables selected with p-value<0.05), since it is the FS method with better results in more evolved BPMs. However, in several cases, $GPFS_1$ was outperformed by the alternative iterative FS method, which means that some relevant variables can be discarded with strict selection (p-value<0.05). The FS methods can be further developed, possibly by including, in addition to the p-value, some other indicator (such as mean or median) of relative frequency and/or by modifying the established p-value limits.

iii. Third, the comparison of the results between the use of sets A and B of explanatory variables (the latter incorporating variables from the economic environment) does not provide conclusive results on the dilemma of whether the use of variables other than financial variables is necessary to improve the performance of predictive models. It can be hypothesized that, in our case, the standardization of variables with the logistic transformation implies that, indirectly, the effect of the economic environment is incorporated in the variables considered in set A (only financial variables), so that set B does not provide additional information to improve the classification performance.

Furthermore, the comparison between GP-based FS approaches was performed using GP as a classifier. A comparison between GP as a classifier and other ML classifiers was also performed, with the automatic tuning (including FS) provided by AutoWeka. Another conclusion can be drawn from this comparison is:

iv. Even with AutoWeka's automatic search for the best classifier (and its parameter tuning), GP provides better performance results. As discussed in the discussion of the results of this comparison, it is noteworthy that the performance (AUC on the test set) of the BPMs that were evolved before the simplification and subsequent pruning processes (aimed at improving AUC performance) is better (in virtually all BPMs) compared to the performance of the AutoWeka classifiers. This demonstrates the synergy obtained by using GP both as a feature selector and as a classifier.

In future work, since GP-defining parameters were experimentally adjusted and are common to all the BPMs considered, an automatic adjustment of the parameters could be performed, also particularized for each BPM and in order to improve the AUC results in each BPM. In addition, an analysis of the types of variables selected (groups of financial variables), not analyzed here for each of the temporal horizons, should also be carried out.

Acknowledgements. This study was funded by the Xunta de Galicia and the European Union (European Regional Development Fund - Galicia 2021-2027 FEDER Program), with grants CITIC (ED431G 2023/01), GPC ED431B 2022/33 and GRC ED431C 2025/49, as well as by the Spanish Ministry of Science, Innovation and Universities (MICIU/AEI/10.13039/501100011033, project PID2023-148531NB-I00).

References

1. Alaka, H. A., Oyedele, L. O., Owolabi, H. A., Kumar, V., Ajayi, S.O., Akinade, O. O., Bilal, M.: Systematic Review of Bankruptcy Prediction Models: Towards a Framework for Tool Selection. Expert Systems with Applications Vol. 94, 164-184. (2018)
2. Altman, E. I.: Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. The Journal of Finance, Vol. 23, 589-609. (1968)
3. Altman. E. I., Iwanicz-Drozdowska, M., Laitinen, E., Suvas, A.: Financial and Non-Financial Variables as Long-Horizon Predictors of Bankruptcy. SSRN Electronic Journal. (2015)
4. Altman, E. I., Iwanicz-Drozdowska, M., Laitinen, E. K., Suvas, A.: A Race for Long Horizon Bankruptcy Prediction. Applied Economics, Vol. 52, 4092-4111. (2020)
5. Aziz, M. A., Dar, H. A.: Predicting Corporate Bankruptcy: Where We Stand? Corporate Governance: The International Journal of Business in Society, Vol. 6, 18-33. (2006)
6. Beade, Á., Rodríguez, M., Santos, J.: Business Failure Prediction Models with High and Stable Predictive Power Over Time Using Genetic Programming. Operational Research, Vol. 24, Article 52. (2024)
7. Beade, Á., Rodríguez, M., Santos, J.: Variable Selection in The Prediction of Business Failure Using Genetic Programming. Knowledge-Based Systems, Vol. 289, 111529. (2024)
8. Beade, Á., Rodríguez, M., Santos, J.: Genetic Programming for Feature Selection in Business Failure Prediction. Comparison of the Use of Financial Variables and Economic Environment Variables. In Proceedings of 2024 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). 1-6. (2024)
9. Beade, Á., Rodríguez, M., Santos, J.: Multiperiod Bankruptcy Prediction Models with Interpretable Single Models. Comput Econ, Vol. 64, 1357-1390. (2024)
10. Beaver, W. H., McNichols, M. F., Rhie, J-W.: Have Financial Statements Become Less Informative? Evidence from the Ability of Financial Ratios to Predict Bankruptcy. Rev Acc Stud, Vol. 10, 93-122. (2005)
11. Bellovary, J. L., Giacomino, D. E., Akers, M. D.: A Review of Bankruptcy Prediction Studies: 1930 to Present. Journal of Financial Education, Vol. 33, 1-42. (2007)
12. Das, S. R., Hanouna, P., Sarin, A.: Accounting-Based Versus Market-Based Cross-Sectional Models of CDS Spreads. Journal of Banking & Finance, Vol. 33, 719-730. (2009)
13. Dasilas, A., Rigani, A.: Machine Learning Techniques in Bankruptcy Prediction: A Systematic Literature Review. Expert Systems with Applications, Vol. 255, 124761. (2024)
14. European Banking Authority: Discussion Paper on Machine Learning for IRB Models - EBA/DP/2021/04. (2021)

15. European Commission: Proposal for a Regulation Laying Down Harmonised Rules on Artificial Intelligence Shaping Europe's Digital Future (2021). Available: https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence.
16. European Parliament: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance) - EU Law in Force - Publications Office of the EU. In: EU Law in Force (2016). Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng
17. García, V., Marqués, A. I., Sánchez, J. S.: Exploring the Synergetic Effects of Sample Types on the Performance of Ensembles for Credit Risk and Corporate Bankruptcy Prediction. Information Fusion, Vol. 47, 88-101. (2019)
18. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene Selection for Cancer Classification Using Support Vector Machines. Machine Learning, Vol. 46, 389-422. (2002)
19. Hosaka, T.: Bankruptcy Prediction Using Imaged Financial Ratios and Convolutional Neural Networks. Expert Syst Appl, Vol. 117, 287-299. (2019)
20. du Jardin, P.: Dynamics of Firm Financial Evolution and Bankruptcy Prediction. Expert Systems with Applications, Vol. 75, 25-43. (2017)
21. du Jardin, P.: Dynamic Self-Organizing Feature Map-Based Models Applied to Bankruptcy Prediction. Decision Support Systems, Vol. 147, 113576. (2021)
22. du Jardin, P., Séverin, E.: Forecasting Financial Failure Using a Kohonen Map: A Comparative Study to Improve Model Stability Over Time. European Journal of Operational Research, Vol. 221, 378-396. (2012)
23. Jović, A., Brkić, K., Bogunović, N.: A Review of Feature Selection Methods with Applications. In Proceedings of 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 1200-1205. (2015)
24. Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., Leyton-Brown, K.: Auto-Weka: Automatic Model Selection and Hyperparameter Optimization in Weka. In Hutter, F., Kotthoff. L., Vanschoren. J. (eds.): Automated Machine Learning: Methods, Systems, Challenges. Springer International Publishing, Cham, 81-95. (2019)
25. Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge, MA, USA. (1992)
26. Laitinen, E. K.: Financial Ratios and Different Failure Processes. Journal of Business Finance & Accounting, Vol. 18, 649-673. (1991)
27. Laitinen, E. K., Camacho-Miñano, M-M., Muñoz-Izquierdo, N.: A Review of the Limitations of Financial Failure Prediction Research. Revista de Contabilidad - Spanish Accounting Review, Vol. 26, 255-273. (2023)
28. Liu, G., Ma, J., Hu, T., Gao, X.: A Feature Selection Method with Feature Ranking Using Genetic Programming. Connection Science, Vol. 34, 1146-1168. (2022)
29. Lombardo, G., Bertogalli, A., Consoli, S., Reforgiato Recupero, D.: Natural Language Processing and Deep Learning for Bankruptcy Prediction: An End-to-End Architecture. IEEE Access, Vol. 12, 151075-151091. (2024)
30. Mai, F., Tian, S., Lee, C., Ma, L.: Deep Learning Models for Bankruptcy Prediction Using Textual Disclosures. Eur J Oper Res, Vol. 274, 743-758. (2019)
31. Mohan, B., Badra, J.: A Novel Automated Superlearner Using a Genetic Algorithm-Based Hyperparameter Optimization. Advances in Engineering Software, Vol. 175, 103358. (2023)
32. Muslim, M. A., Dasril, Y.: Company Bankruptcy Prediction Framework Based on the Most Influential Features Using XGBoost and Stacking Ensemble Learning. International Journal of Electrical and Computer Engineering (IJECE), Vol. 11, 5549-5557. (2021)
33. Nazareth, N., Ramana Reddy, Y. V.: Financial Applications of Machine Learning: A Literature Review. Expert Systems with Applications, Vol. 219, 119640. (2023)

34. Neshatian, K., Zhang, M.: Using Genetic Programming for Context-Sensitive Feature Scoring in Classification Problems. Connection Science, Vol. 23, 183-207. (2011)
35. Nguyen, H. H., Viviani, J-L., Ben Jabeur, S.: Bankruptcy Prediction Using Machine Learning and Shapley Additive Explanations. Rev Quant Finan Acc., Vol. 65, 107-148 (2023)
36. Nyitrai, T., Virág, M.: The Effects of Handling Outliers on the Performance of Bankruptcy Prediction Models. Socio-Economic Planning Sciences, Vol. 67, 34-42. (2019)
37. Papík, M., Papíková, L.: The Possibilities of Using AutoML in Bankruptcy Prediction: Case of Slovakia. Technological Forecasting and Social Change, Vol. 215, 124098. (2025)
38. Pawełek, B., Pociecha, J., Kostrzewska, J., Baryła, M., Lipieta, A.: Problem of Outliers in Corporate Bankruptcy Prediction. Repository KITopen (2017) Available: https://publikationen.bibliothek.kit.edu/1000069937
39. Poli, R., Langdon, W. B. (William B), McPhee, N. F., Koza, J. R.: A Field Guide to Genetic Programming. Lulu Press. Available: lulu.com (2008)
40. Ratajczak, P., Szutowski, D., Szulczewska-Remi, A.: Long-Term Bankruptcy Prediction. Systematic Literature Review. Research in Corporate Finance (2022) Available: http://dx.doi.org/10.2139/ssrn.4054665.
41. Sathyanarayana, N., Narayanan, R.: A Systematic Review of Models for the Prediction of Corporate Insolvency. Salud, Ciencia y Tecnología - Serie de Conferencias, Vol. 3, 952-952. (2024)
42. Suárez, R. R., Valencia-Ramírez, J. M., Graff, M.: Genetic Programming as a Feature Selection Algorithm. In Proceedings of 2014 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC). 1-5. (2014)
43. Szántó, T. K.: Handling Outliers in Bankruptcy Prediction Models Based on Logistic Regression. Public Finance Quarterly, Vol. 69, No. 3. (2023)
44. Tian, S., Yu, Y.: Financial Ratios and Bankruptcy Predictions: An International Evidence. International Review of Economics & Finance, Vol. 51, 510-526. (2017)
45. Tian, S., Yu, Y., Guo, H.: Variable Selection and Corporate Bankruptcy Forecasts. Journal of Banking & Finance, Vol. 52, 89-100. (2015)
46. Tsai, C-F., Sue, K-L., Hu, Y-H., Chiu, A.: Combining Feature Selection, Instance Selection, and Ensemble Classification Techniques for Improved Financial Distress Prediction. Journal of Business Research, Vol. 130, 200-209. (2021)
47. Vellamcheti, S., Singh, P.: Class Imbalance Deep Learning for Bankruptcy Prediction. In Proceedings of First International Conference on Power, Control and Computing Technologies (ICPC2T). 421-425. (2020)
48. Volkov, A., Benoit, D. F., Van den Poel, D.: Incorporating Sequential Information in Bankruptcy Prediction with Predictors Based on Markov for Discrimination. Decision Support Systems, Vol. 98, 59-68. (2017)
49. Wagner, S., Kronberger, G., Beham, A., Kommenda, M., Scheibenpflug, A., Pitzer, E., Vonolfen, S., Kofler, M., Winkler, S., Dorfer, V., Affenzeller, M.: Architecture and Design of the HeuristicLab Optimization Environment. In: Klempous, R., Nikodem, J., Jacak, W., Chaczko, Z. (eds.) Advanced Methods and Applications in Computational Intelligence. Springer, 197-261. (2014)
50. Zhao, J., Ouenniche, J., De Smedt, J.: Survey, Classification and Critical Analysis of the Literature on Corporate Bankruptcy and Financial Distress Prediction. Machine Learning with Applications, Vol. 15, 100527. (2024)
51. Zoričák, M., Gnip, P., Drotár, P., Gazda, V.: Bankruptcy Prediction for Small- and Medium-Sized Companies Using Severely Imbalanced Datasets. Economic Modelling, Vol. 84, 165-176. (2020)
52. SABI database. Available: https://www.einforma.com/soluciones-y-herramientas/sabi
53. HeuristicLab. Available: https://dev.heuristiclab.com/trac.fcgi/

Ángel Antonio Beade Torreiro defended his PhD thesis "Predicting business failure using Genetic Programming" in 2024 at the University of A Coruña, Spain.

José Santos Reyes is Professor in the Department of Computer Science and Information Technologies at the University of A Coruña, CITIC (Centre for Information and Communications Technology Research), Spain. His research interests include artificial life, neural computation, evolutionary computation, autonomous robotics and computational biology.

Manuel López is a Chair Director at Collections and Payments Manager at ABANCA, and adjunct professor at the University of A Coruña, Spain.

Received: February 25, 2015; Accepted: December 01, 2025.

Computer Science and Information Systems 23(1):535–560 https://doi.org/10.2298/CSIS250416082T

Enhancing Network Engineering Capabilities through LLM Fine-Tuning with Automatically Generated Datasets ⋆

Claudiu Trăistaru[1], Florin Pop[2], Costin Bădică[3], Cătălina Mancaş[3], and Ionuţ Murareţu[3]

[1] University of Science and Technology POLITEHNICA Bucharest, University of Craiova, Craiova, Romania
claudiu.traistaru@edu.ucv.ro
[2] University of Science and Technology POLITEHNICA Bucharest, National Institute of Research and Development in Informatics (ICI), and Academy of Romanian Scientists, Bucharest, Romania
florin.pop@upb.ro
[3] Computer and Information Technology Department, University of Craiova, Craiova, Romania
{costin.badica,catalina.mancas, ionut.muraretu}@edu.ucv.ro

Abstract. The paper presents a method for automatically generating domain-specific datasets to fine-tune open-source LLMs in network engineering. Our objective is to address the increasingly complex nature of network configuration and management jobs by supplying LLMs with high-quality training data. We evaluated datasets generated using open-source LLMs, including DeepSeek-R1 671B, LLaMA 3.1 70B, Qwen 2.5 72B, and Mixtral 8x7B, analyzing the quality of unprocessed knowledge data and the efficacy of cleaning and deduplication methods. The resulting dataset addresses various subjects related to routing, security, and network services. Afterward, we fine-tuned smaller LLaMA 3.2 1B, LLaMA 3.2 3B and Qwen 2.5 1.5B models using Low-Rank Adaptation, thereby minimizing computational demands while maintaining the quality of domain knowledge.

1. Introduction

Network engineers are essential to the management and functioning of computer networks, as their choices directly impact the transmission and security of data within these complex systems. Historically, computer networks have facilitated diverse functions, ranging from basic communications like emailing to intricate processes such as cloud computing, online gaming, and the seamless integration of Internet of Things (IoT) devices. The main goal of these systems is to efficiently direct data traffic from source to destination, which is accomplished by properly setting up network devices and services.

Various network components, such as switches, routers, network interfaces, and network services, are part of the configuration process. Current computer networks have grown more complex, requiring a deep understanding of topologies, protocols, and the specific demands of each network. To guarantee strong data security, network engineers must deploy and maintain essential tools, including intrusion detection systems, firewalls,

⋆ This paper is a significant extension of our preliminary work presented in [22]

and safe Access Control Lists (ACLs). Recent research shows how natural language processing (NLP) can be used to automate and validate network designs by employing network parsers as specialized tools that convert network configurations into structured intermediate representations such as Abstract Syntax Trees [2].

The complexity of these systems demands novel strategies for network management. Expert systems supported by domain-specific Large Language Models (LLMs), especially open-source variants, are becoming crucial for assisting network engineers in network engineering problem solving like for example implementing automated network settings and improving network management. LLM-based approaches could allow network engineers to address the increasing difficulties of performance optimization, cybersecurity, and IoT integration while minimizing the manual burden usually linked to network configuration and management.

In this context, tuning LLMs using high-quality, domain-specific datasets proves crucial for enhancing their effectiveness in network engineering. The goal of our research effort is to develop new methodologies for the efficient creation of such datasets by leveraging existing domain-specific literature and synthetic data generation techniques. These datasets are usually structured as sets of pairs comprising questions and answers related to problem solving in network engineering. The goal of our work is to enhance LLM applications in network administration, contributing to the advancement of computer networks to ensure their security, efficiency, and ability to satisfy the requirements of a progressively networked world.

According to [4], Reinforcement Learning with Human Feedback (RLHF) is a machine learning (ML) technique that leverages human feedback to improve ML models so that they can learn more efficiently. RLHF uses rewards based on human feedback so that the model can perform tasks better aligned with human goals. RHLF is a standard technique that ensures that LLMs produce truthful, helpful and harmful content. Our research aims to create network engineering datasets by minimizing the need for RLHF, which is costly, time-consuming and error-prone. The need to use generated datasets to fine-tune LLMs for more specific fields is growing, and being able to generate this kind of dataset is our solution to this need.

We envisage a network engineering assistant acting like a human expert in the field. To develop such an expert assistant in network engineering using publicly available models (i.e., whose weights can be freely downloaded and fine-tuned), we will investigate the feasibility of generating training datasets based on a sizable corpus of texts and also the use of these datasets to fine-tune smaller versions of open-weight models, such as LLaMA 1B and LLaMA 3B.

For example, the expert assistant can assist network engineers in designing configurations of switches and routers that control the routes along which data packets travel over the network, guaranteeing effective and safe data transfer. Among other things, these configurations must consider fault tolerance, traffic load distribution, and network topology. Modern networks can virtualize network functions to provide services like firewalls, load balancers, or intrusion detection systems, but these must be correctly set up.

The development of open-source LLM models has significantly fueled the availability of advanced artificial intelligence technologies, stimulating innovation and fostering diversity across various sectors. Unlike their proprietary equivalents, these models allow the scientific community and developers to investigate, alter, and enhance them. Open-source

LLMs boost research and development and allow us to improve them to achieve better results.

In the quest for Artificial General Intelligence (AGI), the growing availability of both closed and open-source LLMs represents a significant turning point. LLMs have progressed gradually toward reaching AGI by passing through several evolutionary stages. This development has resulted in abundant publicly available and proprietary LLMs, adding to the diverse environment of AI-driven language creation. In this paper, we propose a new method to automatically produce a varied and relevant dataset that can be used to fine-tune open-source LLMs to increase their performance in the field of network engineering.

Our study investigates how open-source models may be improved by fine-tuning using specialized data previously selected and organized into distinct, well-chosen and well-designed datasets.

We extended the data generation methodology presented in [22] by the usage of the following models: DeepSeek-R1 671B, LLaMA 3.1 70B, Qwen 2.5 72B, and Mixtral 8x7B, representing the junction of innovation, accessibility, and adaptation. By tackling fundamental difficulties like computing efficiency, scalability, and domain-specific adaptation, these models enable academics and practitioners to create transformational NLP applications. As the field evolves, the contributions of these models will indeed affect the course of future improvements, ensuring that the advantages of LLMs are both broad and inclusive. The iterative nature of their creation assures that these models are always current, constantly evolving in response to new challenges and possibilities. Afterward, we used the cleaned generated data to fine-tune LLaMA 3.2 1B, LLaMA 3.2 3B and Qwen 2.5 1.5B models.

The paper is structured as follows. In Section 2, we provide an overview of the domain presenting related work of LLM in the network engineering domain. Section 3 depicts the data collection and processing methodology, the evaluation methods and the used metrics. The experimental setup and the results are discussed in Section 4. Finally, we present some conclusions in Section 5.

2. Related Work

2.1. Open-Source Large Language Models

The Open Pre-trained Transformer (OPT) design efficiently uses transformer-based pre-training to provide coherent, contextually appropriate outputs, making it a key component in generalizable NLP applications. A model's emphasis on open-source accessibility promotes repeatability and community-driven innovations, resulting in significant gains in adaptability and applicability across diverse areas. [25].

The LLaMA [20] models aim to reduce the size of the parameter space without sacrificing performance. LLaMA produces high-quality results while being computationally accessible using powerful tokenization approaches and stringent dataset filtering. This model's capacity to thrive in zero-shot and fine-tuning scenarios demonstrates its adaptability, making it ideal for academics and institutions with limited computational resources. Furthermore, LLaMA's efficient architecture design allows it to scale successfully across applications that require dynamic adaptation, such as real-time translation

systems and adaptive content production [21]. Its parameter efficiency also makes it a good choice for inclusion in systems that require scalability and quick deployment.

Mixtral is a Sparse Mixture of Experts (SMoE) language model created by Mistral AI. The design comprises eight unique expert networks, amounting to 47 billion parameters, of which only 13 billion are utilized during inference to enhance computational efficiency, in this case it means only 2 experts are selected per token during inference, minimizing the memory usage.

This approach enables Mixtral to surpass models such as Llama 2 70B and GPT-3.5 across multiple benchmarks, particularly excelling in mathematics, code creation, and multilingual activities. [9].

Phi-4 is a small language model (SLM) with 14 billion parameters, developed by Microsoft, intended to excel in intricate reasoning tasks, including mathematics and coding. In contrast to numerous models that predominantly depend on organic data sources, Phi-4 deliberately integrates synthetic data during its training, resulting in improved performance in STEM-oriented question-answering abilities. Significantly, Phi-4 exceeds its instructor model in these domains, illustrating the efficacy of its data-generation and post-training methodologies. [1].

DeepSeek-R1 is an open-source large language model created by DeepSeek, a Chinese artificial intelligence firm. Launched in January 2025 under the MIT License, it is engineered to excel at logical inference, mathematical reasoning, and real-time problem-solving tasks. DeepSeek-R1 utilizes reinforcement learning methodologies to augment its reasoning abilities. Significantly, it was created at a substantially lower cost than its rivals, highlighting China's increasing competitiveness in AI development. [5].

Qwen 2.5 is an extensive collection of large language models (LLMs) created by Alibaba Cloud, intended to address various requirements. Qwen 2.5 has undergone substantial enhancements in both the pre-training and post-training phases compared to earlier versions. [18].

Together, these models represent the open-source LLM ecosystem's rising diversity and specialization. Mixtral and Phi-4 address domain-specific difficulties with specialized innovations. This variety of capabilities demonstrates open-source models' ability to serve various applications, from broad, generalist tasks to highly specialized sectors. Furthermore, the community-driven collaborative mindset around these models has reduced barriers to entry for researchers globally, creating an atmosphere in which innovation is driven by shared information and collective achievement. This culture of openness has sped progress, ensuring breakthroughs are quickly communicated and implemented in academic and industry settings.

High-performing models have sparked a global research movement, democratizing cutting-edge NLP technology. The repeatability and accessibility of open-source models have opened up new avenues for research, resulting in achievements in fields ranging from computational linguistics to real-time machine learning applications. These models offer a solid framework for multidisciplinary research, enabling applications in social science, legal analysis, and biological research [25].

2.2. Role of Datasets in Fine-Tuning

Datasets are necessary for fine-tuning LLMs, acting as the key connection between their extensive pretraining and the specific requirements of target applications. Pretraining on

extensive, general-purpose corpora equips open-source models like LLama, Qwen, and Phi with exceptional generalization abilities. At the same time, fine-tuning facilitates their adaptation to domain-specific tasks, enabling applications in healthcare, finance, and legal services.

The success rate of fine-tuning can be significantly affected by the dataset's properties, such as size, quality, and diversity. The quantity of the dataset is essential as it dictates the extent of language patterns and domain-specific knowledge that a model can assimilate. Larger datasets often enhance coverage and generalization; nevertheless, overly extensive datasets may result in redundancy, causing decreasing returns and heightened computing expenses. Achieving a good equilibrium between dataset size and task relevance is crucial for enhancing efficiency and performance.

In general, high-quality datasets are essential for ensuring that models acquire accurate and consistent patterns. Dataset quality is defined by low noise, consistent annotations, and applicability to the specific area. A well-curated dataset enables the model to concentrate on obtaining task-specific knowledge, free from the influence of extraneous or erroneous information. The preprocessing stages, including deduplication, noise filtration, and context enhancement, are essential for meeting these quality goals. The influence of datasets on fine-tuning results is significant. Well-curated datasets facilitate models in attaining enhanced accuracy, equity, and interpretability [21]. The success of LLaMA underscores the significance of dataset quality and diversity in enhancing fine-tuning effectiveness.

While the method of collecting and selecting high-quality training data is somewhat complicated, the basic ideas of creating a dataset are relatively straightforward. The first step is to choose the problems that you want our model to solve. After that, you must create a dataset demonstrating these behaviors. Many teams are increasingly designating specific responsibilities to handle the procurement of acceptable datasets while maintaining compliance with privacy standards as they recognize the importance of data.

Pre-training requires different and considerably larger datasets than instruction fine-tuning or preference fine-tuning. Even with these modifications, the fundamental concepts of dataset design, i.e., ensuring quality, obtaining acceptable coverage, and preserving sufficient quantity, stay the same across all training stages.

2.3. Challenges in Dataset Design for Fine-Tuning Large Language Models

Using RLHF to fine-tune language models demonstrates that iterative, human-centered approaches significantly enhance dataset quality and model alignment with real-world applications. This methodology leverages the value of integrating human judgment directly into the data creation pipeline [16]. Including clear and precise instructions for annotators proved instrumental in reducing inconsistencies and streamlining the annotation process. By embedding iterative feedback mechanisms, datasets can undergo continuous refinement, ensuring they remain relevant, reliable, and effective in enabling LLMs to meet specific task requirements.

A critical role of task-specific instructions and illustrative examples in guiding annotators to provide explicit objectives and contextually relevant examples ensures a shared understanding of goals. This is especially vital for complex domains, such as networking tasks, where precision and contextual sensitivity are paramount. Extending these prac-

tices to dataset design for LLM fine-tuning facilitates the creation of high-quality datasets capable of supporting nuanced and domain-specific applications.

The design of datasets for fine-tuning LLMs is a complex but indispensable process that requires addressing significant challenges, including annotation complexity and privacy compliance. By leveraging human feedback mechanisms and aligning to ethical principles [16], researchers and practitioners can create datasets that are high-quality, reliable and aligned with societal values. These advancements are essential for harnessing the full potential of LLMs in a manner that respects both technical rigor and ethical integrity.

2.4. Applications of Large Language Models in Network Engineering

Incorporating refined LLMs into network engineering signifies a fundamental change and actionable knowledge. LLMs have become essential instruments in contemporary network engineering by connecting human purpose with technological execution.

A primary use of LLMs is in automated configuration management, where their capacity to comprehend human-readable configuration requests and convert them into accurate, deployable network configurations has been transformational. [8] elucidates the capacity of LLMs to optimize network configuration procedures, significantly reducing the need for manual involvement. This not only reduces mistakes but also expedites deployment time-frames. Moreover, [15] illustrates how LLMs may generate precise and resilient router configurations from abstract, high-level requirements. These qualities are especially beneficial in extensive and diverse networks, where uniformity and scalability are essential. By generalizing across many networking circumstances, LLMs may adjust to new technologies, suggest optimum setups, and improve the overall agility of network operations.

Security diagnostics constitute another transformational area in which LLMs have demonstrated considerable potential. One approach involves using context-aware, iterative prompting to help LLMs identify and diagnose router misconfigurations [10]. Misconfigurations, a popular source of network vulnerabilities, can diminish performance or render systems susceptible to possible security breaches. LLMs can effectively discover abnormalities and possible dangers by evaluating logs, configurations, and traffic patterns. The incorporation of predictive analytics into LLM procedures facilitates the forecast of potential threats through historical data patterns, allowing for preventative measures to strengthen network defenses and proactively minimize risks.

A popular task is the creation of network management scripts. LLMs can generate task-specific scripts for functions like automatic improvement of device configurations, dynamic traffic routing, and diagnostic assessments [14]. This functionality improves operational processes by automating redundant jobs and enabling quick reactions to changing network circumstances. LLMs may dynamically modify routing pathways to mitigate bottlenecks or execute load-balancing algorithms in real time, dramatically enhancing network speed. Furthermore, by including LLMs in continuous integration and deployment (CI/CD) pipelines, developers can guarantee that created scripts undergo thorough testing and validation, minimizing downtime and improving operational dependability.

Telecommunication networks, known for their complexity and scale, have also gained advantages from the particular characteristics of LLMs. As introduced by [13], Tele-LLMs is a collection of fine-tuned models tailored for telecommunication-specific functions, including performance monitoring, problem diagnosis, and capacity planning. These

models utilize domain-specific data to detect network irregularities and forecast possible failures, enabling engineers to take proactive remedial actions. The predictive skills of Tele-LLMs encompass the optimization of 5G deployments and edge computing infrastructures, where real-time decision-making and agility are essential. By examining historical performance trends and real-time measurements, Tele-LLMs empower telecom operators to improve service dependability and efficiency.

LLMs' adaptability to various networking tasks is further demonstrated in [24], which introduces the NetLLM framework. This framework customizes LLMs for networking applications, enabling them to comprehend and produce domain-specific instructions and settings. NetLLM demonstrates the advantages of domain-specific fine-tuning by integrating networking terminology and contextual knowledge into its models. This guarantees that NetLLM can adeptly manage intricate scenarios with accuracy, including the orchestration of multi-vendor systems and the administration of hybrid network environments. The modular architecture facilitates incremental enhancements, guaranteeing sustained flexibility and efficacy.

The integration of LLMs in network engineering presents significant advantages, although it requires the adequate addressing of specific challenges like data privacy, ethical issues, and regulatory compliance. Huang et al. (2023) underscore the necessity of resilient governance structures to alleviate risks linked to automated decision-making and to guarantee conformity with industry norms. Collaboration among stakeholders, such as researchers, policymakers, and industry executives, is crucial for formulating rules that reconcile innovation with responsibility.

3. Methodology

Before diving into the specific activities, we provide a comprehensive overview of the data generation pipeline, as illustrated in Figure 1. The procedure begins with the acquisition of technical documentation from reliable sources. The raw data is subsequently curated, chunked, vectorized, and indexed into a vector database. Thereafter, relevant portions are extracted and processed, utilizing prompting techniques to generate Q&A pairs based on chosen network engineering topics. These are further automatically checked, deduplicated and sanitized to guarantee quality and validity. Finally, the curated dataset is ultimately used for fine-tuning.

3.1. Data Collection

The initial phase of our methodology entails the acquisition of relevant textual data from various data sources for the field of network engineering, including relevant literature, websites, and repositories comprising, for example, openly accessible volumes from the Network Solution Vendor's repository, specifically the Day One publications [12], supplemented by additional document corpus K documentation. The textual database consists of 79 books.

These textual resources must undergo vectorization, indexing, and subsequent storage within a vector database for practical use. The initial stage in developing such datasets is to collect information from various reliable sources. Relevant sources for network engineering include technical documentation, online manuals with setup instructions, research papers, and community forums like Network Engineering Stack Exchange and

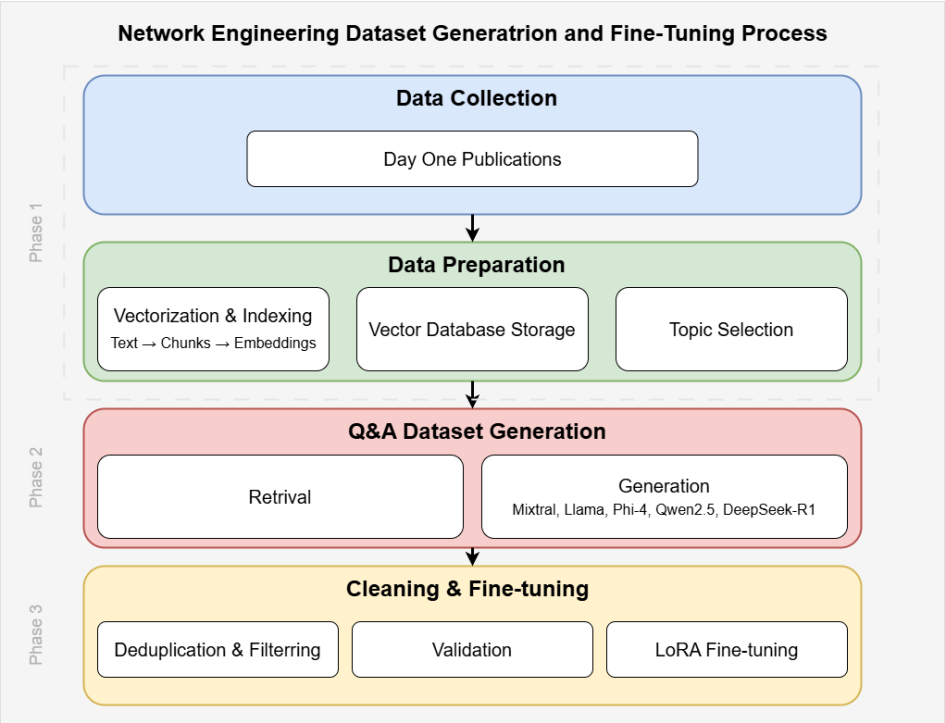


Fig. 1. Data Processing Diagram

Reddit. The development of the LLaMA model demonstrates this approach by integrating domain-relevant information acquired from sources GitHub repositories and Stack Exchange conversations to ensure high-quality and domain-specific content coverage [21]. Another similar effort is provided by NVIDIA research guidelines that encourage getting data from organized, high-quality sources generally underrepresented in common pretraining corpora [17].

Some studies emphasize the necessity of bias reduction and openness across the data process. When obtaining data from public forums or repositories, anonymizing sensitive material and documenting dataset provenance is critical for reducing bias and assuring trustworthiness. Frameworks such as ”Datasheets for Datasets” [7] offer systematic methods for describing qualities and constraints, promoting responsibility and confidence [17].

3.2. Data Processing Pipeline

Creating a comprehensive dataset comprising network engineering knowledge necessitates close attention to relevance, quality, and variety. As recent research has shown, creating suitable datasets requires a systematic data collection, curation, and assessment pipeline to guarantee that LLMs can generalize well and provide the appropriate domain-specific knowledge.

Data preparation involves the following steps: curation, chunking, vectorization & indexing, retrieval, and generation.

Data curation is critical for improving the dataset quality. Techniques like deduplication, noise reduction, and quality filtering are critical for maintaining consistency and focus. The LLaMA pipeline used deduplication and language identification to eliminate unnecessary or duplicated information and maintain dataset integrity [21]. Heuristic-based quality filters, such as sentence structure and the use of technical language based on the proper terminology, help to improve dataset relevance. This is especially important in network engineering, where correctness in vocabulary, technical clarity and precision are essential.

The diversity of datasets has a substantial influence on model robustness and fairness. Effective datasets include many network engineering subjects, including routing protocols, troubleshooting, and security setups. Given the field's worldwide scope, incorporating bilingual content is essential. Furthermore, gathering historical and contemporary data ensures that the dataset reflects changing trends in network technology [17].

Data chunking assumes breaking down and dividing the acquired documents into digestible, searchable chunks, also known as snippets, that are subsequently examined in more detail. The resulting chunks have a size of 1500 characters and an overlap of 200 characters. The text splitter uses a recursive algorithm to determine the best way of splitting the text in order to streamline the semantic flow between consecutive chunks. One approach that improves both chunking and vectorization entails the utilization of NLP capabilities, mainly through instruction-based prompts. These prompts direct the language model to produce task-specific embeddings customized for the desired application, such as information retrieval, semantic similarity, classification, or text assessment. This NLP-driven technique uses dynamic prompting to build embeddings that reflect the semantic subtleties pertinent to the job, unlike typical embedding methods that yield a singular, static vector for each input. This method enhances downstream performance by matching the created embeddings with particular use-case specifications.

Data vectorization and indexing assumes the conversion of data chunks into vector embeddings, as well as their storage and indexing into an appropriate vector database. Vector embeddings, as numerical representations of discrete or symbolic data points, prove indispensable for conducting similarity-based inquiries within artificial intelligence and machine learning frameworks. The indexing represents a fundamental transformation process, converting unstructured content into systematically organized, searchable excerpts. After a thorough analysis, these excerpts are encoded into a high-dimensional space and kept in a vector database. This configuration produces a repository that speeds up data association and retrieval. Also, Vector Databases are useful in addressing shortcomings, like hallucinations and outdated information. This makes them optimal for LLM integrations to provide more precise, current information in domain-specific contexts. [11]

Usually, the cosine similarity measure is used to compute the semantic similarity of two vectors stored in the vector database. This method allows us to compare symbolic concepts by computing the angle between their vector embeddings to estimate their semantic closeness. This measure quantifies the degree of alignment of the vectors, i.e., how well they point in the same direction. So, instead of just looking at the dot product, cosine similarity proposes a normalized quantity that brings a clearer idea of the degree of

similarity of two vectors, as described by Equation 1.

$$S_c(\mathbf{a}, \mathbf{b}) = \cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{1}$$

The research done in [19] shows a method to improve the efficacy of instruction-based fine-tuning. Possible applications are information retrieval, semantic similarity, classification, or text evaluation. By training on 330 diverse tasks with human-curated instructions, the method adapts its representations on the fly without needing additional fine-tuning for each new task. Extensive evaluations on 70 datasets demonstrate an average improvement of 3.4% over previous techniques, highlighting the effectiveness of instruction-based fine-tuning in producing versatile and robust text embeddings [19].

Data retrieval is responsible for finding and extracting relevant information from a vector database in accordance with a specified topic. During our dataset production process utilizing RAG (Retrieval-Augmented production), established topics are employed to query the vector database, extracting the most pertinent text segments to provide precise and contextually informed response generation.

It uses query processing algorithms that analyze and translate user input into relevant embeddings, leveraging the structured data obtained in the vectorization and indexing step.

The retrieval process searches and retrieves the most relevant document chunks by comparing these queries, a high-dimensional vector that semantically encapsulates the chosen topic.

The query embeddings are subsequently compared to pre-indexed document embeddings retained in the vector database. The indexing process preceding retrieval entails vectorizing and structuring the document corpus to facilitate rapid and precise similarity searches. The query embeddings are then compared to the pre-indexed document embeddings stored in the vector database. The indexing procedure, before retrieval, involves vectorizing and organizing the document corpus to enable swift and accurate similarity searches.

Data generation compiles and tunes the recovered document segments to be more coherent and informative (see the example from Figure 2).

The costly generative method is based on prompting techniques utilized on the recovered document segments. These prompts direct the language model to synthesize information and to produce logical, contextually aware queries and responses. This phase uses the model's ability to convert input data into significant outputs by blending many data points into a cohesive and useful outcome. The resultant question-answer pairs are subsequently consolidated into a newly generated dataset.

The workflow of the generating process is outlined by Algorithm 1. This algorithm uses an initial vectorized document corpus K and a mix of retrieval and generative techniques. It returns a large and comprehensive dataset D that complies with the predetermined specifications of the topic T, the document corpus K, the number of documents N that are retrieved, and the generative steps from 1 to S.

The algorithm starts with initializing the generator G using the appropriate generative model M and the specific topic T. These are essential components of data synthesis and sourcing.

Then, the algorithm systematically searches the relevant resources in the vector database K that comply with the topic T. This search is guided by a scoring system that evaluates

Algorithm 1 Retrieval and generation

Require: Topic T
Corpus of vectorized documents K
Generative model M
Number of documents to retrieve N
Number of generating steps S
Ensure: Generated dataset D
Initialize generator $G(M, T)$
$D \leftarrow \emptyset$ {Set of generated data}
$DocScores \leftarrow \emptyset$
for each $doc \in K$ matching T with $score$ **do**
  $DocScores \leftarrow \{(doc, score)\} \cup DocScores$
end for
for $s = 1, S$ **do**
  $TopDocs \leftarrow$ top N documents from $DocScores$
  $DatasetPrompt \leftarrow concatenate(T, TopDocs)$
  $GeneratedData \leftarrow G.generate(DatasetPrompt)$
  $D \leftarrow D \cup GeneratedData$
end for
return D

how well each retrieved document matches the topic T. The documents with the highest relevance are prioritized by selecting the top N that are the most relevant.

Afterward, the selected topic-specific documents are used to create a single, meaningful dataset partition that is relevant to topic T. In addition to synthesizing the chosen documents into a dataset, the generative process enhances the dataset by comprehending context and integrating additional content derived from the same source knowledge data.

The generative process depends on a prompt engineering method that employs topic-conditioned prompts. Each question is formulated by combining the chosen subject T with the top-N most pertinent retrieved document parts. This method directs the language model to produce Q&A pairs that are specific, contextually relevant, and consistent with the specified topic. This strategy, albeit lacking extensive parametric control, successfully confines the generated material to the targeted topic by anchoring the prompt in highly relevant textual evidence.

The assessment of the resultant dataset—specifically its congruence with topic T, the uniformity of replies, and the variety of included viewpoints—is elaborated upon in Section 4, since it constitutes an integral component of the experimental validation of our technique.

The network engineering topics were chosen for their importance in industry certification curricula (e.g., JNCA, JNCP, CCNA, CCNP), training materials, and technical documentation, guaranteeing comprehensive coverage of fundamental and advanced networking ideas. This selection method guaranteed that the dataset would thoroughly encompass both basic and advanced networking topics relevant to real-world professional scenarios. The end result dataset covers a diverse array of topics, including but not limited to Networking Fundamentals, OSI Model, TCP/IP Model, Routing and Switching Principles, Static versus Dynamic Routing, Routing Protocols (e.g., RIP, OSPF, EIGRP, BGP,

IS-IS), MPLS, VRF, PBR, Route Summarization, Load Balancing, Failover Mechanisms, Anycast and Multicast Routing, IPv6 Routing, Router Configuration, among others.

3.3. Q&A Dataset Generation with Multiple LLMs

In our experiments, we produced generated datasets for questions and answers with the following models: Qwen 2.5, Phi-4, LLaMA 3.3, LLaMA 3.1, Mixtral, and DeepSeek-R1, each including 1280 entries, with minor discrepancies in LLaMA 3.3 (1277 entries) and LLaMA 3.1 (1273 entries).

Each open-source LLM model produced a dataset that has 130 distinct topics,

guaranteeing comprehensive coverage of networking and security principles within a balanced distribution. Although the subject distribution is predominantly consistent, LLaMA 3.3 has minor discrepancies, with specific topics occurring 9 times, whilst LLaMA 3.1 features an anomaly where IP Addressing appears just 3 times. Notwithstanding these slight discrepancies, the datasets collectively offer a systematic and thorough depiction of essential networking and security concepts.

In scenarios where reliable data sources are not widely available, the ideas of variety and specificity in datasets are essential for improving the resilience and flexibility of models. For models to be able to adjust to new and unexplored data efficiently, dataset variety is a crucial component. Diverse datasets enable models to work across various activities and domains by lowering biases and enhancing flexibility. This diversity results from using multiple models to generate our data.

Lightly supervised data augmentation is one of the most effective strategies for promoting dataset variety. Exposing models to various instances, these datasets help models establish more robust feature representations and enhance their generalization capabilities.

Another way to ensure diversity is the utilization of multi-source datasets. Models can detect and take advantage of cross-domain correlations when different data types—like textual and structured data—or languages are integrated. Understanding subtle linguistic and structural signals is crucial for success in complicated tasks like named item identification and connection extraction, where these correlations are particularly helpful.

Conversely, dataset specificity offers a clear strategic advantage by allowing models to handle specialized jobs precisely. For instance, we could generate datasets for specialized applications using domain-specific and expert knowledge. This has proven useful in specialized settings as these datasets enable models to perform better than more generic solutions [6].

Additionally, specific datasets encourage innovation by addressing gaps usually missed by ordinary datasets. Private or otherwise difficult-to-acquire data may result in significant performance improvements when publicly accessible datasets are insufficient.

The synergy between specificity and variety is most evident in contexts with limited resources. The lack of labeled data in these situations calls for creative methods of dataset generation. Researchers may achieve a crucial balance between task-specific specialization and broad generalization by combining a variety of data augmentation approaches with domain-specific unique datasets. In addition to being adaptable to various applications, this balance guarantees that models perform very well in the specific settings for which they are created [6].

Improvements in model generalization and performance largely depend on the twin concepts of dataset variety and uniqueness. We may significantly improve the models' effectiveness and versatility by combining various linguistic and contextual components while creating task-specific datasets. In addition to addressing the difficulties of low-resource situations, these strategies help to further improve the data-driven approaches in NLP and other fields. High-quality datasets constitute the basis for practical model training, particularly in network engineering. Precise and flawless data enables models to provide valid setups and adjust to real-world requirements. Unreliable or overly generic information in a dataset might result in erroneous outcomes and constrain the model's capacity to perform specialized functions, such as routing and security.

Quality also encompasses the relevance of the data to the domain. A dataset comprised of irrelevant information will not facilitate a model's comprehension of network engineering intricacies. A model trained on software development data or unrelated tasks will have difficulty configuring firewalls or optimizing traffic flow. Datasets concentrating on critical domains—such as routing protocols, VLAN configurations, or compliance mandates—enhance the model's comprehension of network engineering.

As networks evolve and expand, high-quality datasets guarantee that models can adapt to emerging protocols, devices, and security threats. Training the model on substandard data may result in network configurations that are ineffective or susceptible to vulnerabilities. Upholding rigorous data standards not only mitigates these issues but also provides the model with the necessary adaptability in a dynamic field.

Coverage shows the extent and variety of the dataset, guaranteeing that the model encounters numerous types of devices and configurations. A comprehensive dataset in network engineering encompasses instances of network components, including switches, routers, and ACLs, as well as diverse protocols such as BGP, OSPF, and VLAN settings. This variability enables the model to manage diverse devices, traffic patterns, and operational requirements more efficiently.

To ensure complete coverage, the dataset must encompass many network situations, including fault tolerance designs, traffic load management tactics, and intrusion detection systems. Incorporating these real-world examples enhances the model's preparedness to address the varied difficulties in network management. This degree of coverage ultimately results in more precise, dependable, and versatile outputs that can accommodate intricate network environments.

Although the number is significant, it must not compromise quality or coverage. Extensive datasets are genuinely valuable only when they offer significant and varied instances that accurately represent the realities of network environments. Practically, possessing diverse data encompassing various device types, distinct protocols, and an array of configurations—from basic installations to intricate architectures—is significantly more advantageous than depending solely on volume. For example, incorporating thousands of nearly identical router configurations may increase the dataset's size. Yet, it minimally enhances the model's proficiency in handling specialized tasks such as managing complex routing topologies, implementing security protocols, or diagnosing rare failure conditions.

An effectively designed dataset harmonizes depth and breadth, encompassing routine procedures and exceptional cases requiring specialist expertise. This diversity enhances the model's comprehension of the network domain while mitigating the risk of overfitting to a limited range of configurations. When the data matches with real-world tasks, it fa-

cilitates more precise forecasts and suggestions, guaranteeing that essential network scenarios—such as high-traffic load management, fault tolerance, or intrusion detection—are not neglected. Consequently, the model acquires the necessary flexibility and reliability to function efficiently in the field instead of becoming ensnared by recurring instances that fail to test its capabilities.

3.4. Cleaning and Deduplication

Controlling dataset overlap and redundancy is a crucial part of large-scale data management. Data quality, storage efficiency, and analytical performance can all be severely harmed by duplicate entries and overlapping data. Deduplication algorithms have been created to overcome these obstacles and keep datasets optimal for contemporary uses.

Cutting down on duplication has wider ramifications than being a technical need. Redundant data makes data retrieval more difficult, raises expenses, and wastes storage. Analytical models and decision-making systems operate more quickly and accurately when dataset redundancy is reduced [6]. Furthermore, removing duplicate entries improves data integrity by avoiding inconsistencies resulting from keeping different copies of the same information. This is particularly important in fields where data reliability is crucial, such as healthcare and finance [6].

Controlling the semantic overlap of the dataset corpus becomes even more critical in low-resource contexts, where every data point has a significant influence on model training. The impact of duplicate or overlapping entries increases when data is scarce. According to [6], methods like weak supervision and targeted data augmentation highlight how crucial well-selected datasets are to maximize their usefulness. In addition to these techniques, deduplication is essential for ensuring models generalize well to new data.

Table 1. Cleaning Metrics

Model	**Before**	**After**	**Decrease (%)**	**Retained (%)**
Qwen 2.5	1280	1040	18.8	81.3
Phi-4	1280	596	53.4	46.6
LLaMA 3.3	1277	910	28.7	71.3
LLaMA 3.1	1273	496	61.0	39.0
Mixtral 8x7B	1280	274	78.6	21.4
Deepseek-R1	1280	55	95.7	4.3
Total	**7670**	**3371**	**56.0**	**44.0**

To refine LLMs and help network engineers, Wuang et al. propose multiple directions in which this improvement could be made: Network Design, Network Diagnosis, Network Configuration, and Network Security [8]. They also underline the issues for data security and privacy when using an LLM that is not open-source. By fine-tuning these models with

network-specific data, they can perform more relevant and secure operations, potentially transforming how network tasks are managed and executed.

An alternative approach that was researched involves converting the natural language high-level rules and requirements into low-level network Application Programming Interfaces (API) [23]. Thanks to this translation, network engineers can apply complex network settings more easily, eliminating the need to manually understand and apply complicated regulations. This technique translates verbose natural language instructions into exact network commands and settings using LLMs. This automates converting network needs into executable configurations, simplifying the process and reducing mistakes.

In the process of using LLMs to help network engineers configure or understand the network devices configuration, the main issue is validating the output [15] and the stated objective is to maximize the use of automated verifiers and minimize the use of human ones.

In a context where the need to fine-tune LLMs to more specific fields is growing, we aim to create a specific dataset that can help us improve LLMs for use in network engineering applications. This research aims to generate an extensive overview of the state of networking domains, utilizing various freely available sources.

The selection of books that together reflect the state-of-the-art in network engineering is the first step in the process. Another of our objectives is to create those datasets and minimize the need for RLHF. The Q&A dataset created from this corpus can be used as LLM fine-tuning material.

These expansions not only promise to enhance the current system's capabilities but also align with the ongoing advancements in AI and machine learning technologies, aiming to create more sophisticated, accurate, and user-friendly tools for data processing and analysis.

3.5. Fine-Tuning Approach

We began the training process by running three initial epochs for both LLMs to establish a baseline for their performance. To assess if this training period resulted in quantifiable gains in the model's performance, we raised the number of training epochs to 10. With this modification, we evaluated how more training iterations affected the models' capacity for efficient learning and generalization.

LoRA (Low-Rank Adaptation) is a method used to effectively fine-tune extensive machine learning models by modifying only a limited number of parameters, hence considerably diminishing computational resources and storage needs. This method is very helpful when dealing with resource-intensive models since it allows for effective task adaptability without requiring a lot of GPU memory or processing power. To maintain the scalability and affordability of the fine-tuning process, we sought to compromise model flexibility and computing efficiency by using LoRA . However, empirical studies have shown that in standard low-rank settings, LoRA often underperforms full fine-tuning regarding sample efficiency and target domain performance, particularly in challenging domains such as programming and mathematics. For instance, when it comes to instruction finetuning, full finetuning might get better results on tests like HumanEval and GSM8K, but LoRA keeps the base model's abilities better on tasks that aren't in the target domain. Moreover, a detailed singular value decomposition analysis reveals that full fine-tuning tends to learn perturbations with ranks 10 to 100 times higher than those typically used in LoRA

configurations. This suggests that the whole fine-tuning process captures a richer set of adaptations. Despite its limitations in achieving peak domain-specific performance, the reduced forgetting and enhanced efficiency offered by LoRA make it an attractive alternative, particularly in scenarios where GPU memory is a limiting factor. These insights provide a promising direction for future research to balance the trade-off between adaptation and generalization in LLMs [3].

4. Experiments and Results

4.1. Experimental setup

We evaluated our methodology with a dataset of 3,097 question-answer pairs of network engineering subjects, divided into a 9:1 training-to-test ratio for each topic. This guarantees diversity in both the training and testing datasets.

The experiments were conducted in three phases:

Phase 1: Embedding Generation from Source documents The first step involved processing and converting textual material from our selected sources into embeddings. These embeddings were stored in collections in a Chroma[4] database. The text embeddings were generated using LangChain's GPT4AllEmbeddings Python package [5]. With an average of roughly 292.5 embeddings per source, the resulting collection has 23,400 embeddings with 384 dimensions. This organized representation makes effective semantic search and retrieval possible in later stages.

Phase 2: Q&A Pair Generation The Q&A pairs were generated using DeepSeek-R1 671B, LLaMA 3.1 70B, Qwen 2.5 72B, and Mixtral 8x7B. Usually, running these model sizes requires dedicated, expensive infrastructure, so we rely on cloud-based LLM providers, such as OpenRouter [6], which provides a dedicated API to access the above-mentioned models.

Phase 3: Fine-tuning Fine-tuning was performed on LLaMA 3.2-1B, LLaMA 3.2-3B and Qwen 2.5-1.5B using LoRA, and we evaluated both training and computational performance key metrics summarized in Table 2. We run the fine-tuning process on cloud-based infrastructure with an RTX 6000 Ada GPU card with 48 GB VRAM, 64GB RAM and 16vCPU AMD EPYC 9354.

We conducted experiments comparing training for 3 versus 10 epochs to evaluate the impact of extended training, assessing the predictive and generative capabilities of the model using the metrics summarized in Table 3.

4.2. Results

In our experiments, we have used the following models for dataset generation: Mixtral, Llama, Phi-4, Qwen2.5, and DeepSeek-R1. After applying cleaning and deduplication steps, we combined the resulting data from all the LLMs and used it for fine-tuning.

[4] https://www.trychroma.com/
[5] https://python.langchain.com/
[6] https://openrouter.ai/

Table 2. Summary of Training and Computational Performance Metrics

Category	Metric	Description
Training Metrics	Gradient Norm	Measures the norm of gradients to monitor training stability.
	Loss	Represents the overall training loss function value.
	Training Loss	Captures the evolution of loss throughout the training process.
Computational Performance Metrics	Total FLOPs	Estimates the number of floating-point operations during training.
	Train Samples per Second	Indicates training throughput based on processed samples.
	Train Steps per Second	Measures the frequency of completed training steps.

Table 3. Summary of model's predictive and generative Evaluation Metrics

Metric	Description
Perplexity	Measures how well a model predicts the next word in a sequence based on predicting context. Lower perplexity indicates better language model performance.
BLEU	Calculates the similarity between generated and reference text based on overlapping n-grams, commonly used in machine translation evaluation.
METEOR	Considers synonyms, stemming, and word order while comparing generated text with reference text, improving upon BLEU.
ROUGE-L	Measures the longest common subsequence between generated and reference text, often used in summarization evaluation.
BERTScore F1	Evaluates semantic similarity using contextual embeddings from BERT, capturing meaning rather than exact word matches.
Average Inference Time	Records the average time taken to generate a response, which is crucial for evaluating model efficiency.
Sentence Similarity	Measures how semantically close the generated text is to the reference text using vector-based similarity methods.

We deployed a data-cleaning workflow to achieve accuracy and relevance. This involved clustering-based filtering, eliminating empty or excessively similar items, and deduplication, all designed to prevent recurrent queries from biasing the training process. Furthermore, we employed a configuration extractor that automatically recognized and validated the structural coherence of configurations incorporated into answers. This stage was crucial for preserving consistency and accuracy across datasets, which increased their dependability for jobs that came next.

With the preparation of the datasets, we see a reduction in size for each dataset, as depicted by Table 1: Qwen25 decreased by 18.8%, Phi-4 by 53.4%, LLaMA3.3 by 28.7%, LLaMA3.1 by 61%, Mixtral by 78.6%, and DeepSeek-R1 by 95.7%. It is noted that DeepSeek-R1 has the largest volume of useless data. Utilizing all the datasets produced by the models and following the cleaning process yielded 3,371 entries. The dataset comprises a total of 3,352,361 tokens (questions tokens: 361,865 and answers tokens: 2,990,496).

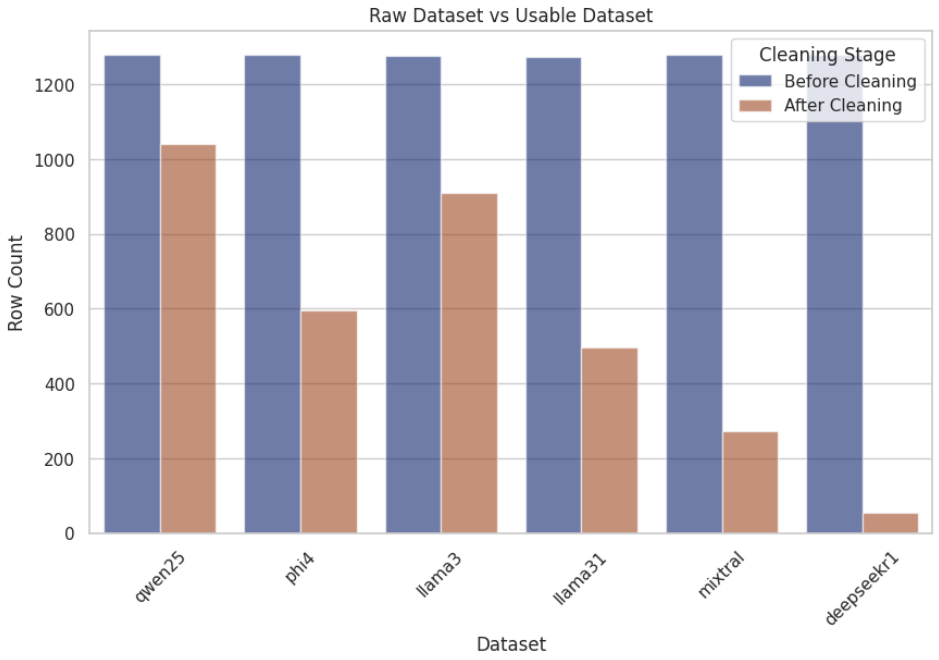


Fig. 2. Generated dataset vs usable dataset

Fine-tuning an LLM using LoRA has proven a more efficient way to adapt LLMs to new tasks and domains while significantly reducing the memory and computational overhead associated with full model fine-tuning. LoRA leverages low-rank adaptation by training only low-rank perturbations to selected weight matrices, thereby saving memory by freezing most of the pre-trained model's parameters [3].

This method not only cuts down on the number of trainable parameters but also keeps the model from going too far from its original state, which helps prevent catastrophic forgetting.

We used the resulting dataset to fine-tune models and improve the output for network-related tasks. We ran into a number of difficulties, especially while the model was being evaluated. The handling of the context window was one of the primary problems since there were times when the input sequence length was longer than the model's maximum permitted limit. We addressed this by implementing truncation, which made sure that any input that was longer than the maximum sequence length established by the model was trimmed appropriately.

Table 4. Performance Metrics for LLaMA 3.2 1B Before and After Fine-Tuning (3 Epochs)

	Test		Train	
Metric	**Before**	**After**	**Before**	**After**
Average Inference Time (s)	7.8891	10.1844	20.7003	11.0384
BERTScore F1	0.9389	0.9389	0.9398	0.9398
BLEU	0.6905	0.6438	0.5474	0.6448
METEOR	0.7687	0.8307	0.7549	0.7847
Perplexity	4.8191	4.1848	1.1811	3.3652
ROUGE-L	0.8124	0.8057	0.7931	0.7889
Sentence Similarity	0.9085	0.9085	0.9187	0.9187

Each model performance metric has been essentially impacted by fine-tuning, as shown in Table 4, with noteworthy variations in each model's quality and efficiency. For example, the LLaMA 3.2 1B model optimized for three epochs demonstrates a 13% decrease in perplexity (from 4.8191 to 4.1848) and an 8% rise in METEOR on the test dataset (from 0.7687 to 0.8307), suggesting enhanced fluency and confidence in language creation. Although the text quality increases in some ways, the production speed is hampered, as seen by the minor 7% decrease in BLEU and the 29% increase in inference time (from 7.8891 s to 10.1844 s) as specified in Table 5. Even while the perplexity rises—possibly as a result of a change in distribution brought on by regularization effects during fine-tuning—BLEU and METEOR improve by around 18% and 4%, respectively, on the training dataset for the same model.

The test results significantly improve after fine-tuning to 10 epochs for the LLaMA 3.2 1B model showing an overall improvement in fluency and resemblance to human reference texts as ROUGE-L shows a 3% boost, perplexity decreases by almost 15% (from 4.8191 to 4.0984), and METEOR rises by about 10% (from 0.7687 to 0.8465) despite a minor 6% decrease in BLEU and a 37% increase in inference time (from 8.0152 s to

Table 5. Performance Metrics for LLaMA 3.2 1B Before and After Fine-Tuning (10 Epochs)

	Test		Train	
Metric	**Before**	**After**	**Before**	**After**
Average Inference Time (s)	8.0152	10.9568	20.9625	6.7583
BERTScore F1	0.9389	0.9390	0.9398	0.9398
BLEU	0.6905	0.6492	0.5474	0.7186
METEOR	0.7687	0.8465	0.7549	0.8449
Perplexity	4.8191	4.0984	1.1811	1.8397
ROUGE-L	0.8124	0.8374	0.7931	0.8400
Sentence Similarity	0.9085	0.9085	0.9187	0.9187

Table 6. Performance Metrics for LLaMA 3.2 3B Before and After Fine-Tuning (3 Epochs)

	Test		Train	
Metric	**Before**	**After**	**Before**	**After**
Average Inference Time (s)	373.4122	23.2606	15.1097	35.6101
BERTScore F1	0.9389	0.9389	0.9398	0.9398
BLEU	0.2984	0.5834	0.7468	0.5064
METEOR	0.8008	0.8398	0.7998	0.8066
Perplexity	4.1884	2.0123	3.6710	2.3354
ROUGE-L	0.8539	0.7860	0.8529	0.7560
Sentence Similarity	0.9085	0.9085	0.9187	0.9187

10.9568 s). More time spent fine-tuning the model makes it better at picking up on small linguistic patterns, even if it means that the training distribution is not as closely matched to the test distribution. On the training side, BLEU improves by 31% and METEOR by 12%. Our findings are presented in Table 6. The LLaMA 3.2 3B models show even more significant gains. The test dataset demonstrates a nearly 94% decrease in inference time (from 373.41 s to 23.26 s), a nearly 96% increase in BLEU (from 0.2984 to 0.5834), and a 52% decrease in perplexity (from 4.1884 to 2.0123) for the 3-epoch version. These improvements show that fine-tuning significantly boosts efficiency while improving the output text's quality. Even though the training dataset shows a 32% decrease in BLEU, the model's higher test performance suggests that it improves after fine-tuning. The LLaMA

Table 7. Performance Metrics for LLaMA 3.2 3B Before and After Fine-Tuning (10 Epochs)

	Test		Train	
Metric	**Before**	**After**	**Before**	**After**
Time (s)	373.4629	15.0883	15.5681	14.1322
BERTScore F1	0.9389	0.9389	0.9398	0.9398
BLEU	0.2984	0.6986	0.7468	0.6919
METEOR	0.8008	0.8667	0.7998	0.8622
Perplexity	4.1884	2.6968	3.6710	1.6675
ROUGE-L	0.8539	0.8303	0.8529	0.8156
Sentence Similarity	0.9085	0.9085	0.9187	0.9187

Table 8. Performance Metrics for Qwen 2.5 1.5B Before and After Fine-Tuning (3 Epochs)

	Test		Train	
Metric	**Before**	**After**	**Before**	**After**
Average Inference Time (s)	0.9512	1.4414	0.8115	0.6249
BERTScore F1	0.9664	0.9664	0.9675	0.9675
BLEU	0.9502	0.9504	0.9468	0.9466
METEOR	0.9640	0.9614	0.9651	0.9668
Perplexity	4.8994	3.5097	3.5957	2.7674
ROUGE-L	0.9850	0.9816	0.9848	0.9859
Sentence Similarity	0.9469	0.9469	0.9500	0.9500

3.2 3B model achieves even more significant gains with 10 epochs: perplexity decreases by 36% (from 4.1884 to 2.6968), BLEU rises by about 135% (from 0.2984 to 0.6986), METEOR improves by 8%, and inference time decreases by almost 96% (from 373.46 s to 15.09 s). Significant improvements are also shown in the training dataset, including a 55% increase in perplexity. These findings clearly show that thorough fine-tuning of the 3B model results in a much more effective model that generates text that is more similar to high-quality human references.

On the other hand, the Qwen 2.5 1.5B models already have scores that are close to ideal. The test dataset's perplexity decreased by 28% (from 4.8994 to 3.5097) in the 3-epoch version, although BLEU, BERTScore, METEOR, ROUGE-L, and Sentence Sim-

Table 9. Performance Metrics for Qwen 2.5 1.5B Before and After Fine-Tuning (10 Epochs)

	Test		Train	
Metric	**Before**	**After**	**Before**	**After**
Average Inference Time (s)	0.9569	1.1790	0.8272	1.1962
BERTScore F1	0.9664	0.9664	0.9675	0.9675
BLEU	0.9502	0.9504	0.9468	0.9470
METEOR	0.9640	0.9620	0.9651	0.9589
Perplexity	4.8994	3.5218	3.5957	2.2646
ROUGE-L	0.9850	0.9835	0.9848	0.9816
Sentence Similarity	0.9469	0.9469	0.9500	0.9500

ilarity remained high, ranging from 0.95 to 0.97. Nevertheless, the inference time rises from 0.9512 s to 1.4414 s, a 52% increase. The values for BLEU, BERTScore, METEOR, and ROUGE-L stay constant, but inference time even drops by 23% on the training set as shown in Table 8. The 10-epoch version of the Qwen model 9 shows a slight rise in inference time (about 23%) and a decrease in perplexity on the test dataset of around 28%, but other quality metrics essentially stay the same. This illustrates that fine-tuning improves the Qwen model's performance, mainly by reducing perplexity, which implies a smoother, more coherent generation.

Out of all the assessed models, the LLaMA 3.2 3B model that has been adjusted for 10 epochs performs the best overall. Its test dataset shows a remarkable 135% gain in BLEU, 36% decrease in perplexity, and 96% reduction in inference time. These enhancements show that the model is more effective in practice, which is crucial for real-world applications. In conclusion, even if each model may be improved, the LLaMA 3.2 3B 10-epoch model offers the most appealing alternative for deployment when speed and text quality are crucial due to its significant efficiency improvements.

5. Conclusions

In this paper, we developed and evaluated a novel approach for automatically generating domain-specific datasets to fine-tune open-source Large Language Models and we described our approach in the context of network engineering applications.

We introduced an effective pipeline for dataset generation that leverages existing technical documentation and literature in the network engineering field that contributes to building knowledge databases to create high-quality question-answer pairs spanning diverse network engineering topics. The automated generation process produced initial datasets from multiple LLMs (LLaMA, Qwen, Mixtral, Phi-4, and DeepSeek), which were refined through rigorous cleaning and deduplication. This process resulted in a final

dataset of 3,371 entries containing over 3.3 million tokens, covering networking concepts like routing protocols, security configurations, and network services.

We conducted a comparative analysis of data quality across different source models that revealed significant variations in the usability of generated content. We indicate that models like Qwen 2.5 maintained high retention rates (81.3%) after cleaning. In contrast, others, such as DeepSeek-R1, showed extremely low retention (4.3%), underscoring the importance of model selection in the dataset generation phase.

Our experiments with fine-tuning smaller LLMs using LoRA showed significant performance improvements. The LLaMA 3.2 3B model fine-tuned for 10 epochs showed the most dramatic improvements. This increased the BLEU score by 135%, decreased perplexity by 36%, and reduced the inference time by 96%. Based on these observations, we can say that the results confirm that even relatively small models can achieve significant domain-specific performance gains through targeted fine-tuning with high-quality datasets.

The methodology depicted in this paper offers a practical approach for adapting open-source LLMs to the network engineering domain without requiring extensive computational resources or human-annotated datasets. This approach is particularly valuable where domain expertise is critical, but labeled training data may be insufficient.

Future research directions include extending the proposed methodology to other domains, such as cloud-based systems, to develop a more complex validation procedure, including human expert evaluation, to guarantee the correctness of the generated content and explore data augmentation techniques for improving model performances.

References

1. Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R.J., Javaheripi, M., Kauffmann, P., Lee, J.R., Lee, Y.T., Li, Y., Liu, W., Mendes, C.C.T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., Zhang, Y.: Phi-4 technical report (2024), https://arxiv.org/abs/2412.08905
2. Ben Houidi, Z., Rossi, D.: Neural language models for network configuration: Opportunities and reality check. Computer Communications 193, 118–125 (Sep 2022), http://dx.doi.org/10.1016/j.comcom.2022.06.035
3. Biderman, D., Portes, J., Ortiz, J.J.G., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., Cunningham, J.P.: Lora learns less and forgets less (2024), https://arxiv.org/abs/2405.09673
4. Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4302–4310. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
5. DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z.F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J.L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R.J., Jin, R.L., Chen,

R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S.S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W.L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X.Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y.K., Wang, Y.Q., Wei, Y.X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y.X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z.Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., Zhang, Z.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning (2025), https://arxiv.org/abs/2501.12948
6. Deng, S., Ma, Y., Zhang, N., Cao, Y., Hooi, B.: Information extraction in low-resource scenarios: Survey and perspective (2024), https://arxiv.org/abs/2202.08063
7. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., III, H.D., Crawford, K.: Datasheets for datasets (2021), https://arxiv.org/abs/1803.09010
8. Huang, Y., Du, H., Zhang, X., Niyato, D., Kang, J., Xiong, Z., Wang, S., Huang, T.: Large language models for networking: Applications, enabling techniques, and challenges (2023), https://arxiv.org/abs/2311.17474
9. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., de las Casas, D., Hanna, E.B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L.R., Saulnier, L., Lachaux, M.A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T.L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mixtral of experts (2024), https://arxiv.org/abs/2401.04088
10. Jiang, X., Gember-Jacobson, A., Feamster, N.: Caip: Detecting router misconfigurations with context-aware iterative prompting of llms (2024), https://arxiv.org/abs/2411.14283
11. Jing, Z., Su, Y., Han, Y.: When large language models meet vector databases: A survey (2024), https://arxiv.org/abs/2402.01763
12. Juniper: Day one books (2023), https://www.juniper.net/documentation/jnbooks/us/en/day-one-books, accessed on 20 november 2023
13. Maatouk, A., Ampudia, K.C., Ying, R., Tassiulas, L.: Tele-llms: A series of specialized large language models for telecommunications (2024), https://arxiv.org/abs/2409.05314
14. Mani, S.K., Zhou, Y., Hsieh, K., Segarra, S., Chandra, R., Kandula, S.: Enhancing network management using code generated by large language models (2023), https://arxiv.org/abs/2308.06261
15. Mondal, R., Tang, A., Beckett, R., Millstein, T., Varghese, G.: What do llms need to synthesize correct router configurations? (2023), https://arxiv.org/abs/2307.04945
16. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback (2022), https://arxiv.org/abs/2203.02155
17. Parmar, J., Prabhumoye, S., Jennings, J., Liu, B., Jhunjhunwala, A., Wang, Z., Patwary, M., Shoeybi, M., Catanzaro, B.: Data, data everywhere: A guide for pretraining dataset construction (2024), https://arxiv.org/abs/2407.06380
18. Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report (2025), https://arxiv.org/abs/2412.15115
19. Su, H., Shi, W., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., tau Yih, W., Smith, N.A., Zettlemoyer, L., Yu, T.: One embedder, any task: Instruction-finetuned text embeddings (2023), https://arxiv.org/abs/2212.09741

20. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. CoRR abs/2302.13971 (2023), https://doi.org/10.48550/arXiv.2302.13971
21. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models (2023), https://arxiv.org/abs/2302.13971
22. Trăistaru, C., Pop, F., Bădică, C., Ciochiu, D., Bădoi, M., Nedianu, G.C.: Leveraging open source large language models to generate datasets from existing field-specific texts (2024)
23. Wang, C., Scazzariello, M., Farshin, A., Kostic, D., Chiesa, M.: Making network configuration human friendly (2023), https://arxiv.org/abs/2309.06342
24. Wu, D., Wang, X., Qiao, Y., Wang, Z., Jiang, J., Cui, S., Wang, F.: Netllm: Adapting large language models for networking. In: Proceedings of the ACM SIGCOMM 2024 Conference. p. 661–678. ACM SIGCOMM '24, ACM (Aug 2024), http://dx.doi.org/10.1145/3651890.3672268
25. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P.S., Sridhar, A., Wang, T., Zettlemoyer, L.: Opt: Open pre-trained transformer language models (2022), https://arxiv.org/abs/2205.01068

Claudiu Trăistaru is an assistant lecturer at the Department of Computer and Information Technology, within the Faculty of Automation, Computers and Electronics at the University of Craiova, Romania. His research interests include cloud computing technologies, computer networks, cybersecurity, and applications of artificial intelligence. He has been working as an IT professional for more than 20 years and has extensive experience in systems engineering, DevOps, and infrastructure management.

Florin Pop is full Professor at the Computer Science and Engineering Department of National University of Science and Technology POLITEHNICA Bucharest, Romania. His main research interests are in large-scale distributed systems, adaptive and autonomous methods, multi-criteria optimization methods, Cloud middle-ware tools and applications development, prediction methods, self-organizing systems, applications for Big Data and IoT systems, Smart Cities, Climate Neutral Governance. Florin Pop is an editor of Future Generation Computer Systems. He is senior researcher at National Institute for Research Development in Informatics - ICI Bucharest and member of Academy of Romanian Scientists. He is IEEE Senior Member.

Costin Bădică b. February 19, 1966, in Craiova, Dolj. He is a Professor Doctor Engineer at the University of Craiova, Faculty of Automation, Computers, and Electronics. Since 2012, he has been the Director of the "Constantin Belea" Doctoral School within the Faculty of Automation, Computers, and Electronics at the University of Craiova. As of 2024, he is an Associate Member of the Academy of Technical Sciences of Romania (ASTR), Section: Information and Communications Technology, Computers, and Telecommunications. He serves as Co-Chair of the Technical Committee on Computational Collective Intelligence for the IEEE SMC Society. This committee received the "Most Active SMC Technical Committee Award" in Cybernetics from the IEEE SMC Society in both 2021 and 2024. He is a member of the editorial boards for the following journals: Computer

Science and Information Systems (ComSIS), ISSN 1820-0214; System Theory, Control and Computing Journal, ISSN: 2810–4099; Cybernetics and Information Technologies (CIT), The Journal of the Institute of Information and Communication Technologies of the Bulgarian Academy of Sciences, eISSN 1314-4081; Simulation Modelling Practice and Theory, ISSN 1569-190X (WoS); Vietnam Journal of Computer Science, ISSN (print): 2196-8888 — ISSN (online): 2196-8896. He is a founding member of the international conference IDC (Intelligent Distributed Systems) – 2007. His areas of interest include: artificial intelligence, multi-agent systems, distributed systems, and software engineering.

Catalina-Felicia Mancas is a lecturer and a researcher of the Computer and IT department at University of Craiova since 2010. With a solid teaching expertise in Computer Networks and a strong research interest in Network Engineering and Quantum Communication, Catalina combines academic expertise with practical knowledge to support innovation in computing and communication technologies.

Ionuț Murarețu is a lecturer in the Department of Computer and Information Technology, within the Faculty of Automation, Computers and Electronics at the University of Craiova, Romania. His research spans several areas of computer science and engineering, with a focus on artificial intelligence, software engineering, embedded systems, the Internet of Things (IoT), multi-agent systems, and computational optimization. He is actively involved in teaching, research, and collaborative projects at the intersection of AI and embedded technologies.

Received: April 16, 2025; Accepted: July 25, 2025.

Computer Science and Information Systems 23(1):561–583 https://doi.org/10.2298/CSIS250425014G

Cascade Systems as An Implementation of a Gray-Box Architecture: A Case Study in Traceability for Microservice Scaling ⋆

Jorge Jiménez García[1], Ignacio Lacalle Úbeda[1], Paweł Szmeja[2], Katarzyna Wasielewska-Michniewska[2], Przemysław Hołda[2], Maria Ganzha[2], Carlos E. Palau Salvador[1], Costin Bădică[3], Stefka Fidanova[4], and Marcin Paprzycki[2]

[1] Universitat Politècnica de València, Valencia, Spain
jjimgar1@upv.es
iglaub@upv.es
[2] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
pawel.szmeja@ibspan.waw.pl
marcin.paprzycki@ibspan.waw.pl
[3] Department of Computers and Information Technology, University of Craiova, Craiova, Romania
badica_costin@software.ucv.ro
[4] Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria
stefka@parallel.bas.bg

Abstract. There is great potential in leveraging Artificial Intelligence (AI) systems to optimize complex infrastructures, automate difficult tasks, or support autonomy and coordination between networked devices. However, advances in state-of-the-art AI often neglect features and/or requirements that businesses care deeply about, namely traceability and explainability. A majority of available research has not explored much the deployment of semi-physical architectures combining fuzzy rule-based systems with more opaque models to improve explainability, being this specially true for the management of microservices in cloud and cloud-edge environments. This contribution builds on previous work that proposes a middle ground of mixed AI architectures that combine the performance of black-box AI models with "a more explainable overall architecture" by implementing a microservice scaling system for distributed cloud environments using a cascade approach. This work demonstrates and evaluates an application case of such an approach departing from a Service Level Agreement compliance, in a case of microservice scaling decision over cloud (and cloud-like) infrastructures.

Keywords: explainability, fuzzy-rule-based-systems, semi-physical architectures, AI network, gray-box

1. Introduction

Artificial Intelligence (AI) systems have an increasing impact across modern society, from personal life, where these systems can know songs one would like to hear, answer questions, or enable creativity by generating images; to the enterprise world, where AI is used

⋆ This paper is a continuation of a Conference Paper first published in INISTA 2024 as DOI 10.1109/inista62901.2024.10683838

to try to predict stock market trends, identify faults in assembly line products, or aid development of new materials, among many other uses.

Enterprise use cases, however, present additional requirements. While an average user might just be annoyed when recommended a book they did not like, the operator of the recommendation system may want to know exactly "what went wrong" in making the recommendation. This is doubly true for various infrastructures, where the impact of an AI error may be more significant than simple annoyance. If, for example, in the Smart City, the energy generation and provisioning is driven by an AI system and, as a result of a mistake made by that system, the city experiences a blackout, the citizens, officials, and grid's operator(s) have to know what happened and fix the problem if (and as soon as) possible.

Another relevant example is found in microservices allocation across a computing infrastructure. Cloud providers, and other related businesses, quantify their success on their capacity to efficiently dedicate their resources to the dynamic demand. Often, intricate models are used to support such decisions, that may lead to widely different levels of productivity and Quality of Service (QoS) to the final user.

Knowing why AI-driven systems obtain specific results is complicated and research in Explainable AI (XAI), is concerned with finding answers. Here, some systems are relatively simple to understand. For instance, rule-based or decision tree-based systems are inherently explainable, due to their design and inner workings. However, modern architectures like Neural Networks or Reinforcement Learning based models are not easily explainable. This concerns, in particular, large and complex models and/or models trained on large (and very large) datasets. Unfortunately, these inexplicable (black-box) models show significantly increased performance when compared to their simpler, explainable (glass-box) relatives. Therefore, a lot of research effort is devoted to explain why a given model "behaves the way it does". The work in this paper departs from a previous effort by the authors to conceptualize an AI system building paradigm based on the use of discrete, independently trained AI subsystems, combined to create a system with a complex goal such as microservice allocation, by leveraging independent elements in a cascade manner [1].

In the current work, authors go beyond and analyse deeply the potential applicability of XAI methods in cloud-edge environments, as well as illustrating the XAI-as-architecture design with a practical example. In particular, the options of using different techniques over AI models to allocate resources in a distributed environment illustrate that there are various possible avenues. From our perspective, exploring gray-boxed design in the ML algorithms that recommend the commissioning of workloads should prevail over opaque, end-to-end systems. Here, by leveraging discrete components, the goal is to increase overall explainability thanks to their known purpose(s), and to use their interrelations to bridge the performance gap, when compared to a purely black-box model. Such approach helps also to identify flaws in AI systems, by being able to identify which (compartmentalized) element(s) of the overall system is(are) causing errors, as well as increasing the ease of fixing by allowing integrators to more easily replace "faulty" segments of the network.

2. Related work

2.1. Characteristics and types of XAI

Analysing existing literature on the benefits and necessity of XAI, authors of [2], [3], deliver a comprehensive study, focused on regulation compliance, user trust, and the potential to develop, or gain, deeper understanding of the cases where AI systems are used. They also go over the types of information that are relevant for the different user profiles, their roles, and expectations, as well as the insight gained from each sort of explanation of the model, and the issues it can help identify. From the system operators viewpoint, it is possible to identify the understanding of general model behaviour and of individual predictions, as the critical aspects of explainability for the infrastructure, known as global and local explainability, respectively [4].

In another work, authors of [5] studied current approaches to XAI in an industrial context. They proposed an end user, which is conceptually similar to an infrastructure operator, with similar requirements and concerns. It is evident from their compiled studies that the majority of recent advances tend to use post-hoc methods for explainability, which attempt to explain the behaviour of inherently non-explainable models after training, instead of attempting to improve the glass-box models to match the other systems.

Besides this, [6] builds upon the computational analysis framework introduced in [7]. In it, the author differentiates levels of explainability in three broad categories - computational ('what is it doing'), algorithmic ('how is it doing it'), and implementational ('where is it doing it'). To further the reliability of AI models for industry, all of these categories must be accounted for. From the perspective of industry stakeholders, *what* is already accounted for, and *where* is relatively controllable with the advent of distributed compute. It is the remaining question then, *how*, that is the main roadblock for explainability - given the popularity of black-box predictors.

2.2. Explainability for black-box models

Methods for explainability in black-box models, in particular in the case of system operators, which are of core concern here, are: for local explainability covering predictions, Shapley values-based techniques (SHAP) [8], and for global explainability regarding the behaviour of the model, LIME-based methods [9]. However, studies of both methods come to the conclusion that they are not robust enough to be relied on, to deal with the industrial-world explainability concerns [10], [11]. Regardless, the currently accepted practice is still to utilize decoupled XAI system, and relying on post-explainability, meaning the outputs are only explained *a posteriori*. [12]. More broad, comprehensive studies, covering black-box models report that the current techniques are insufficient or lacking in delivering understanding of local and global behaviour(s) [13]. Additionally, many scholars have criticised the use of black-box models combined with explainability techniques, since more explainable models would suffice [14], [15]. Nonetheless, reasons for such solutions being still used, regardless of their limitations, may be related to strict requirements for a specific model architecture that cannot be influenced, but at the same time the necessity to include explainability in the system in any feasibility form. Decoupled and a posteriori explainability is most intuitive and easy to include without influencing the black box type system internal structure. Moreover, such architectural approach can be

treated as an optional "add-on" that from the administrative point of view may be easier to manage.

2.3. Explainability in microservices management over heterogeneous Cloud-Edge environments

The cloud computing paradigm, although predominant for the last decade or so, is increasingly widening its scopes, considering other -more heterogeneous- type of devices to support modern services execution. The continuously changing technological landscape has witnessed recent advances related to more intelligent devices capable of combining their capacities to perform tasks with varying demands. The previous, together with programmable networks, development of cognitive cloud systems and advances in orchestration across different cloud environments has resulted in a whole new set of challenges. Additionally, the above exposed advent of explainability opens up a new research field aiming at providing a better interpretable AI-based decisions with regards to microservices management (e.g., allocation, failure diagnosis, dynamic scaling, etc.).

In this context, specially in the last very few years, some literature can be found dealing with this matter. In 2021, [16] already envisioned the application of XAI metrics and visualization techniques to improve transparency and interpretability of cloud resource allocation (which was, in turn, commissioned by the result of generative models and reinforcement learning models). However, those techniques were mentioned, in abstract, without specification or demonstration.

In [17], the usage of XAI techniques was proposed over a simulated cloud-only environment to improve robustness and transparency. Departing from modelled cloud resources, it generates synthetic data and suggests the applicability of generic XAI characteristics over Generative AI models (Variational Autoencoders -VAEs- and Generative Adversarial Networks -GANs) that predict the modification of those resources after workload allocation. However, again, auhors did not propose any particular methods or tools to be applied, stopping at generic/theoretical level. Remarkably, the work did not explore the implications of moving to edge or IoT elements neither. In [18], authors went deeper and proposed a high level framework that depicts how to possible embed such XAI concepts into a Deep Q-Network-based resource allocation over cloud-only resources, dealing with a state, action reward tuples following a reinforcement learning structure. Again, concreteness regarding XAI is very scarce and assumptions are only discussed over simulated scenarios.

Notwithstanding, at a certain point specific methods and tools were commenced to be applied over cloud-native environments. In this sense, several XAI techniques exist and can be classified into global (explaining the entire model) or local (explaining specific predictions). Other authors also add classification depending on its agnosticity to the model and on the stage of development of such model (ante, post) [19]. Very recently, a comprehensive survey [20] exposed that explainable methods are currently instilled in microservices management (mostly, malfunctioning and failure diagnosis). Although not strictly related with the scope of this paper, such work illustrates how certain techniques such as causal inference-based methods (e.g., DoWhy, Granger causality) or rule-based or graph-based dependency tracing (e.g., service dependency graphs) can improve system-level explainability and transparency in cloud environments dealing with microservices.

As discussed, microservices malfunctioning was a major concern in the first practical utilisation of XAI in cloud environments. Here, the goal was to showcase to the human why such failures occurred, or highlighting which would be the causes based on ML predictions. For instance, Authors in [21] proposed a novel system (ShapleyIQ) based on Shapley weighted values to analyse cloud-data-center-like system diagnosis (https://github.com/lonyle/ShapleyIQ). The approach followed was to decompose a complex system into smaller pieces, and adding splitting invariance over the traditional Shapley values method. A positive note is the utilization over 15,629 operations on 86 database services, deployed on 2,546 machines in a relevant Kubernetes enviroment.

Furthermore, resources scaling in cloud environments was also a main research target. Cloud vertical scaling involves pre-defining and allocating specific compute resources (CPU, RAM, etc.) to a server/cluster/deployment based on known workload requirements, ensuring it can handle expected performance demands. Among other works, in [22] authors proposed a resource scaler for Kubernetes environments and the application of XAI methods to elaborate on the cases where, after such scaling, a service would still experience some degradation. SHAP (SHapley Additive exPlanations) is used to deduce the most important features that yield such degradation that was, in turn, predicted by an XGBoost model. The final goal was to understand the root cause in order to enable more efficient use of the resources available in the infrastructure, which has certain commonalities to the field under study in this paper (service allocation). In a similar line, [23] aimed as well at building an explainable autoscaling framework for Kubernetes based on custom controller and requirements based on user surveys, thus proposing a relevant evaluation framework on how to assess the explainability of such system. Also in the cloud scaling context, a relevant experiment was performed in [24], where XAI is inserted when using LLMs for a trustworthy anomaly detection in dynamic-scaling, cloud-native based 6G networks. In particular, a use case is presented using XGBoost as AI, SHAP as XAI, and Llama2 as LLM. After the experimentation, which is available in a video demonstration [5], authors outline that the combination of usual XAI tools (e.g., SHAP), which derive numerical results, with LLMs (that provide actionable, human-readable information) might remarkably improve the explainability and transparency. However, the work does not consider microservice allocation, and does not explore the heterogeneity of cloud-edge-IoT environments.

Significantly, the interest in including XAI techniques in cloud environments has not stopped at research practitioners. The reference [25] outlines the already on-going applicability of XAI techniques in usual cloud CI/CD processes, naming renowned cloud providers and commercial tools that are currently working in this line TensorFlow Explain, H2O.ai, AWS SageMaker or embedded XAI in Azure Machine Learning or Google AI Platform.

However, literature is much scarcer whenever considering heterogeneity in computing resources (i.e., not cloud-only). The fact of encompassing edge servers and other less capable devices has only been covered in a few works. In [26], a group of researchers presented ORRIC, a linear complexity online algorithm that does indeed consider computation-limited resources (e.g., edge servers), devising a proper way of allocating AI training and inference loads on such devices. While not including standard explainability tools, the work actually tests over CIFAR-10-C dataset in a real edge com-

[5] https://www.youtube.com/watch?v=1CoDNVWJcqA

puting environment. ORRIC focuses on data and model drift for resource allocation upon MobileNetv2 and RestNet50 models, improving the performance of existing approaches. Explainability-wise, since the model does not rely on neural networks or other opaque mechanisms, the procedure followed can be traceable. However, it does not focus on model or process explanation per se. Contrary to other publications, an open source repository is provided, allowing for experiment replication [6].

Still, one of the most relevant references for the study in question in our paper, is [19] where authors outlined which XAI methods would be more appropriate in cloud-edge orchestration cases. Services allocation in a distributed environment build upon the application of certain AI algorithms over a set of gathered data (monitoring). Such data (e.g., network status, CPU, RAM of the nodes, services/workloads characteristics, etc.) can be expressed in various formats. Authors suggest that, for models that address allocation based on tabular data, SHAP, LIME (Local Interpretable Model-Agnostic Explanations) and counterfactual facts would perform well. In cases where timeseries data is prominently used, recommended XAI methods would be Integrated Gradients, Grad-CAM (Gradient-weighted Class Activation Mapping) and DeepLIFT (Deep Learning Important FeaTures). Finally, those cloud-edge orchestration models on top of raw text data (e.g., NLP or LLM) should be focused on explaining where the attention mechanisms have been placed.

As a conluding remark, while some insights on explainability and evaluation of explainability are available in cloud environments, they only cover a broad spectrum of applications, limiting its deployment in cloud-edge cases to a few isolated examples.

2.4. Gray-box models and the cascading approach

The in-between of explainable models (neither black- nor glass-box) are referred to as gray-box models. Typically they combine black-box models with different techniques that increase the explainability, without significantly decreasing the accuracy [27]. For instance, [28] starts with a Convolutional Neural Network (CNN) which can detect if a drain is obstructed or not. In an effort to improve the explainability, the authors propose to segment the behaviour of the network, devising a Rule-Based System for classification, once certain elements have already been detected within the images. The intermediate element symbolizes the "reasoning" of the system; i.e. if a drain is detected to be obstructed, it can be explained because there were a number of obstructing elements detected in the image.

Interestingly, in [29], the opposite strategy is adopted, where the authors develop a Guided Bayesian Optimizer (GBO) for the memory allocation. Here, a white box model is used first, to refine the incoming variables, that are then fed to a black box Bayesian Optimizer (BO). In terms of explainability, the refinement of the input variables leads to an increased understanding of the underlying dynamics of the memory allocation, and is reflected on the lower time-to-converge, accuracy of parameters, compared to an exhaustive configuration search and the inference time, compared to a RL algorithm. Authors also compare these systems to RElM, a fully white-box expert system they developed that performs significantly better, ultimately highlighting that incorporating expert knowledge into these algorithms improve their performance. In a similar manner, [30] separates

[6] https://github.com/caihuaiguang/ORRIC

the face detection process in a two step approach, to leverage Support Vector Machines (SVM), tasked with detecting the separate parts of a face, and a geometry model that takes in the detected parts and assesses if they form a face. This explicit separation benefits the model in significantly improving its robustness, as well as indirectly improving explainability.

Note that these approaches do not make the models completely interpretable, but rather make them "*more interpretable*" than a pure CNN or BO approaches, while retaining the performance benefit of these types of deep learning architectures.

Further, works such as [31] envision graphical-based explainability methods resistant to online updates, and crucially, mostly decoupled from the inference and framework, such that the additional cost for explainability is reduced as much as possible. Given this graphical structure, devising a graph structure for AI systems in a cascade-way seems to be helpful to further explainability. Obviously, this approach is only feasible if we have intermediate computations to base our cascade predictors on. More generally, architectural explainability builds on the previously mentioned computational analysis framework, accounting for the 'how' of explainability: the chain of operations to obtain the output.

2.5. Semi-Physical Network architectures

Exploring explainability in complex models, [32] introduces the idea of a Dynamic Semi-Physical Network. A Semi-Physical Network uses preexisting knowledge of a domain space to model the internal connections of a Neural Network system, composed of distinct subnetworks to reflect known relationships, found in the real world. In general, it can be stated that, for a function that is the result of several intermediate functions that have some sort of relation, the network that represents the interconnection of these (intermediate) functions would be more understandable than a fully black-box model, by making certain connections explicitly known. An example of such a representation can be seen in Figure 1. The authors also assert that more detailed (and faithful) model structures, matching real world relationships, result in increased performance from the system, delivering benefits of applying expert knowledge. Note that the intermediate functions do not necessarily have to be Neural Networks. They can be directly modelled by mathematical functions, or other estimators.

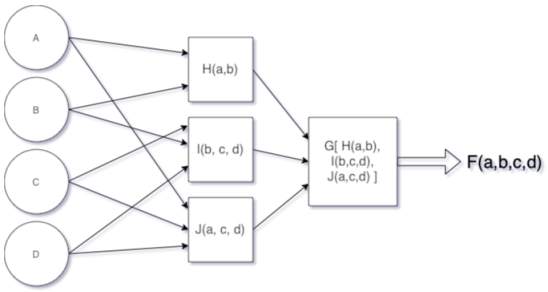


Fig. 1. Semi-Physical representation of function $F(a, b, c, d)$, that is composed of $H(a, c)$, $I(b, c, d)$, and $J(a, c, d)$

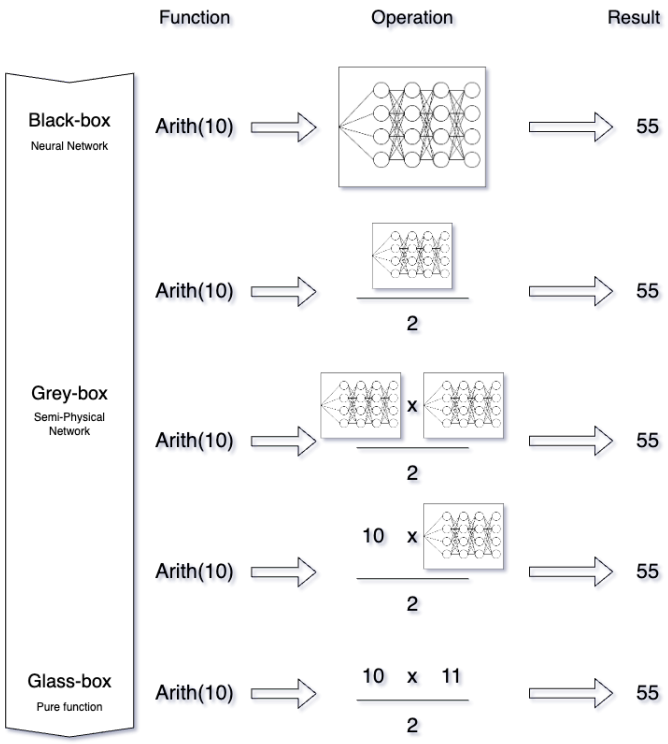


Fig. 2. Gradient of explainability showing different model architectures to calculate the arithmetic sum $1 + 2 + 3 + \cdots + 10$

This approach is supported by research reported in [33] that analyzes the sparsity of parameters in Neural Networks, which can be understood to represent the intermediate functions mentioned previously. Additionally, researchers show that, since this relationship is structural, improvements to NNs through pruning cannot be learned from scratch. In a number of studies, Semi-Physical Networks are successfully applied and shown to outperform NN approaches [34]–[36]. In this way, Semi-Physical Networks provide a middle ground between black and glass-box architectures (See Figure 2)

2.6. Fuzzy Logic for Semi-Physical Networks

Fuzzy logic [37] is defined as a logic system where a truth value can take any real number between 0 and 1, as opposed to the Boolean logic, which only allows the discrete values 0 and 1. Applied to Semi-Physical Networks, this can be used to help describe the in-between intermediate functions that are nebulous or uncertain. Conceptually, it can be seen as knowing what level of relationship or dependence exists between an intermediate function and the final result, without knowing what the relation actually is. Fuzzy rule-based systems (FRBS) are a kind of rule-based systems that aims to capture the uncertainty of the represented knowledge [38] by assigning a degree of truth to the

rule predicates, which may also be overlapping instead of the binary rule-based-systems. An example is shown in Figure 3. This flexibility allows a FRBS to address non-linear problems, such as in Neural Networks, with the inherent explainability, transparency, and determinism of the rule-based-systems.

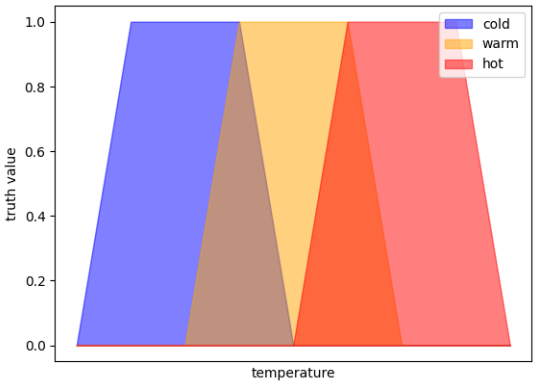


Fig. 3. Example of fuzzy set for 'temperature'. The discrete terms have overlap among them, representing their uncertain threshold, which are auto-determined.

This means that the obtained rules are as globally explainable as they can be, due to their deterministic nature and explicit representation of knowledge [39], [40], while still having a satisfiable degree of local explainability thanks to the fuzzy determination of predicates. In [41], a Fuzzy Semi-Physical Network is successfully developed, for approximating properties of Lithium-Ion based batteries. Fuzzy Semi-Physical models consistently outperform their non-fuzzy and non-semi-physical counterparts, by using a FRBS regressors as approximators of the unknown intermediate function in a model [42], [43]. Going back to the example in Figure 1, one can imagine that H and J could be approximated by a fuzzy estimator instead of by a mathematical relation.

3. Architecture of a Cascade Gray Box System for Replication in Cloud Infrastructure

A specific use case for Dynamic Semi-Physical Network can be inspired from the container allocation research explored in 2.3. In particular, concerning the problem of cloud container scaling. In the context of cloud applications, developers deploy and make available their programs, or services, on provider's servers, so that they don't have to deal with the cost, or the complexity of acquiring and managing their own server infrastructure. Often, service deployers own several clouds (or cloud-native intrinsic [44]) resources that can be commissioned to run such containers. An additional aspect influencing the cost and features for operator is the number of replicas, or mirrors of the deployed service, are available at any given time. Current solutions rely on reactivity, modifying the number of

replicas when usage hits predefined thresholds, rather than using a predictive approach [45].

Dynamic selection of the proper level of replication is paramount in achieving the promised performance and latency for the users. Here, rule-based systems are widely utilized, by filtering the available capacities, e.g., bandwidth, CPU, pod scalability, and reacting based on the current state of the service deployment. However, infrastructure owners may want to increase the productivity of their equipment (physical or virtual) by turning to AI systems to manage the replication level of their client's application. Keeping in mind that Cloud Providers typically have some Service-Level Agreement (SLA) that dictates the conditions of reliability that the provided resources have to the client, this poses a key question for the proposed AI system and the service owners:

"If the SLA is broken, how do we audit our system to investigate what went wrong?"

Generally, the state-of-the-art systems rely on Reinforcement Learning (RL) Neural Networks to handle allocation, with variations on the specific kind of RL system [46], [47]. RL models are a good fit for systems in which one wants to perform sequential decision-making, where actions affect not only immediate outcomes but also future states, as is the case when creating additional replicas over an entire cloud infrastructure. However, using these kinds of systems (that still rely on black-box models) presents crucial difficulties in answering the auditing question, when they fail.

Despite the difficulties, it is still possible to achieve some level of explainability in black-box based systems. A possible strategy is to enable a "plug-in" explainability step following the Function-as-a-Service (FaaS) approach – with an external service called the explainer. Here, for example, one may use SHAP values to provide explainations to the decisions being taken. In order to prepare such a solution, one needs to decide which part of the system (for instance, a model that allocates a set of interconnected tasks to a set of computing nodes) should be explained, e.g., a policy network. Then, the explainer assumes that the selected part behaves as a function in which each input feature's contribution to the output will be assessed. Additionally, access to a representative (ground) dataset is needed to prepare the explainer. Finally, examples to be explained can be sent by the AI-based service to the explainer service. However, the produced explanation in its raw form is purely numeric, which can be time-consuming for a human to inspect. Therefore, other ways of presenting the final SHAP output can be employed, such as an importance graph that gives insight into the relative significance of input features influencing the output, i.e. insight into which attributes where most significant when making the decision.

The described approach can be applied to a wide range of different tasks, and the instances of the explainer are easily scalable. However, apart from the previously mentioned limitations of the SHAP values, the solution may not scale well with respect to the number of the input features explained. Moreover, the behavior of the explainer is sensitive to the choice of the ground dataset. Finally, the solution can be resource-intensive, limiting its use in edge environments. From the aforementioned considerations it can be seen that before choosing such an approach some preliminary analysis focused on practical considerations should be performed.

3.1. Allocation and Replication in Cloud-Edge Infrastructure

Cloud Allocation and Replication is a very complex topic, which we will cover briefly. Although cloud servers usually share common traits, i.e., interfaces homogeneity, high availability, or built-in protection, addressing their dynamic status is still a hot research topic [48]. Escalating in difficulty, whenever other computing elements come into play, additional aspects such as network reliability, resources availability and format heterogeneity must be considered [49]. This happens especially when a complex, distributed system of computing nodes is put together in a so-called "computing continuum". There, a myriad of devices ranging from smart IoT assets to fog nodes, or edge computing equipment, are included [50], [51]. Furthermore, when restricting to the case of replication (which can be considered as an allocation of a service identical to an already-deployed service), some information about the state of the original service is available to help guide the creation of the new replica, namely the usage of the original service. In this context, we also have additional needs that influence the replication process, such as a system that determines when the replication is needed. To improve over current solutions in terms of performance, predictive elements can be of use here. Additionally, explainability is always a goal keeping in mind for situations where infrastructure failure is a possibility.

Given the compute limitations of nodes in an IoT or fog continuum, reaching a point where we reactively need to replicate a service is not desirable, since this restriction in compute means the latency to rescale imposes an unacceptable delay on the user's service.

First, some logical assumptions need to be made. In general, if one has several machines in the server side, one would not want to have a particular, busy machine have another allocation on it when there are unused machines available. Second, and specific to service replication, is resource availability. Given that we expect (or even have run against) resource limitations with the current set of replicas, we require plenty of available resources for our soon-to-be-created replicas. This resource limitation question is where predictive elements may be of use, establishing a bound on the resources the replica needs, ensuring efficient use of the limited resources on the network-wide continuum. Additionally, this predicted bound may be lower than the current level of usage, indicating a prediction to scale down the service.

There may be other metrics of relevance, like the network traffic of the machine, but this should prove sufficient for a sample example of a simple network.

$$\Delta Replica(...) = Best[C \mid C_{Future}(...), R \mid R_{Future}(...), ...] \tag{1}$$

In this case, the CPU and RAM usage are available for the variables that have been established to be relevant to make the decision. However there is no way of knowing their future value, thus, they are the intermediate functions for a decision, an the estimate of the future state of the machines. To predict these future values, one can also assume that an intermediate function may use previous measurements to estimate future usage. These values can be supplied to a final function that estimates the optimal number of replicas (or the change in number of replicas, Replica Δ), which is then enforced through K8S' Horizontal Pod Autoscaler (HPA). The sample proposal is shown in Figure 4, and is described in short in Eq. 1 where C_{Future}, R_{Future} are predictive intermediate functions for the future values of CPU and RAM usage, respectively, $Best$ is a selection algorithm that optimizes given arguments according to some criteria, and "Trigger Replication?" is a rule-based system.

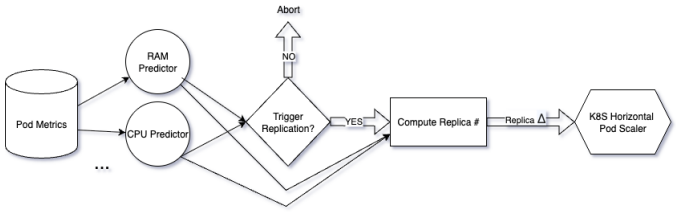


Fig. 4. System Diagram for the proposed Gray-Box Horizontal Pod Autoscaler

It should be stressed that the implementation of these intermediate functions could be a black-box model. Furthermore, systems such as the rule based system that determines if the replication should trigger could be automatically determined through Fuzzy-Rule Based System (FRBS) or even set manually by the user, including a human-in-the-loop element in the system and enhancing explainability.

3.2. Characteristics of an intermediate estimator

In the cases where the intermediate function is an estimator or model, there are some important characteristics that they must have, to maintain the desired properties in the system, such as being relatively explainable and self-contained.

First, note the benefit of having some degree of explainability on them, by knowing *a priori* the role the component plays in the overall system. The fact that the explainability is derived from knowing the intermediate function's role also hints at an interesting constraint: these intermediate functions have to be pre-trained to be included in the system. This has some benefits, like being able to individually evaluate their performance or bias, without compound error, or being able to choose different implementations, based on a performance-to-explainability or performance-to-relevance tradeoff. In our example, we train individual CPU and RAM predictors, which are locally explainable given that we know their function in the system, and that do not have compound error at this stage in the system because they are the first stage of the replica scaling process. Additionally, they can be trained independently of the rest of the system, and thus upgraded and replaced.

However, the same way that the compound or cascade error is not a factor for the individual components close to the start of the system, intermediate components that depends on others, i.e., RL systems, may imply a certain error propagation, or an accentuated dependence on the behaviour of such subcomponent. It is important to bear this in mind. in order to favour modularity of the system, and to minimize the codependency of subcomponents, as well as influencing, which kinds of subcomponent systems may be a good fit. Ideally, intermediate components should favour glass box models to minimize cascade error as much as possible. In our example, "Compute Replica #" is a mathematical, glass box system. Continuing with the themes of the initial predictors, this component is also locally explainable, with a defined function. Finally, since it depends on the previous predictors and is a mathematical (and thus not trained) model, upgrading previous parts of the system in which we depend on improves this component's accuracy, since there is no codependence between them.

3.3. The lifecycle of a component-based Semi-Physical System

The component-based perspective brings also advantages, in terms of maintaining or upgrading the system, which is a crucial factor to retain competitive advantage for the operators. Given that the Semi-Physical Network is a software-defined-architecture (SDA), it can be modified at any time, to reflect the changes in the knowledge dynamics, which are reflected in the internal structure of the system, as well as enabling very easy addition of new information, or intermediate functions, to the system. Therefore, it can easily be refined further, which is a key factor in providing speed and agility, when compared to other full black-box or monolithic systems, which would have to be rebuilt, or retrained, from scratch.

This is not only applicable to the complete system, but also to the intermediate functions, similar to a microservice architecture, a structure the software industry is quite used to. Intermediate functions can also be easily upgraded, replaced, and remade, provided the previously established constraints, especially self-containment. This comparison also extends to the development of the estimators, where several different teams could develop and own the components, as it is commonly done for their microservice counterpart. Additionally, although it may be detrimental to the explainability, it enables operators to use complete black-box models as their intermediate estimators, if they wish to, or require, the increased performance.

3.4. Traceability and Explainability in Gray Box Cascade systems vs Black-Box systems

Traceability in Dynamic Semi-Physical systems lies in the causal relations between the intermediate functions and the final output of the system. Continuing with the example, if a given service fails to uphold the SLA requirements, for example a responsiveness requirement, the initial trace may look first at the the estimated replica count (which is a glass-box model), from there, one can easily link to which individual predictor(s) (CPU, RAM, or both) failed, which is also easily debuggable because we have access to the real vs predicted values for that particular metric. Additionally, notice that this approach is independent of the particular systems used in our whole cascade: if the replica count calculation was model-driven and black-boxed, classical explainability techniques could apply to estimate the importance of features and backtrack to previous system elements.

In global terms, since the Semi-Physical system's architecture is expert-defined, it should be decidedly more understandable, as opposed to a dense NN-style architectures, where the underlying sparse structure is determined over several iterations of the nebulous gradient updates.

Furthermore, traceability becomes mire in a black-box NN system, due to the monolithic nature of that system. While explainability techniques like SHAP – which attempts to explain the effect of each input value on the final result – may be used, the phenomenon of NN coadaptation hinders traceability, due to the unforeseen internal dependencies in the network among inputs [52]. It is important to note that while the coadaptation may also be present in Semi-Physical Networks, it is somewhat self-contained in the different intermediate function estimators, rather than materialize throughout the entire system.

4. Self-Scaling: a Gray Box Cascade System for Replication Management in Kubernetes

4.1. Experimental Setup

To validate the development claims of Gray Box systems, we develop Self-Scaling, a custom cloud-native software influencing Kubernetes' features and parameters, to implement the design proposed in Fig. 4. As above mentioned, the goal is to intelligently, autonomously manage the replication of services (i.e., number of pod replicas of a service in a Kubernetes cluster) under the works performed in the Horizon Europe project aerOS [49]. For this system, we implement the notion of the Gray Box system by utilizing a cascade approach in model design, where independent components are interconnected in a left-to-right manner. System information from the pods managed by our custome software is used as primary input, in this case only CPU and RAM as starting metrics to track, as outlined in the original proposed system, though this could be expanded. Note that given we are in a Kubernetes cluster, some resources such as CPU are measured in Kubernetes' specific units, intended to harmonize the metric across heterogeneous compute nodes [53]. In keeping with the idea of having discrete components for each intermediate estimator, future timeseries models for both metrics are implemented independently of each other, utilizing ARIMA/AR-NET for timeseries regression [54] [55] [56]. This implementation uses Neural Prophet [57]. The decision component ("Trigger Replication?" in Fig. 4) is set by the service deployer in order to replicate the existing Kubernetes Horizontal Pod Autoscaler (HPA) behaviour as closely as possible, but in a more automated and intelligent way (i.e., predicting the required demand). As outlined previously, and as example of the modularity of a Gray-Box system, this component could be replaced with a FRBS module with minimal impact to other downstream components. Finally, to reduce the number of opaque or black-box models as we advance downstream in the cascade, "Compute Replica #" is implemented as a purely mathematical model based on margins, where the replica number is optimized to maintain a set resource margin available for every node in which the service is replicated, 75% in this case. To recap, Self-Scaling utilizes a predictive black-box component to estimate future usage of pods in a Kubernetes cluster, which is then handed to a rules module that can be user-set or automatically defined (and thus is a white-box in the case of being user-set, or gray-boxed in the case of an automatically defined FRBS), and finally utilizes a purely-numerical, white-box system to compute the target number of replicas. This system then communicates the new number of replicas to the Kubernetes Control Plane.

The test case chosen to evaluate the performance of Self-scaling consists of an API-based system implementing a single-task job submission service, managed by the proposed solution. This deployment is repeated afterwards and managed only with the default Kubernetes HPA. In order to simulate user access to the job queue by this service, several user profiles have been defined to simulate the anticipated behaviour interactions. User profiles are implemented as `cronjobs`, which submit computing jobs of random complexity via `curl` on a set schedule depending on the user behaviour they replicate. The number of users simulated and their schedule are defined in Table 4.1, and are described as follows:

- `burst-morning`: A high influx of jobs at the beginning of the workday.

Table 1. User simulation counts and schedules

Profile name	User count	Cron schedule
`burst-morning`	100	`0 9 * * *`
`whole-day`	50	`*/1 9-17 * * *`
`burst-evening`	80	`0 15 * * *`
`evening`	20	`*/1 15-20 * * *`

- `whole-day`: A regular influx of jobs throughout the day.
- `burst-evening`: A moderate influx of jobs at the beginning of the evening work hours.
- `evening`: An additional, constant influx of jobs during the evening work hours.

Data for both scaling managers are collected during a 24 hour period. Considering the overlap in user profiles, the modelled user influx is displayed in Fig. 5. According to its distribution, the default HPA is expected to decrease its performance or even hit the maximum number of allowed replicas with the sudden influx of users at 09:00 or 15:00, because it cannot anticipate to them. Additionally, since the jobs have varying degrees of complexity within a range, it is expected that the jobs will start to accumulate in a queue, given that the system will not be able to keep up with them, resulting in usage numbers different than the expected influx, since they cannot be processed in time before another round of jobs is submitted.

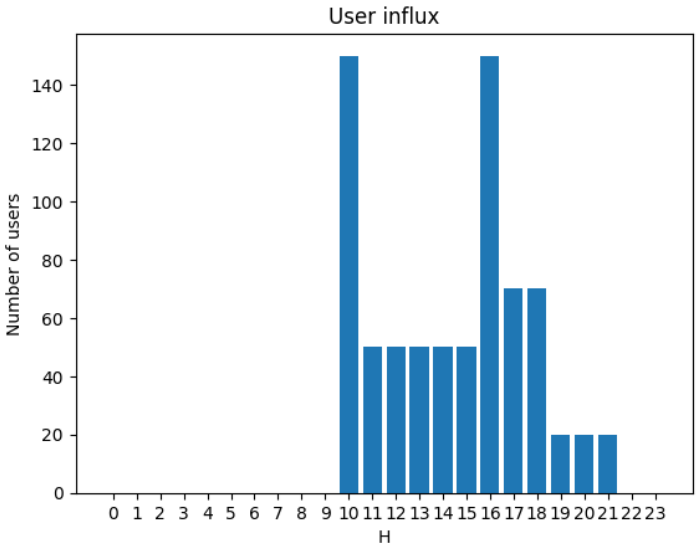


Fig. 5. Modelled user job influx

4.2. System Evaluation

Results from the tests performed in the experimental setup are presented and compared with the predictions returned by Self-scaling.

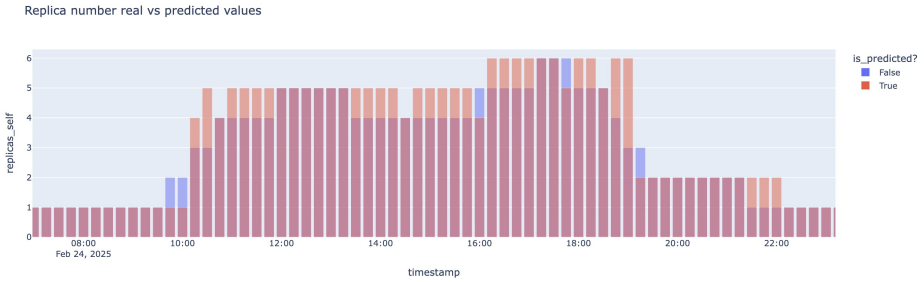


Fig. 6. Replica number in the HPA experiment (blue) vs Self-scaling predictive values (red)

In Fig. 6, the number of replicas obtained in the experiment is compared to the number predicted by Self-scaling. It is important to note the frequency of the data, which follows a default step of 15 minutes. For the default implementation of the HPA, scaling is evaluated every minute, and therefore it is not possible to see the sharp number of replicas that the HPA instantiates until reaching the required number. Regardless, this comparison demonstrates the predictive nature of Self-scaling, which for example anticipates a long-term rise in the number of required replicas at 11:00, before that increase is required later at 12:00, or notice how the predictive element accounts for the expected increase in users at 17:15.

This behaviour highlights the trade-off between resource efficiency and system readiness and reactivity, with our Self-scaling solution increasing the number of replicas way before they are actually needed. This implies that the system incurs in additional costs when the predicted usage rises above the actual one, since Self-scaling is forcing Kubernetes to spin up more replicas, but benefits the end users in terms of reactivity and system responsiveness.

4.3. An Example of Traceability and Debugging in Gray-Box Cascades

To showcase how to trace errors with Gray-Box Cascades, we return to Figure 6, in particular to the increased number of replicas in the interval between 18:45 and 19:00, or drops in the replica count in the period 14:30 to 17:45, where the standard Kubernetes HPA maintains a high number of replicas but Self-scaling has sudden reductions.

To trace these errors, given we know how the decision process is implemented, we work backwards from the scaling order to the ”Compute Replica #” module, which recall is a clear box mathematical model. Given it is clear box, we can see that it is implemented as a simple calculation that aims to maintain a maximum of 75% usage in all replicated pods, and emits a reduction if not. Since we do not know at this time which pod variable caused the replication, we work backwards to ”Trigger Replication?”, a rule-based

system, in this case user-set. The define ruleset is, in tune with the replica computation, defined to trigger if any of the predicted future variables exceeds 75%. Given this, we can now establish and diagnose the reason for each of the replication up/downscaling events. In this case, we focus on the period previously outlined, shown in Figure 7.

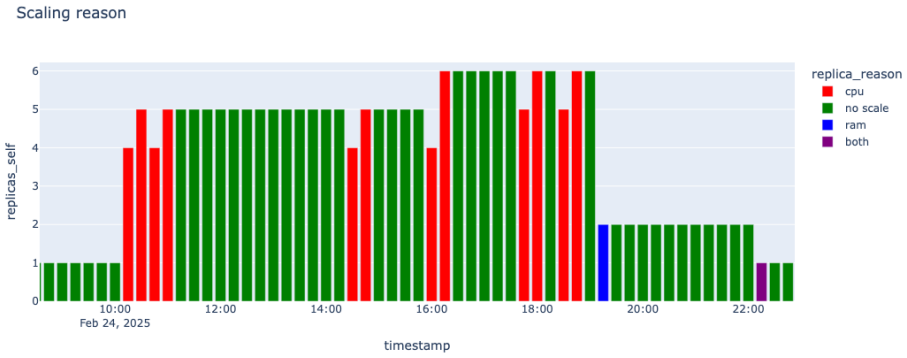


Fig. 7. Scaling reason for the Self-scaling manager

Noticing that all undesirable downscaling events were triggered because of the CPU rule, we can now look at the CPU predictor to further quantify the problem, which at this stage could still lie in any of the modules involved with CPU usage in the decision: the CPU predictor and its associated, user-set rule.

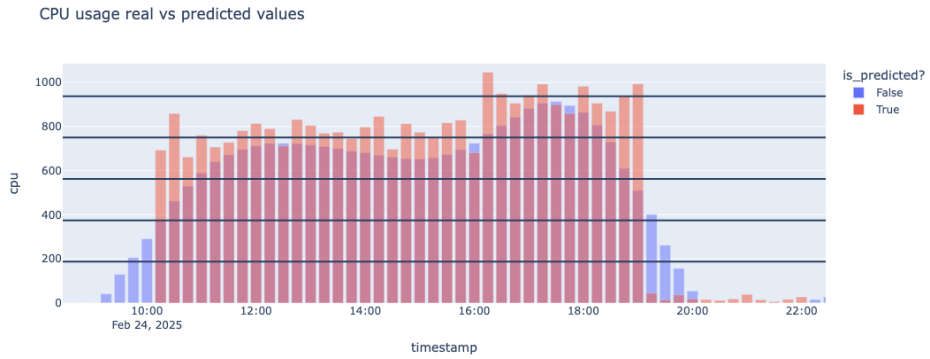


Fig. 8. CPU Module real values (blue) vs Self-scaling intermediate predictor (red). Horizontal lines marked at the up/downscale thresholds

Further inspecting and comparing the predicted CPU usage values and the rule thresholds in Figure 8, we can clearly identify points in the graph where two issues occur: the

predicted value falls under the threshold point, triggering a downscale, and it is also underneath the real observed value in blue. Given this discovery, and considering that the rule threshold worked well for all other instances, we can diagnose the issue to be a lack of accuracy in the CPU predictor module. Recall that this module is implemented as an AR-Net predictor, which is a mathematical model that uses Neural Networks and Gradient Descent to estimate the coefficients of the equation. While we could go further back in the Cascade, given that AR-Nets are somewhat interpretable, with this module being at the very start of the cascade we find no need. A system integrator may now take action having identified the root of the issue, for example expanding the data used to train the model, or deciding for a more complex black-box now that the need for performance is justified.

5. Limitations of the considered approach

A first obvious roadblock is the need to determine an overall architecture for the Semi-Physical Network. Since it would be informed by the knowledge of the field, a lack of such knowledge means an inaccurate estimate of the relationships of the intermediate functions, which may result in results worse than using a completely black-boxed model. Providing incorrect, or unnecessary, intermediate functions may also be detrimental to the overall system, as it would be fed misleading variables. As a result explainability is tightly coupled with the architecture which may be considered as harder to maintain in systems that may require any adaptations with time. On the other hand, the alternative approach with an external explainability service is easier to maintain as more detached from the model itself, however more difficult to return a meaningful result.

Furthermore, due to the inherent move to a gray-box approach, the approach is also constrained for the industry applications with strict explainability constraints, stemming from the legal or the operational requirements. As previously outlined, the proposed style of architecture only obtains *more explainable* models, as opposed to completely explainable ones. Here, however, it should be noted that requirements related to the degree of explainability should be collected at the beginning and considered during the design of the architecture. The gray-box approach may prove still valid in numerous cases.

Finally, traceability is not guaranteed, depending on the different interconnections between the subcomponents. While it is a reasonable expectation to use more explainable "aggregators" (the components that combine the inputs of intermediate functions) the architecture by itself does not guarantee traceability when using one, due to the presence of coadaptation internal to the aggregator component.

6. Concluding remarks

Drawing from the AI explainability theory, achieving transparency on where a model failed or behaved in a suspicious way that needs to be investigated, implies several degrees of difficulty. Current approaches are mostly limited to SHAP weights allocation and applying gray-box modelling to a certain extent. With the advent of Dynamic Semi-Physical Architecture (which authors leverage from different sources), one step beyond is envisioned.

This work proposes a cascading approach, that bids for gradually increasing the "transparency" of an AI system to be more explainable for the human in charge of understanding potential faults in the result. It does so by suggesting the introduction of intermediate aggregators, rooting on previously known relationships between inner blocks of the architecture. That way, a Semi-Physical Architecture incrementally approaches gray-box fragments, which will, in turn, be more traceable and dynamically analysed, compressing and reducing the scope of black-box type models (such as NN or RL) in those systems.

A direct conclusion is that, while this appears as suggestive roadmap, it must be validated with real test cases. In this regard, authors have proposed a simple but quite illustrative case where cloud-native services require intelligent policies for replica creation (based on some ML process). Although the results are good, authors are already working to substantiate the claim via experimenting on the proposed example of service and resource allocation possibilities in an IoT-edge-cloud environment. The results of these experiments, performed within the scope of various research projects (such as aerOS and O-CEI), will be reported in subsequent publications. Moreover, the underlying software being produced will be released under Open Source licenses.

Acknowledgments. This research was funded by the European Commission, under the Horizon Europe project aerOS, grant number 101069732. Work a result, in part, of a bilateral collaborations between the Polish Academy of Sciences and the Bulgarian Academy of Sciences, and between the Polish Academy of Sciences and the Romanian Academy. Tests were supported by Wojciech Buchwald.

References

1. W. Samek and K.-R. Müller, "Towards explainable artificial intelligence," in Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Eds. Cham, Switzerland: Springer International Publishing, 2019, pp. 5–22. ISBN: 978-3-030-28954-6.
2. R. Dwivedi, D. Dave, H. Naik, et al., "Explainable ai (xai): Core ideas, techniques, and solutions," ACM Comput. Surv., vol. 55, no. 9, Jan. 2023, ISSN: 0360- 0300. DOI: 10 . 1145 / 3561048. [Online]. Available: https://doi.org/10.1145/3561048
3. A. Rai, "Explainable AI: From black box to glass box," Journal of the Academy of Marketing Science, vol. 48, pp. 137–141, 2020
4. I. Ahmed, G. Jeon, and F. Piccialli, "From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where," IEEE Transactions on Industrial Informatics, vol. 18, no. 8, pp. 5031–5042, 2022. doi: 10.1109/TII.2022.3146552.
5. K. E. Mokhtari, B. P. Higdon, and A. Başar, "Interpreting financial time series with SHAP values," in Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, 2019, pp. 166–172.
6. M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," arXiv preprint arXiv:1606.05386, 2016.
7. E. Galinkin, "Robustness and usefulness in AI explanation methods," arXiv preprint arXiv:2203.03729, 2022.
8. A. Gramegna and P. Giudici, "SHAP and LIME: An evaluation of discriminative power in credit risk," Frontiers in Artificial Intelligence, vol. 4, p. 752-558, 2021.
9. V. Hassija, V. Chamola, A. Mahapatra, et al., "Interpreting black-box models: A review on explainable artificial intelligence," Cognitive Computation, vol. 16, no. 1, pp. 45–74, 2024.

10. C. Rudin and J. Radin, "Why are we using black box models in AI when we don't need to? A lesson from an explainable AI competition," Harvard Data Science Review, vol. 1, no. 2, pp. 10–1162, 2019.
11. C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," Nature Machine Intelligence, vol. 1, no. 5, pp. 206–215, 2019.
12. K. Arendt, M. Jradi, H. R. Shaker, and C. Veje, "Comparative analysis of white-, gray- and black-box models for thermal simulation of indoor environment: Teaching building case study," in Building Performance Analysis Conference and SimBuild: Co-organized by ASHRAE and IBPSA-USA, ASHRAE, 2018, pp. 173–180.
13. D. Thakker, B. K. Mishra, A. Abdullatif, S. Mazumdar, and S. Simpson, "Explainable artificial intelligence for developing smart cities solutions," Smart Cities, vol. 3, no. 4, pp. 1353–1382, 2020, ISSN: 2624-6511. doi: 10.3390/smartcities3040065. [Online]. Available: https://www.mdpi.com/2624-6511/3/4/65.
14. M. Kunjir and S. Babu, "Black or white? How to develop an autotuner for memory-based analytics," in Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD/PODS '20, ACM, May 2020. doi: 10.1145/3318464.3380591. [Online]. Available: http://dx.doi.org/10.1145/3318464.3380591..
15. Y. Oussar and G. Dreyfus, "How to be a gray box: Dynamic semi-physical modeling," Neural Networks, vol. 14, no. 9, pp. 1161–1172, 2001.
16. T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," arXiv preprint arXiv:1902.09574, 2019.
17. U. Forssell and P. Lindskog, "Combining semi-physical and neural network modeling: An example of its usefulness," IFAC Proceedings Volumes, vol. 30, no. 11, pp. 767–770, 1997. doi: 10.1016/S1474-6670(17)42938-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017429387.
18. J. Adou, A. Brou, and B. Porterie, "Modeling wildland fire propagation using a semi-physical network model," Case Studies in Fire Safety, vol. 4, pp. 11–18, 2015.
19. L. Haviez, R. Toscano, M. El Youssef, S. Fouvry, G. Yantio, and G. Moreau, "Semi-physical neural network model for fretting wear estimation," Journal of Intelligent & Fuzzy Systems, vol. 28, no. 4, pp. 1745–1753, 2015.
20. L. A. Zadeh, "Fuzzy logic," Computer, vol. 21, no. 4, pp. 83–93, 1988.
21. L. Magdalena, "Fuzzy rule-based systems," in Springer Handbook of Computational Intelligence, 2015, pp. 203–218.
22. P. Mishra, "Model explainability for rule-based expert systems," in Practical Explainable AI Using Python: Artificial Intelligence Model Explanations Using Python-based Libraries, Extensions, and Frameworks, Springer, 2021, pp. 315–326.
23. J. M. Mendel and P. P. Bonissone, "Critical thinking about explainable AI (XAI) for rule-based fuzzy systems," IEEE Transactions on Fuzzy Systems, vol. 29, no. 12, pp. 3579–3593, 2021.
24. L. Sánchez, I. Couso, and M. González, "A design methodology for semi-physical fuzzy models applied to the dynamic characterization of LiFePO4 batteries," Applied Soft Computing, vol. 14, pp. 269–288, 2014. doi: 10.1016/j.asoc.2013.03.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S156849461300135X.
25. I. Rodríguez-Fdez, M. Mucientes, and A. Bugarín, "Fruler: Fuzzy rule learning through evolution for regression," Information Sciences, vol. 354, pp. 1–18, 2016.
26. L. Wang, Z. Mu, and H. Guo, "Fuzzy rule-based support vector regression system," Journal of Control Theory and Applications, vol. 3, no. 3, pp. 230–234, 2005.
27. R. Vaño, I. Lacalle, P. Sowiński, R. S-Julián, and C. E. Palau, "Cloud-Native Workload Orchestration at the Edge: A Deployment Review and Future Directions," Sensors, vol. 23, no. 4, pp. 2215, 2023. doi: 10.3390/s23042215.
28. Kubernetes, "Horizontal Pod Autoscaler." Kubernetes, kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/.

29. F. -D. Eyitemi and S. Reiff-Marganiec, "System Decomposition to Optimize Functionality Distribution in Microservices with Rule Based Approach," 2020 IEEE International Conference on Service Oriented Systems Engineering (SOSE), Oxford, UK, 2020, pp. 65-71, doi: 10.1109/SOSE49046.2020.00015.
30. N. Liu, Z. Li, J. Xu, et al., "A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning," in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2017, pp. 372–382.
31. E. Barrett, E. Howley, and J. Duggan, "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud," Concurrency and Computation: Practice and Experience, vol. 25, no. 12, pp. 1656–1674, 2013.
32. Z. Ding, S. Wang and C. Jiang, "Kubernetes-Oriented Microservice Placement With Dynamic Resource Allocation," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1777-1793, 1 April-June 2023, doi: 10.1109/TCC.2022.3161900.
33. "aerOS project." https://aeros-project.eu (accessed May 25, 2024).
34. R. S-Julián, I. Lacalle, R. Vaño, F. Boronat, and C. E. Palau, "Self-* Capabilities of Cloud-Edge Nodes: A Research Review," Sensors, vol. 23, no. 6, pp. 2931, 2023. doi: 10.3390/s23062931.
35. X. He, H. Xu, X. Xu, Y. Chen and Z. Wang, "An Efficient Algorithm for Microservice Placement in Cloud-Edge Collaborative Computing Environment," in IEEE Transactions on Services Computing, doi: 10.1109/TSC.2024.3399650.
36. P. Ngatchou, A. Zarei, and A. El-Sharkawi, "Pareto multi objective optimization," in Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, 2005, pp. 84–91. doi: 10.1109/ISAP.2005.1599245.
37. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012

Jorge Jiménez-García is telecommunications engineer with more than 5 years of research experience in the Biomedical Engineering Group (University of Valladolid, Spain). I have worked on the development of machine/deep learning algorithms to detect and assess the severity of various diseases. To achieve that goal, I gained expertise on image processing and analysis (retina/fundus images) as well as biomedical signal processing (cardio-respiratory signals). My research interests focus on AI applications in health care.

Ignacio Lacalle Úbeda (male) is a Researcher and Teaching Assistant at the Universitat Politècnica de València (UPV), a public University in the South-East of Spain. Ignacio is a Telecommunications Engineer (2014) from UPV and received his Ph.D. (Dr. Ing.) in 2022. The expertise of Ignacio is mainly rooted in the Internet of Things field, having participated in 10 national and EU research projects such as HE aerOS, MSCA AIAS, HE MISSION, H2020 ASSIST-IoT and others, related mainly to distributed systems, interoperability, added value services, data processing, cybersecurity and manageability, inter alia. Ignacio has performed various roles in those projects (ranging from Developer to Community Manager or Project Manager). In addition, most of those projects focused on applying IoT-related innovations, particularly in the fields of energy, maritime ports, manufacturing, agriculture, and others. Ignacio has authored or co-authored more than 30 publications in relevant journals and conferences.

Pawel Szmeja is an academic researcher from Polish Academy of Sciences. The author has contributed to research in topics: Interoperability & Semantic interoperability. The

author has an hindex of 13, co-authored more than 40 publications. Previous affiliations of Pawel Szmeja include Systems Research Institute & University of Warsaw.

Katarzyna Wasielewska-Michniewska is a researcher and practitioner with over 15 years of experience in scientific and commercial projects. Graduated from Warsaw University of Technology, Faculty of Mathematics and Information Technology, defended her PhD thesis in Systems Research Institute of Polish Academy of Sciences. Her initial scientific interests were agent programming, distributed systems, decision support and Internet of Things. With time, they evolved towards semantic technologies and (frugal/explainable) AI. Participant of H2020 projects: INTER-IoT, Assist-IoT, aerOS. She has published over 45 papers.

Przemysław Hołda, as a Computer Science student, was recognized as an award winner for "The Best Work in the field of Public Administration Digitization" by the Ministry of Digital Affairs and for "Beyond the Horizon" by the Polish Information Processing Society, in the "Miedzyuczelniany Konkurs Młodych Mistrzów 2022" contest. Some of his scientific activities were financially supported by a grant named "Najlepsi z Najlepszych 4.0" from the Ministry of Science and Higher Education. He attended the 4th Summer School on Cyber Physical Systems and Internet of Things (SS-CPSIoT'23) and the 23rd European Agent Systems Summer School (EASSS'23). He also had the opportunity to give guest lectures on Agent Systems and Applications at the Warsaw University of Technology. His current areas of interest are distributed systems and artificial intelligence.

Maria Ganzha is working as Associate Professor at the Warsaw University of Technology. She has received her M. A. and Ph. D. in mathematics at the Moscow State University in Russia, and a Doctor of Science degree in computer science from the Polish Academy of Sciences. Maria has published nearly 200 scientific articles, is editor in chief of a book series and a member of the editorial boards of 3 journals. She has been invited to the program committees of over 250 conferences. Her scientific interest is focused on agent-based technology, semantic technology, and machine learning. She is Technical Coordinator of EU project ASSIST-IoT, and principal investigator of SRIPAS team in another EU project aerOS. She is a President of Mazovia Branch of Polish Information Processing Society.

Carlos E. Palau Salvador is Full Professor in the Escuela Tecnica Superior de Ingenieros de Telecomunicación at the Universitat Politècnica de València. He has more than 20 years of experience in the ICT research area in the area of Networking. He has collaborated extensively in the RD of multimedia streaming, security, networking and wireless communications for government agencies, defence and European Commission. He has been the main UPVLC researcher in several funded projects, which has funded this work. He is author and co-author of more than 120 research papers and member of the TPC of several IEEE, ACM and IFIP conferences. He is Senior Member of IEEE. On the academical side he has been supervisor of 4 PhD Thesis, currently supervising three more and more than 100 MSc Thesis. Main researcher in TEMPUS and Asia Link projects with educational purposes.

Costin Bădică b. February 19, 1966, in Craiova, Dolj. He is a Professor Doctor Engineer at the University of Craiova, Faculty of Automation, Computers, and Electronics. Since

2012, he has been the Director of the ”Constantin Belea” Doctoral School within the Faculty of Automation, Computers, and Electronics at the University of Craiova. As of 2024, he is an Associate Member of the Academy of Technical Sciences of Romania (ASTR), Section: Information and Communications Technology, Computers, and Telecommunications. He serves as Co-Chair of the Technical Committee on Computational Collective Intelligence for the IEEE SMC Society. This committee received the ”Most Active SMC Technical Committee Award” in Cybernetics from the IEEE SMC Society in both 2021 and 2024. He is a member of the editorial boards for the following journals: Computer Science and Information Systems (ComSIS), ISSN 1820-0214; System Theory, Control and Computing Journal, ISSN: 2810–4099; Cybernetics and Information Technologies (CIT), The Journal of the Institute of Information and Communication Technologies of the Bulgarian Academy of Sciences, eISSN 1314-4081; Simulation Modelling Practice and Theory, ISSN 1569-190X (WoS); Vietnam Journal of Computer Science, ISSN (print): 2196-8888 — ISSN (online): 2196-8896. He is a founding member of the international conference IDC (Intelligent Distributed Systems) – 2007. His areas of interest include: artificial intelligence, multi-agent systems, distributed systems, and software engineering.

Stefka Fidanova is a Professor at the Institut of Information and Communication Technologies. The author has an hindex of 13, co-authored more than 240 publications.

Marcin Paprzycki works as an Associate Professor in the Systems Research Institute, Polish Academy of Sciences. The author has an hindex of 37, co-authored more than 750 publications.

Received: April 25, 2025; Accepted: December 22, 2025.

Computer Science and Information Systems 23(1):585–601 https://doi.org/10.2298/CSIS250418083N

Audio Feature-based User Profiles for Personalized Music Recommendation: A Dataset-driven Evaluation*

Ionuţ-Dragoş Neremzoiu and Andreea Liliana Bădică

University of Craiova, Craiova, Romania
neremzoiu.ionut.k4c@student.ucv.ro
andreea.badica121@yahoo.com

Abstract. In this paper, we propose an experimental content-based track recommendation system, which relies on an aggregated features which considers audio feature values from Spotify Data Catalog, track lyrics and popularity ratings. As system evaluation protocol, we evaluate how relevant the top-5 recommended tracks are to the user profile, which is based on average audio feature (danceability, energy, valence, loudness, instrumentalness, liveness, speechiness and acousticness) values from the user's listening history. Based on the evaluation results, we come to the conclusion that the recommended tracks are similar to the user's profile.

Keywords: recommender systems, pca, k-means, aggregated feature, mean absolute error

1. Introduction

Ever since the rise of music streaming services such as Spotify or Youtube Music, recommender systems have played a huge role in promoting new songs. These systems rely on various types of data, such as geographic user data, listening history, list of liked songs, etc. Recommendation strategies are generally partitioned into three major approaches:

- *Collaborative filtering* that recommends items to users based on how other users with similar preferences and behavior have interacted with that item [14].
- *Content-based* that uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback [10] .
- *Hybrid* that combines two or more recommendation strategies in different ways to benefit from their complementary advantages [20].

One noteworthy content-based music recommendation system that focuses on the acoustic similarity of musical compositions rather than relying on metadata such as genre, artist or user ratings, is presented in [15]. This paper showcases how acoustic analysis and spectrogram-based deep learning can enhance music recommendations without relying on user data.

A noteworthy recommendation system for music tracks using collaborative filtering and two optimization methods: *dataset rescaling* and *automatic halting* is presented in

* This paper is an extended version of our previous work published in INISTA 2024 [13]. The extensions include a reduced but more useful version of a dataset that was previously used, a new evaluation protocol, new results derived from the evaluation and new discussions in terms of conclusions, which significantly enhance the depth and scope of the original study.

[9]. These methods employed various parameters to address the performance issue while retaining an acceptable accuracy. Those parameters are the following:

- A user similarity threshold (θ),
- A fraction (τ) of the full dataset randomly sampled for processing used for generating song recommendations,
- A hash-table representation to store sparse user-song interactions efficiently.

As for the hybrid approach to music recommendation, [11] combines content-based filtering with the K-means clustering algorithm. The hybrid approach aimed to improve the relevance of music recommendations and to address the cold-start problem, which occurs when a system lacks enough data about a new user or new content.

Our work proposes a new *content-based* recommendation approach based on a specially customized dataset that we created by aggregating audio features extracted from Spotify Catalog, i.e. danceability, energy, valence, loudness, instrumentalness, liveness, speechiness, acousticness, duration, tempo, as well as lyrics and popularity ratings of songs. We proposed the aggregation of these features into a new dataset in order to enhance user experience by recommending songs that better match user preferences in terms of audio features, lyrics and popularity ratings. Moreover, before recommending the tracks, we also classify the tracks from our custom-made dataset in numbered labels using K-Means algorithm as [11]. The way we choose the number of labels is highlighted in Section 3.

This paper is a significant extension and improvement of our preliminary work [13] by strengthening our experimental evaluation using a more appropriate set of test users. For every test user, we assume a user profile. This profile is based on the average values of key audio features — danceability, energy, valence, loudness, instrumentalness, liveness, speechiness, and acousticness — calculated from all tracks[1] the user played at least three times. Using this profile, we compare the averaged feature values from the user's listening history with the corresponding feature values of the top 5 recommended tracks.

The paper is structured as follows:

- Section 1 introduces the context, motivation and related works, and summarizes the content of the paper.
- Section 2 introduces the dataset description and preparation.
- Section 3 approaches the experimental system design.
- Section 4 addresses both the evaluation protocol and results.
- The last section presents our conclusions and points future work.

2. Dataset Description and Preparation

We created a novel and more suitable dataset to configure our recommendation system by integrating the dataset from the work of Ronak Doshi et al. [17] (Github dataset, 10,449 track entries) and the dataset created by Xiaomeng Zhang [19] (Jovian dataset, 174,389 track entries). We extracted only the Spotify Data Catalog features found in both datasets and we applied a preprocessing pipeline consisting of *data filtering*, *data cleaning* and

[1] The terms *track* and *song* are used interchangeably.

Table 1. Track dataset. Except lyrics and label, the field descriptions are detailed from [7, 2]

Field Name	**Field Description**
Name	Track name
Artist	The name of an artist or multiple artists separated by commas.
Duration	The duration of a track in milliseconds.
Popularity	Rating value between 0 (least popular song) and 100 (most popular song).
Key	The estimated overall key of the track. Integers map to pitches using standard "Pitch Class notation" (E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on).
Acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic.
Danceability	Track suitability (on a scale from 0 to 1) for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength and overall regularity.
Energy	A measure from 0.0 to 1.0 representing intensity and activity of a track.
Instrumentalness	Measure of lack of vocals (from 0 to 1). "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal".
Liveness	Detects the presence of an audience in the recording. Higher liveness values (especially above 0.8) represent an increased probability that the track was performed live.
Loudness	The overall loudness of a track in decibels (dB).
Speechiness	Measure of the presence of exclusively spoken words (speech-like). Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
Valence	Describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
Tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece, and derives directly from the average beat duration.
Mode	Indicates the modality (major or minor) of a track.
Lyrics	Track lyrics collected through Basile Bruneau's [4] JSON API and web-scraping (*Genius*, *MusixMatch* and *AZLyrics*)
Label	The cluster label a track is assigned to.

joining, resulting a final dataset of 180,220 unique tracks. The description of the resulting dataset is provided in Table 1.

Data filtering aimed to remove duplicates. We found that only the Jovian dataset had 2159 track duplicates according to the *track id* feature. for identifying duplicates, which were ultimately removed. Thus, from this dataset we kept the remaining 172,230 tracks.

Data cleaning aimed to harmonize the format and syntax of the external representation of data. In the Jovian dataset, the artist names involved in singing a track are written between ' or " into a comma-separated list with brackets. The artist names from the Github dataset don't have any ' or " and are not included in a list with brackets. To make a simple and consistent representation, we chose the data format from the Github dataset for the artist names. For instance, the initial value *['Mamie Smith', 'The Harlem Trio']* will become *Mamie Smith, The Harlem Trio* after cleaning.

When joining the datasets by taking into account track id, we firstly checked whether the track id occurs in both datasets. Then, for each track id shared in both datasets before joining, the means of all non-categorical data fields (*acousticness*, *danceability*, etc.) were computed and saved into the final dataset. We managed to gather the lyrics of 131,651 songs out of the 180,220 of the entire initial dataset. The lyrics data was further preprocessed to compute Term Frequency – Inverse Document Frequency (TF-IDF) coefficients (see next section).

3. Experimental System Design

3.1. System overview

This section describes in further details each process from Figure 1 that contributes to the functionality of the system. For simplicity purposes, we categorize all the processes involved in three main phases: the *data loading*, the *preprocessing phase*, and the *recommendation algorithm*.

The *data loading* phase simply represents the process in which we load our custom-made dataset introduced in Section 2.

The preprocessing phase aims to cluster the songs based on their aggregated audio features (principal components) and to evaluate the importance of each word in the lyrics of each song relative to the collection of songs with lyrics (a corpus). The steps of the preprocessing phase are the following: *Feature Selection*, *Min-max Scaling*, *Dimensionality reduction*, *Clustering*, *Train-test split*, *Track label classification*, and *Lyrics TF-IDF matrix computation*.

The proposed recommendation algorithm was extended based on the proposal from [17] by aggregating the similarity of audio features as detailed in 3.3, the popularity rating and the similarity of the lyrics. The initial algorithm from [17] uses two major sources for computing the similarity measure between songs:

- the principal component analysis-based Euclidean distance between the song used for generating recommendations and a song belonging to the same label as the former song.
- the Spotify track popularity rating of a song belonging to the same label as the song used for generating recommendations.

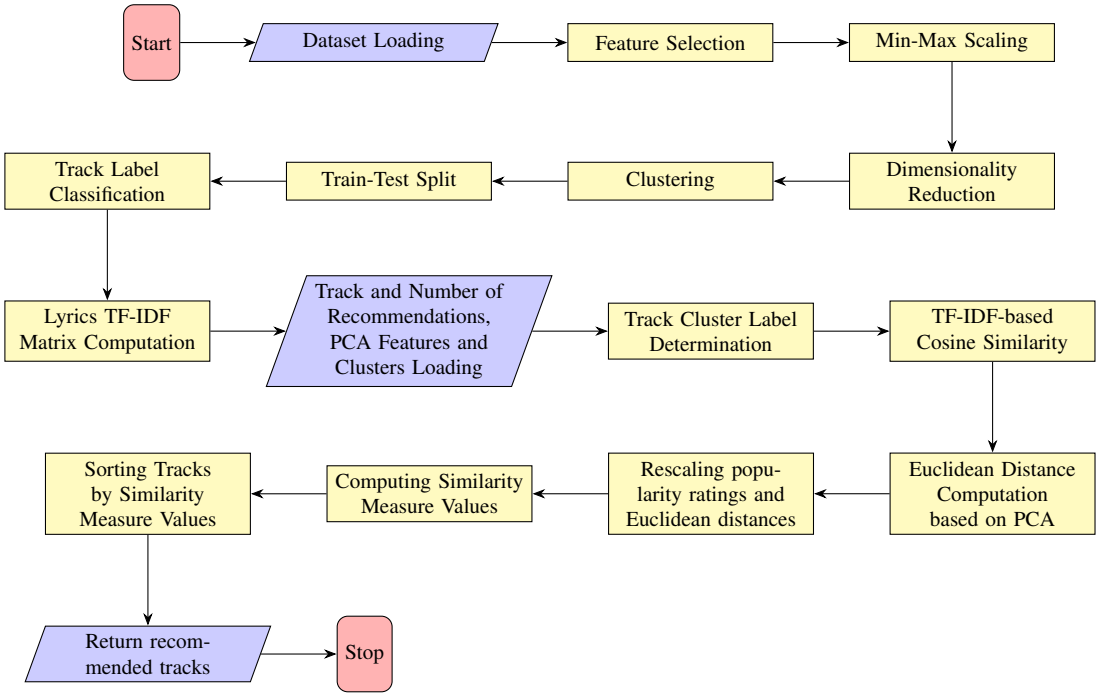


Fig. 1. *Workflow of the recommendation system.* The preprocessing phase involves all processes from *Data Loading* to *Lyrics TF-IDF Matrix Computation*, whereas the recommendation algorithm the next stage, which involves reading its necessary inputs until returning the recommendations

The similarity measure proposed by [17] is computed using Equation (1):

$$S(t, t') = \frac{P(t')}{(D(t, t'))^2} \tag{1}$$

where:

- $S(t, t')$ is the similarity measure
- $P(t')$ is the rating value of a current track t' belonging to the same label as the input track t
- $D(t, t')$ is Euclidean distance between tracks t and t'

[17] uses Principal Component Analysis (PCA) on the audio Spofity features for dimensionality reduction and *K-Means* for clustering the tracks in 4 clusters, which we also consider for the preprocessing in our dataset. However, unlike [17], we investigated in [13] a wider range of values for two parameters: i) the number of principal components that satisfy a threshold ranged between 90-95% cumulative variance and ii) the number of clusters for *K-Means Clustering*. Moreover, we adopted the parameter values obtained by *Elbow method*, *Silhouete method* and *Calinski-Harabasz Index* as evaluation measures for the optimal number of clusters.

We updated the computation of the similarity measure from (1) by using, apart from PCA features and popularity rating, the similarity measure for lyrics detailed further in Section 2. The motivation is that we observed that the initial proposal from (1) weights the popularity rating too much to the point where even if a track used for recommendations is similar to another track from the same group (based on PCA features and audio characteristics), if that track has a low popularity rating, the overall similarity measure value will be too low, if not the lowest, thus hindering its selection for recommendation.

The steps of the *recommendation algorithm* involved the determination of: *Track cluster label*, *TF-IDF-based cosine similarity*, *Euclidean distances based on the principal components*, *Rescaling popularity ratings and Euclidean distances*, and, respectively, *Track Ranking based on Similarity Measure Values*. Apart from that, it is important to highlight the fact that our recommendation algorithm has a track and the number of recommendations as set of inputs and the number of PCA features and clusters as set of hyper-parameters, which are used for evaluation in Section 4.

3.2. Preprocessing Stage

Once we load the track dataset, we perform the*feature selection* by categorizing each feature based on its data type and utility:

- **Non-categorical features**: Numerical features suitable for analysis (e.g., *danceability*, *energy*, *loudness*).
- **Categorical features**: Non-numerical or descriptive attributes (e.g., *artist*, *song title*), retained only for presentation purposes.

Categorical features are excluded from further processing. The selected **non-categorical features** used in subsequent steps are:
danceability, *energy*, *loudness*, *speechiness*, *acousticness*, *instrumentalness*, *liveness*, *valence*, *tempo*, and *duration*. Once we choose only the non-categorical features, we apply *Min-max scaling* to the selected features to map them in the range $[0, 1]$.

The main idea of PCA is to reduce the dimensionality datasets containing a large number of interrelated features, while retaining as much as possible of the variation present in the dataset. PCA accomplishes this by transforming the data into uncorrelated principal components (PCs), ordered such that the first few PCs capture the majority of the dataset's variability. [12].

To select the number of principal components, we compute the individual and *cumulative explained variance* across all possible component counts. In our case, the maximum number of principal components is 10, because the total number of selected features is 10. Table 2 displays both the individual explained variance (percentage of coverage per principal component) and the cumulative explained variance (total percentage up to a given number of components). Using these measures, we assess the variability of the original data in each component or a selected number of components and depending on how much of the initial data we are willing to discard to reduce space complexity, we select the most appropriate number to achieve the proposed threshold.

Table 2. Explained Variance of Principal Components

Component/Number of Components	% Explained Variance	% Cumulative Explained Variance
1	43.006%	19.900%
2	19.9%	62.905%
3	13.279%	76.185%
4	7.538%	83.723%
5	6.788%	90.510%
6	3.408%	93.919%
7	3.098%	97.016%
8	2.408%	99.424%
9	0.444%	99.869%
10	0.131%	100%

We used K-Means for clustering tracks given its simplicity, scalability, convergence guarantees and ability to handle diverse cluster shapes (e.g., elliptical) [8, 18]. While K-Means is widely used for clustering, it has the following limitations [8]:

1. **Choosing the Number of Clusters**: To address this, we employed the *Elbow Method*, *Silhouette Score*, and *Calinski-Harabasz Index* as evaluation measures to determine the optimal cluster count.
2. **Sensitivity to Initial Values**: We mitigated this issue using the *K-Means++* initialization algorithm [3], which improves cluster stability.
3. **Scalability with High Dimensions**: This limitation is less relevant here, as we applied *PCA* for dimensionality reduction.
4. **Handling Varying Sizes and Densities**: A key limitation of K-Means is that it works best with evenly sized, ball-shaped clusters. Since real data often violates these assumptions, we considered this factor in our methodology.

To evaluate the optimal number of clusters used for K-Means, we considered the range of [2, 10] to choose from and the following evaluation measures: *Elbow method*, *Silhouette coefficient* and *Calinski–Harabasz index*.

The idea behind the *Elbow method* is that the explained variability changes rapidly for a small number of clusters and then slows down, resulting in an elbow shaped curve. The point at which the so-called elbow is formed indicates the number of clusters that should be optimal for a given clustering algorithm [16].

The *Silhouette coefficient* [16] is slightly stricter than the *Elbow method*, but it is better at identifying relationships within and between groups. This measure indicates whether individual points are correctly assigned to their clusters. When using this coefficient, we can interpret the measure $S(i)$ of a point i as follows:

- $S(i)$ close to 0 means that the point is between two clusters.
- If $S(i)$ is close to -1 then the target point should be in a different cluster.
- If $S(i)$ is close to 1 then the point belongs to the "correct" cluster.

The *Calinski–Harabasz index* is defined based on the assumption that clusters are very compact and are sufficiently distant from each other to form a good distribution. The index is calculated by dividing the variance of the sums of squares of the distances of individual objects to their cluster centre by the sum of squares of the distance between the cluster centres. The higher that index value, the better the division [16].

In the *track label classification* stage, we implemented a k-NN model to categorize new songs according to the K-Means cluster assignments. The model takes as input the dimensionally-reduced PCA features generated in the preprocessing stage.

For the testing phase of the classification model, we performed an 80-20 train test split of the dataset, which resulted in 144,176 songs for training set and 36,044 songs for testing set. We evaluated the performance for the number of neighbours k on a $[1, 30]$ range and we decided to choose $k = 25$, because it has a good enough accuracy, since only 412 out of 36,044 tracks had inaccurate labels, when compared to their actual label values.

In the *Lyrics TF-IDF Matrix Computation* stage, we compute the $TF\text{-}IDF$ values of all terms (or words) of the lyrics of each track. Let $W = \{w_0, w_1, \ldots, w_{|W|-1}\}$ be the set of words from the lyrics across all tracks and $T = \{t_0, t_1, \ldots, t_{|T|-1}\}$ the set of tracks grouped in a label determined in *Clustering* stage, where the track t, based on which recommendations are generated, is also found. According to [1], the term frequency tf is defined as the number of occurrences of a word w in the lyrics of a track l_t and the inverse document frequency IDF, which takes into account the document frequency of a word $df(w)$ (the number of occurrences of the word w across the set of track lyrics) and the total number of tracks $|T|$, is defined as in equation (2):

$$\begin{aligned} &idf : W \to [1, 1 + \ln\left(\frac{1 + |T|}{2}\right)] \\ &df : W \to [1, |T|] \\ &idf(w) = 1 + \ln\left(\frac{1 + |T|}{1 + df(w)}\right) \end{aligned} \tag{2}$$

Once all $tf(w, t)$ values of $TF\text{-}IDF$ matrix are computed for $\forall w, t,\ 0 \leq w < |W|, 0 \leq t < |T|$, they will be normalized through Euclidean norm, as depicted in equation (3):

$$TF\text{-}IDF_{|T|\times|W|} = \left[\frac{tf\text{-}idf(w,t)}{\|TF\text{-}IDF(t)\|}\right]_{0\le w<|W|, 0\le t<|T|}$$
$$\|TF\text{-}IDF(t)\| = \sqrt{\sum_{w=0}^{|W|-1}(tf\text{-}idf(w,t))^2} \tag{3}$$

$\|TF\text{-}IDF(t)\|$ is the Euclidean norm of the $|W|$-dimensional $TF\text{-}IDF$ vector of $\forall t \in T$.

3.3. Recommendation algorithm

The first step of the recommendation algorithm is to determine the cluster label of the input track.

Next, we identify the cluster label for the input track t. Then, using the TF-IDF matrix (created during preprocessing) which includes track t and all other tracks in the same cluster, we calculate lyrics similarity. This is done by computing cosine similarity (Equation 4) between track t's TF-IDF vector and all other tracks' vectors in its cluster.

$$\cos(t,t') = \frac{TF\text{-}IDF(t)\cdot TF\text{-}IDF(t')}{\|TF\text{-}IDF(t)\|\cdot\|TF\text{-}IDF(t')\|} \tag{4}$$

Then, we determined the similarities between each pair of tracks t and t' based on their PCA features, denoted by $PCS(t,t')$, as shown in equation (7). We then used Euclidean distance $D(t,t')$ computed based on the best p PCA features of tracks. The resulting distances were "min-max" scaled into $[0,1]$ using equations (5) and (6).

Finally, the similarity $S(t,t')$ between two tracks t and t' is computed by aggregating three values (see equation (9): i) the similarity of tracks based on their acoustic features, captured as $PCS(t,t')$; ii) the cosine similarity of lyrics of tracks t and t' (see equation (4)); iii) the scaled popularity to $[0,1]$ of track t.

$$D\colon T\times T\to[0,\sqrt{p}]$$
$$D(t,t') = \sqrt{\sum_{k=0}^{p-1}(PC_k(t)-PC_k(t'))^2} \tag{5}$$

$$D_{\text{MinMax}}\colon T\times T\to[0,1]$$
$$D_{\text{MinMax}}(t,t') = \frac{D(t,t')-\min(D)}{\max(D)-\min(D)} \tag{6}$$

$$PCS\colon T\times T\to[0,1]$$
$$PCS(t,t') = 1 - D_{\text{MinMax}}(t,t') \tag{7}$$

$$P_{\text{MinMax}}\colon T\times T\to[0,1]$$
$$P_{\text{MinMax}}(t) = \frac{P(t)-\min(P)}{\max(P)-\min(P)} \tag{8}$$

$$S\colon T \times T \to [0,1]$$
$$S(t,t') = \frac{P_{\text{MinMax}}(t) + \cos(t,t') + PCS(t,t')}{3} \tag{9}$$

Equation (9) deserves an explanation. Analyzing it, we observe that function S is not symmetric in t and t'. Apparently this could be problem, but actually it is not if we realize how $S(t,t')$ is used by our recommendation algorithm. t' acts as input track, based on which recommendations t are generated. Equation (9) actually says that the provided recommendations are those with a high popularity $P_{\text{MinMax}}(t)$ and that are similar with t' in lyrics $\cos(t,t')$ and acoustics $PCS(t,t')$.

4. System Evaluation

A recommendation system should provide recommendations of items that are rather similar to items seen in the past – known as exploitation capability, as well as of items different from those seen in the past – known as exploration capability. In this section, we initially provided our evaluation of the exploitation capability of our content-based recommendation system, considering the number of principal components, the number of clusters and the number of recommendations to evaluate the precision of the recommendations generated by the system across all test users available. Next, considering the average precision of the recommendations generated from each evaluation track, we determined the optimal number of recommendations. Contrary to our expectations, the results were not so different and promising regardless of the pair comprising the number of clusters and the number of principal components. As such, we came up with a new evaluation protocol of the experimental recommendation system, which was designed as follows:

1. **User-Track Interaction Simulation**

   We simulate interactions between a test user and tracks they've listened to multiple times.

2. **Recommendation Evaluation**

   We assess how many recommended tracks align with the user's listening history, considering tracks played $\geq$ 3 times as "liked." This evaluation uses a refined test dataset that focuses exclusively on users with 5-10 frequently played tracks, which they listened to at least 3 times.

In this section, we firstly describe the evaluation dataset and how it was prepared in 4.1, then the evaluation measures in 4.2 and finally the results derived from the evaluation protocol in 4.3.

4.1. User-Track Interaction Dataset Description & Preparation

In order to evaluate the quality of the track recommendation system, we used a dataset that contains user-track interactions. This dataset was created independently from the datasets of tracks and lyrics that were used for training the recommender system.

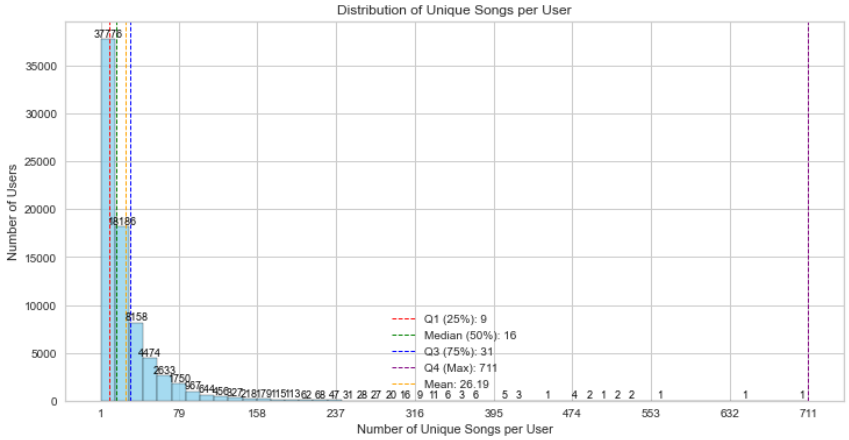


Fig. 2. Distribution of 76353 users across 9733 tracks

The dataset has 2 million entries represented by pairs (*user id*,*track id*) capturing interactions between users and tracks. The dataset contains 9733 unique tracks and 76353 unique users. Basically in our evaluation we simulate users interacting with tracks based on random samples extracted from this dataset.

In **Table 3**, we highlighted the distribution of values of each feature $f \in F$, where $F = \{$*danceability*, *energy*, *valence*, *acousticness*, *loudness*, *speechiness*, *liveness*, *instrumentalness*$\}$ across all songs which all test users listened to. Apart from **loudness**, which represents a value in dB, each feature belongs more or less to [0,1] range. As such, we can summarize the following conclusions:

- **Danceability**, **energy**, **valence** and **acousticness** are fairly evenly distributed. The mean of those features suggests that most songs exhibit a moderate tendency to be danceable, to have an overall neutral to slightly more positive emotional tone, moderate energy, and a mix of acoustic and electronic tracks.
- **Loudness** is highly variable and negative in value. The mean loudness indicates that the songs are generally quieter, since they have less than 0 dB sound levels. Furthermore, the difference between the loudest and the quietest song confirms the significant variation of sound levels.
- **Speechiness** is generally low, which means that most tracks have little ”spoken” content. The 75th percentile confirms the lack of spoken content in most songs, but considering the maximum value, there are, however, some songs that can be very speech-dominant.
- **Liveness** is generally low, but in some songs it indicates that there is probably a live performance in the audio recording.
- **Instrumentalness** has a very low mean, suggesting that most tracks have vocals. However, despite the low value, the maximum value indicates that there are some tracks that are entirely instrumental.

Taking into account Figure 2, we plotted the distribution of user-track interactions by the number of users according to the number of unique songs played per user. Then we

categorize the interaction based on the minimum number of unique songs played per user (Q0), the first quartile of unique songs played per user (Q1), the median value of unique songs played per user (Q2), the third quartile of unique songs played per user (Q3) and the maximum number of unique songs played per user (Q4), as follows:

I.) $1 \leq$ number of unique songs played per user < 9 : 41706 users.
II.) $9 \leq$ number of unique songs played per user < 16 : 10911 users.
III.) $16 \leq$ number of unique songs played per user < 31 : 6848 users.
IV.) $31 \leq$ number of unique songs played per user ≤ 711 : 2691 users.

In addition, each user listened to each song a certain number of times (has a listen count). On average, a user listens to a song roughly 3 times. In both the median and the first quartile, a user listens to a song a single time. In the third quartile, a user listens to a song 3 times and the maximum number of times a user listened to a song is 2213.

As such, we decided that if a user listened to a song at least 3 times, we consider it a meaningful or relevant user-track interaction. Therefore, we filter out all rows where a user listened to a song less than 3 times. After filtering those rows out, we get the distribution of 62156 users in 9731 songs from Figure 3 and the following range values, similarly categorized, but with different values, from the previous enumeration:

I.) $1 \leq$ number of unique songs played per user < 2 : 9055 users.
II.) $2 \leq$ number of unique songs played per user < 5 : 18620 users.
III.) $5 \leq$ number of unique songs played per user < 11 : 18315 users.
IV.) $11 \leq$ number of unique songs played per user ≤ 283 : 16166 users.

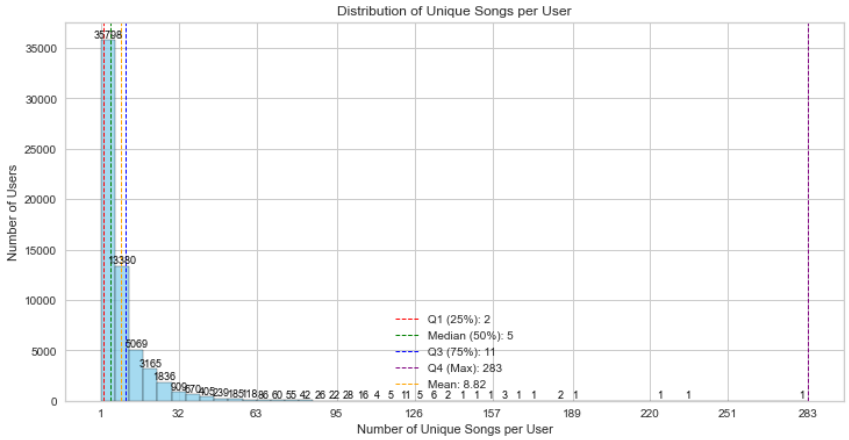


Fig. 3. Distribution of 62156 users across 9731 tracks after filtering

Table 3. Summary statistics for audio features

	Danceability	Energy	Valence	Loudness	Speechiness	Liveness	Acousticness	Instrumentalness
mean	0.533	0.48	0.517	-11.743	0.102	0.212	0.505	0.199
std	0.177	0.273	0.267	5.721	0.175	0.182	0.379	0.335
min	0	0.002	0	-42.66	0	0.017	0	0
25%	0.411	0.247	0.299	-14.993	0.035	0.1	0.098	0
50%	0.543	0.463	0.528	-10.831	0.045	0.138	0.529	0
75%	0.664	0.708	0.741	-7.452	0.076	0.271	0.901	0.262
max	0.982	1	0.984	0.781	0.968	0.996	0.996	1

4.2. Evaluation Protocol Description

For a fair evaluation protocol, we only considered the set of users U who listened to between 5 and 10 songs, and the set of unique songs T from user-track interactions, because that distribution contains a considerable number of unique users and tracks, as highlighted in Figure 3 and **Section 5.1**. For this sample of user-track interactions, we firstly created a profile for each user. The profile consists of the average values of each specific audio feature (*danceability*, *energy*, *valence*, *loudness*, *speechiness*, *liveness*, *acousticness* and *instrumentalness*) as highlighted in Definition (10), where:

- $P'(u)$ is the user profile of user $u \in U$
- $P'(u, f)$ is the user profile feature value of user u relative to an audio feature $f \in F$
- T_u is the set of tracks T played by a user $u \in U$,
- $f_{t,u}$ is the feature value of a track $t \in T$ which a user $u \in U$ listened to
- $|T_u|$ is the total number of tracks a user $u \in U$ listened to
- $|F|$ is the total number of features

$$\begin{aligned} P'(u) &= [P'(u, f)]_{|F|} \\ P'(u, f) &= \frac{\sum_{t_u \in T_u} f_{t,u}}{|T_u|} \end{aligned} \tag{10}$$

Once the user profile is determined, we randomly choose a track from the list of tracks the user listened to at least 3 times. Each track t_u has a probability $P(T_u = t_u) = \frac{listen_count(u,t_u)}{\sum_{t \in T_u} listen_count(u,t_u)}$ such that $P(T_u) = \sum_{t_u \in T_u} P(T_u = t_u) = 1$, where:

- $listen_count(u, t_u)$ is the number of times a user u listened to a track t_u
- $\sum_{t \in T_u} listen_count(u, t_u)$ is the total number of times a user u listened to all the tracks in T_u

After a track is randomly chosen, we use it as input in the recommender system to generate the top 5 most similar tracks. The recommendations are assessed as in Equations (11) and (12). Equation (11) measures on average how close specific feature values f_r of recommended tracks $R(t_u)$ are to the same features represented as the average values $P'(u, f)$ of the tracks T_u found in the user profile $P'(u)$, while Equation (12) computes the aggregated feature error $AFE(u)$ for each user $u \in U$ based on each individual feature from

Equation (11), which offers an overall assessment of similarity across all features $f \in F$ of all tracks $t_u \in T_u$ for each user $u \in U$.

$$MAE(t_u, f) = \frac{\sum_{r \in R(t_u)} |P(u, f) - f_r|}{|R(t_u)|} \tag{11}$$

$$AFE(u) = \frac{\sum_{f \in F} MAE(t_u, f)}{|F|} \tag{12}$$

4.3. Evaluation Results

In **Table 4**, we computed the mean absolute error $MAE(t_u, f)$ for each audio feature $f \in F$ between a randomly chosen track $t_u \in T_u$ played by each user $u \in U$ and all recommended tracks $R(t_u)$, and, the aggregated feature error $AFE(u)$ across all features $f \in F$ for each user $u \in U$, which are defined in Equations (11) and (12). Then, based on values from **Table 5**, we can notice that overall the recommendations provided to all test users are relevant based on their user profile, which consists of certain threshold values for each audio feature $f \in F$, because the difference in terms of audio features between the recommendations and the user preference pattern found in all audio features is barely noticeable, which is a good indicator of relevancy.

Table 4. Mean Absolute Error $MAE(t_u, f)$ of each feature $f \in F$ and Aggregated Feature Error $AFE(u)$ for each test user $u \in U$

User ID	Danceability	Energy	Valence	Loudness	Speechiness	Liveness	Acousticness	Instrumentalness	Aggregated Feature
0	0.066	0.119	0.319	1.559	0.01	0.055	0.569	0.001	0.337
1	0.082	0.424	0.159	6.765	0.023	0.01	0.794	0.03	0.941
2	0.094	0.318	0.125	7.048	0.022	0.123	0.166	0.266	0.885
3	0.28	0.035	0.156	0.817	0.611	0.138	0.457	0.001	0.312
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18314	0.212	0.239	0.642	3.549	0.014	0.05	0.653	0	0.670

Table 5. Summary statistics for across all $MAE(t_u, f)$ and $AFE(u)$ values

	Danceability	Energy	Valence	Loudness	Speechiness	Liveness	Acousticness	Instrumentalness	Aggregated Feature
mean	0.151	0.206	0.202	4.886	0.065	0.095	0.287	0.179	0.759
std	0.082	0.133	0.126	3.469	0.133	0.105	0.217	0.299	0.469
min	0.005	0	0.015	0.229	0	0	0.001	0	0.11
25%	0.092	0.102	0.108	2.568	0.013	0.036	0.111	0.001	0.451
50%	0.132	0.169	0.165	3.939	0.023	0.063	0.215	0.012	0.627
75%	0.191	0.282	0.27	6.047	0.050	0.113	0.424	0.2	0.906
max	0.605	0.830	0.848	30.478	0.921	0.879	0.983	0.971	4.154

5. Conclusions and Future Works

In this extended version of the paper, we achieved the following contributions:

- We focused on evaluating a smaller but more meaningful set of user-track interactions, unlike the previous one, which was more vast, but exhibited a significant imbalance in user listening behavior. Specifically, a large proportion of users have interacted with few songs, while a small subset of users accounts for the majority of listening activity. The smaller subset used for the evaluation was achieved by filtering out users who listened to less than 3 songs and then considering only users who listened to between 5 and 10 songs. In other words, the number of songs per user had a range of [Q1, Q2), where Q1 is the first quartile of unique songs a user listened to and Q2 is the median value of unique played songs. As such, we ended up with 128,834 unique user interactions consisting of 18,315 unique users and 9,733 unique songs.
- We created a user profile for each user in the set of test users, where we considered the aggregated value of each specific audio feature in all the songs found in the set of users, who played between 5 and 10 songs at least 3 times.
- We compared how suitable the recommendations are for each feature with each user profile using the mean absolute error measure.
- We compared how suitable the recommendations are across all features with each user profile by averaging the mean absolute error measure of each feature involved.

As future work, we can enhance the recommender system by incorporating a weighting mechanism to capture more user preferences (track genre, release year, album, etc.) for all tracks which have features that allow this. In terms of system evaluation, we can improve the evaluation protocol by using more appropriate information retrieval measures mentioned in [6] and [5].

- **Precision**: evaluates the proportion of relevant tracks in top-k recommendations. Although it was done in the previous version of this paper, we realized that the setup of the recommender system was not designed appropriately so that the precision of recommendations to each user can be properly and successfully evaluated by taking into account the available human ground truth. Hence, this is one reason why the recommendation system should be revised.
- **Average Precision (AP)**: evaluates the quality of classification within the top-k recommendations.
- **Normalized Distributed Cumulative Gain (NDCG)**: evaluates a given ordered classification with an "ideal" classifciation (can be thought of human ground truth) to a condensed list of recommended tracks.
- **Hit-Rate**: evaluates if there is a least one relevant in top-k recommended items for each user in the set of test users. However, one significant drawback of hit-rate is that a hit that occurs in the first position is treated equally with a hit that occurs in the last position of the list of recommendations.
- **Average Reciprocal Hit-Rank (ARHR)**: addresses the limitation of the hit-rate that rewards each hit equally regardless of its position by rewarding hits that occur earlier in top-k list with higher weights. This way, the highest value of ARHR is equal to the hit-rate and occurs when all the hits occur in the first position, whereas the lowest value of the ARHR is equal to the ratio between the hit-rate and the number of test

users when all the hits occur in the last position in the list of the top-k recommendations.

References

1. 6.2 feature extraction - sci-kit learn documentation. https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction
2. Alex Hsieh, Owen Luo, T.P.: Music attributes and its effect on popularity, https://ahsieh53632.github.io/music-attributes-and-popularity/
3. Arthur, D., Vassilvitskii, S., et al.: k-means++: The advantages of careful seeding. In: Soda. vol. 7, pp. 1027–1035 (2007)
4. Burneau, B.: New website: lyrics.ovh (January 2017), https://ntag.fr/lyrics-ovh-only-the-lyrics/
5. Castells, P., Moffat, A.: Offline recommender system evaluation: Challenges and new directions. AI Magazine 43 (06 2022)
6. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. 22(1), 143–177 (Jan 2004), https://doi.org/10.1145/963770.963776
7. for Developers, S.: Get several tracks' audio features. https://developer.spotify.com/documentation/web-api/reference/get-several-audio-features
8. Google Developers: K-Means Advantages and Disadvantages. https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages, last updated 2022-07-18 UTC
9. Khuat, V.: Music Recommendation Using Collaborative Filtering. https://portfolios.cs.earlham.edu/wp-content/uploads/2019/08/final_paper_draft_Vuong.pdf (2018), senior Capstone Experience, Earlham College
10. Lops, P., de Gemmis, M., Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends, pp. 73–105 (01 2011)
11. Mandloi, K., Mittal, A.: Hybrid music recommendation system using content-based filtering and k-mean clustering algorithm. International Journal of Computer Sciences and Engineering 6, 1498–1501 (07 2018)
12. Mishra, S.P., Sarkar, U., Sanjay, S.T., Devi, D., Swain, P., Saikhom, R., Panda, S., Laishram, M.: Multivariate statistical data analysis- principal component analysis (pca). International Journal of Livestock Research 7(5), 60–78 (2017)
13. Neremzoiu, I.D., Bădică, A., Ganea, E.: Track recommender system based on lyrics, audio and popularity features. In: 2024 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). pp. 1–6 (2024)
14. Nilashi, M., Bagherifard, K., Ibrahim, A.P.D.O., Alizadeh, H., Lasisi, A., Roozegar, N.: Collaborative filtering recommender systems. Research Journal of Applied Sciences, Engineering and Technology 5, 4168–4182 (04 2013)
15. Niyazov, A., Mikhailova, E., Egorova, O.: Content-based music recommendation system. vol. 29, pp. 274–279 (05 2021)
16. Rachwał, A., Popławska, E., Gorgol, I., Cieplak, T., Pliszczuk, D., Skowron, Ł., Rymarczyk, T.: Determining the quality of a dataset in clustering terms. Applied Sciences 13(5), 2942 (2023)
17. Ronak Doshi, Nithin Raj, A.R.S.: Song recommendation system (November 2017), https://github.com/ronak66/Song-Recommendation-System
18. Yuan, C., Yang, H.: Research on k-value selection method of k-means clustering algorithm. J 2(2), 226–235 (2019), https://www.mdpi.com/2571-8800/2/2/16
19. Zhang, X.: spotify-dataframe (2021), https://jovian.ai/zhangxm963/spotify-dataframe/v/39/files?filename=spotify-dataset-19212020-160k-tracks/data.csv
20. Çano, E.: Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis 21, 1487–1524 (11 2017)

Ionuț-Dragoș Neremzoiu is a first-year PhD student in Computer Science at the University of Craiova, Faculty of Automation, Computers and Electronics. He also works as a Systems Engineer at the University's Faculty of Law. His current research focuses on Explainable LLM-based Recommender Systems, which is also the topic of his doctoral thesis. His broader research interests include artificial intelligence, machine learning, recommender systems, and large language models.

Andreea Liliana Bădică holds a PhD in Management from the University of Craiova, where she conducted research between 2019 and 2022. She is currently employed as a Sales Operations Specialist at BMW Group in Salzburg, Austria, where she contributes to operational processes related to ordering across the European region. Her academic background and professional experience reflect a strong interest in the intersection of business operations, strategic management, and international markets.

Received: April 18, 2025; Accepted: September 01, 2025.

Computer Science and Information Systems 23(1):603–619 https://doi.org/10.2298/CSIS250410081B

Trustless Exchange for Personal Data: Adapting Intellectual Property Trading Security Protocols for Data Sovereignty*

Vijon Baraku[1], Simeon Veloudis[2,3], Iraklis Paraskakis[2,3], Poonam Yadav[1], and Rezon Baraku[3]

[1] Department of Computer Science, University of York, York, United Kingdom
vibaraku@seerc.org
poonam.yadav@york.ac.uk
[2] SEERC - South East European Research Centre, Thessaloniki, Greece
{iparaskakis, sveloudis}@seerc.org
[3] CITY College, University of York Europe Campus, Thessaloniki, Greece
rbaraku@york.citycollege.eu

Abstract. This paper presents an approach for adapting blockchain-based security protocols originally designed for intellectual property trading to support data sovereignty through secure exchange in personal data stores. As personal data becomes increasingly valuable in the digital economy, individuals need mechanisms to control and potentially monetise their information while ensuring its proper use. We examine how our existing multi-stage verification protocol can be applied to address the requirements of personal data exchange. The protocol uses symmetric and asymmetric encryption, smart contracts, and blockchain verification to create secure trading mechanisms. The proposed adaptation maintains the trustless verification and secure transfer mechanisms of the original protocol, creating a "meta-trust" layer, a higher level of confidence that extends beyond the foundational trust offered by blockchain technology alone. Through this conceptual adaptation, we explore a complementary framework that could enhance existing Personal Data Store (PDS) architectures. This framework enables individuals to selectively share their personal data with cryptographic guarantees about access and use, contributing to the advancement of data sovereignty in digital ecosystems.

Keywords: Data Sovereignty, Personal Data Stores, Blockchain technology, Smart contracts.

1. Introduction

In today's digital landscape, personal data has become a key enabler of economic and technological progress. Organisations collect and process vast amounts of personal information to drive decision-making, enhance services, and create new business opportunities, effectively transforming personal data into a commodity of extraordinary value [22, 11]. However, unlike other commodities, personal data is intrinsically linked to individuals,

* This is an extended version of the conference paper "Blockchain-Based Decentralised Marketplace for Secure Trading of Intellectual Property" presented at INISTA 2024 - The 18th International Conference on Innovations in Intelligent Systems and Applications/BCI 2024

raising significant ethical and legal concerns surrounding its use. In particular, the power asymmetry between data controllers, organisations that legally own and determine how personal data is processed, and individuals has led to growing concerns about privacy, transparency, and ethical data handling practices [19].

In response, the concept of *data sovereignty*, the principle that individuals should have ultimate control over their personal data, has emerged as a critical paradigm shift in data management [2, 10]. Data sovereignty encompasses both legal and technical dimensions. Legally, it is supported by regulatory frameworks such as the General Data Protection Regulation (GDPR), Data Governance Act (DGA), and the Data Act (DA), which aim to protect individuals' rights over their personal information. Technically, however, implementing data sovereignty requires innovative solutions that truly empower individuals within the digital ecosystem.

A promising approach to enabling data sovereignty involves Personal Data Stores (PDS) [6, 13, 15, 21, 9, 17], platforms that allow individuals to create and maintain private repositories of their personal data, and to define the terms under which this data can be accessed or shared. Moreover, many PDS incorporate mechanisms for data monetisation, enabling individuals to profit from personal data exchanges. However, for such digital asset trading to gain traction, a critical challenge must be addressed: establishing buyer trust in the quality and authenticity of the traded data, while preserving the individual's ultimate control over the data.

Contributions: This paper makes the following key contributions:

- We adapt a blockchain-based security protocol from intellectual property trading to personal data sovereignty contexts, addressing buyer-side trust challenges in existing PDS implementations.
- We present detailed algorithmic modifications that enable secure preview of personal data before purchase while maintaining individual control and data sovereignty.
- We provide a proof-of-concept implementation that validates the technical feasibility of the adapted protocol.

The paper explores how our previously developed blockchain-based security protocol for trading intellectual property [5] can be adapted specifically for personal data stores. Rather than proposing a new PDS architecture, our contribution lies in developing a protocol that enhances existing PDS operations by enabling the secure preview of personal data before purchase. The adapted protocol aims to establish a higher level of trust in personal data trading, a form of meta-trust that extends beyond the foundational trust offered by blockchain technology. It enables individuals to selectively share or trade their data with cryptographic assurances regarding access and use, thereby fostering broader participation in the data economy while maintaining data sovereignty.

To validate our approach, we have developed a proof-of-concept implementation that demonstrates the core mechanisms of the protocol adaptation. This technical demonstration confirms the feasibility of our approach while providing practical insights that complement our theoretical framework.

To illustrate the potential impact of this approach, consider Alice, who manages a chronic health condition and regularly generates valuable health data through monitoring devices. If Alice wanted to monetise this health data through a PDS, potential buyers (such as pharmaceutical researchers) currently have no way to verify the quality or relevance of

her data without accessing it first, creating a trust paradox. Our protocol addresses this dilemma by allowing researchers to preview a cryptographically secured samples of the data before purchase, while ensuring that Alice maintains control over what is shared and that the complete dataset remains protected.

The remainder of this paper is structured as follows. Section 2 provides background on blockchain technologies, data sovereignty, and personal data stores. Section 3 describes our blockchain-based security protocol for trading digital assets, with emphasis on the generic mechanisms that enable trustless transactions. Section 3.2 presents our main contribution: the adaptation of this protocol for personal data stores. Implementation considerations and use cases are discussed in Sections 3.3. Sections 3.5 and 4 conclude the paper with discussion and final remarks.

2. Background and Related Work

This section reviews relevant literature in three areas: blockchain technologies in digital asset trading, personal data sovereignty concepts, and existing personal data store implementations.

2.1. Blockchain Technologies for Digital Asset Trading

Blockchain technology has paved the way for secure digital asset exchange without centralised intermediaries. The foundational work of Nakamoto [16] introduced the blockchain as a distributed ledger for value transfer, while Buterin's Ethereum platform [8] expanded these capabilities through programmable smart contracts that enable complex trading logic and automated agreement enforcement.

Several researchers have explored blockchain applications for intellectual property and digital rights management, primarily focusing on ownership tracking and secure transfers. For instance, Savelyev [18] examined blockchain's role in managing intellectual property rights, analysing both technical and legal frameworks for tracking ownership and facilitating secure transfers. Ma et al. [14] presented a blockchain-based digital rights management system with cryptographic verification mechanisms for secure content transfer. Bodó et al. [7] proposed a blockchain-based licensing system for digital content that leverages smart contracts for automated enforcement.

While these works demonstrate blockchain's potential for managing digital rights and facilitating asset transfers, they overlook a fundamental trust paradox in personal data trading: buyers must be able to verify the quality of the data before purchase, while sellers must keep the traded data confidential until the transaction is complete.

2.2. Personal Data Sovereignty

Data sovereignty encompasses both legal considerations and individual control. Legally, data sovereignty refers to the principle that data is subject to the laws and governance structures of the country in which it is gathered or held [10]. This includes regulatory frameworks such as the EU's GDPR, the DGA, and the DA, which establish provisions for data protection, usage, and rights. This interpretation has gained relevance and prominence with the rise of cloud computing and global data flows.

Alongside this legal view, a more granular and individual-centric interpretation of data sovereignty has emerged, aided by the efforts of organisations such as the International Data Spaces Association (IDSA) and MyData.org. This interpretation emphasises individual agency, the ability to control how personal data is used, even across technological boundaries. This viewpoint defines data sovereignty as the ability of individuals to retain complete control over their data assets by determining how these may be used across technological boundaries [2]. This perspective shifts focus from legal frameworks to personal agency, emphasising that data sovereignty also involves technical methods that enable data subjects to:

- Specify and enforce usage conditions for personal data assets, regardless of storage location.
- Grant or revoke access rights dynamically, as needed.

However, despite these advancements, practical implementation remains a challenge, as individuals currently lack effective technical means to oversee how their personal information is collected, used, and shared across the digital ecosystem [12].

2.3. Personal Data Stores

PDS represent a paradigm shift in personal data management, placing individuals, rather than organisations, at the center of data control. Unlike traditional models where organisations retain custody of user data in siloed systems and dictate its use, PDS enable individuals to collect, store, manage, and selectively share their personal information from a repository under their direct control. This approach allows for sovereign personal data trading, where individuals determine how their data is accessed and exchanged.

Despite their advantages, PDS do not eliminate organisational data collection: businesses will continue to track user interactions, maintain customer records, and store legally required data. However, PDS introduce greater transparency and control, allowing individuals to negotiate access to their data, reduce excessive tracking, and align data-sharing with their privacy preferences. They also potentially reduce the amount of personal data traded directly between organisations. While challenges remain, PDS have the potential to reshape power dynamics in the digital economy by shifting the balance of control towards individuals.

The conceptual foundations of PDS emerged from early work on user-centric identity and data management, evolving through several generations of implementations to address increasing complexity in personal data governance. Early implementations focused primarily on secure storage and basic permission management. As the need for greater control and interoperability grew, later implementations incorporated more sophisticated features including granular access control, revocation mechanisms, data portability tools, and integration with emerging standards for consent and authentication [4, 3, 1].

Modern PDS architectures integrate several core technical components (see Figure 1) to address security, usability, and interoperability challenges. At their core, PDS feature a secure data repository capable of handling diverse data types while maintaining integrity and supporting efficient queries. This repository can be centralised (cloud-based), or distributed (content-addressable storage), each with implications for control and accessibility. Beyond storage, PDS incorporate identity and authentication mechanisms to

verify both the individual's identity and those of potential data recipients. Access control systems enforce permissions at various levels of granularity, from entire datasets to specific attributes. Many implementations also include transparency logging to record all access events and data operations, consent management interfaces for capturing explicit data-sharing authorisations, and interoperability features that facilitate cross-system data exchange [20].

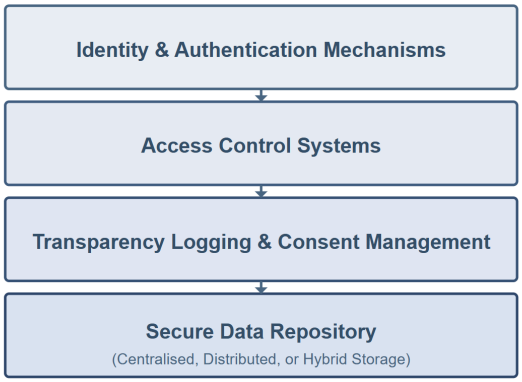


Fig. 1. Personal data store architecture with core components: data repository, identity management, access control, logging, consent interfaces, and interoperability features

Several implementations of PDS have been proposed, each emphasising different priorities including decentralisation, privacy, governance, or data economics. Below, we review key PDS platforms that illustrate these diverse approaches.

Solid (Social Linked Data), developed by Berners-Lee [6], prioritises decentralisation and interoperability through its pod-based architecture. Each pod functions as a personal web server hosting data in RDF format, with access managed via Web Access Control (WAC) and Access Control Policies (ACP). WAC enforces basic access permissions at the resource level, while ACP enables more complex, policy-driven controls. Solid offers both self-hosted deployment for technical users and managed hosting through providers like inrupt.com, balancing control with usability. The platform emphasises interoperability leveraging standard web protocols and linked data principles to facilitate seamless data portability across different Solid servers. While Solid has gained significant attention in academic and privacy advocacy communities, its real-world adoption remains limited primarily to research projects and early adopters. The platform currently lacks built-in monetisation mechanisms, focusing instead on data ownership and control rather than economic exchange.

Mydex [13] established one of the earliest operational PDS platforms, implementing a centralised "personal data account" model with strong governance provisions. As a Community Interest Company, Mydex operates under legally binding commitments that prevent exploitation of user data for corporate interests. Regarding deployment, Mydex uses a cloud-based infrastructure model where the organisation maintains the technical infrastructure while individuals maintain control of their data. The platform implements at-

tribute exchange protocols that enable selective disclosure of validated information rather than sharing entire records. Their security architecture includes both symmetric and asymmetric encryption of data at rest and in transit, with keys controlled by the individual, and a verification framework that can authenticate the source and integrity of stored information. For interoperability, Mydex supports standardised data schemas and exchange protocols, though with less emphasis on open standards than Solid. The platform includes monetisation provisions through data exchange agreements where individuals can receive compensation for sharing validated attributes. In terms of adoption, Mydex has achieved limited but concrete implementation in several UK public sector projects.

Databox [15, 21, 9] takes a privacy-centric approach by implementing a localised deployment model based on edge computing. Databox processes data on users' own hardware, such as home routers, minimising external data movement and reducing exposure to third-party systems. Instead of focusing on secure sharing, Databox brings computation to the data through containerised "drivers" that perform specific processing tasks within the secure environment. This architecture implements the principle of data minimisation by enabling third parties to derive insights without accessing raw data, creating a strong privacy protection mechanism. The system includes a "store-first" model where data is captured from various sources before any processing occurs, an app platform for running containerised processing code, and an audit system that documents all data uses. Interoperability in Databox focuses on API-based integration with data sources and processing applications rather than direct data exchange with other systems. The platform includes potential for monetisation through an app marketplace model where users can selectively allow processing of their data in exchange for services or compensation. While technically sophisticated, Databox remains primarily in the academic research domain with limited commercial adoption.

The Hub-of-All-Things (HAT) platform [17] differentiates itself with a strong emphasis on data economics and value exchange. HAT implements "Microservers" as personal data repositories owned and controlled by individuals, with built-in capabilities for data transformation, combination, and analysis. Unlike Solid pods which focus primarily on storage and access control, HAT Microservers include built-in analytics capabilities to derive value from personal data. Regarding deployment, HAT offers a hybrid approach with distributed ownership but centralised technology provision through the HAT Community Foundation. The architecture includes "Data Debits" as structured exchange mechanisms that formalise the terms under which data can be shared, creating explicit value exchange relationships between individuals and data consumers. While similar in concept to smart contracts, Data Debits are specifically designed for personal data exchange with predefined duration and purpose limitations, compared to the more general computational capabilities of blockchain-based smart contracts. HAT's security model combines encryption, access control, and a market-based approach to data valuation. Interoperability is addressed through standardised APIs and data schemas. The platform has achieved moderate adoption through partnerships with commercial entities in the UK and Singapore, with working implementations in the retail and transportation sectors.

This analysis highlights the trade-offs between decentralisation, security, interoperability, and economic models in PDS implementations. While Solid emphasises decentralisation and web standards compatibility, Mydex focuses on governance and validated attribute exchange. Databox prioritises privacy-preservation through local computation,

Table 1. Comparison of Personal Data Store Implementations

PDS	Deployment	Security/Privacy	Interoperability	Monetisation	Adoption
Solid	Self-hosted or managed services; decentralised	WAC/ACP access control; encryption in transit	High: Web standards, RDF, linked data	None built-in	Limited: Research projects
Mydex	Cloud-based; centralised with individual control	Encryption at rest/transit; selective disclosure	Medium: Standard schemas and protocols	Attribute exchange agreements	Limited: UK public sector
Databox	Local hardware; edge computing; highly decentralised	Data minimisation; containerised processing; local computation	Medium: API integration with data sources	App marketplace model	Very limited: Academic only
HAT	Hybrid: distributed ownership, centralised technology	Encryption; Data Debits with time/purpose limits	Medium: Standard APIs and schemas	Data Debits as economic exchange	Moderate: Commercial partnerships

and HAT explores data monetisation possibilities with its Data Debits model. Each implementation represents different priorities within the personal data sovereignty space, with varying degrees of success in real-world adoption. However, none of these approaches make provisions to enhance buyer-side trust through mechanisms that would allow data consumers to verify data quality or authenticity before completing transactions.

These implementations demonstrate diverse approaches to personal data management, yet none fully leverage blockchain technology. Moreover, current PDS architectures lack robust mechanisms for buyers to verify data quality before committing to an exchange, creating significant challenges in personal data markets where trust is essential for enabling scalable data transactions between strangers. This trust deficit represents a key distinction between existing approaches and our blockchain-based framework. While current systems rely primarily on conventional access control mechanisms, our blockchain-based approach offers several unique advantages: transparent and immutable transaction records that create verifiable evidence of data exchanges and trustless verification through cryptographic techniques. These capabilities are particularly valuable in scenarios where individuals wish to exchange data with previously unknown parties or monetise their personal information while maintaining cryptographic guarantees about access. By exploring how these blockchain-based mechanisms could be applied to personal data contexts, our work highlights a complementary technical approach that could enhance the security and trust characteristics of future personal data exchange systems.

3. Blockchain-Based Protocol for Trading Intellectual Property and its Adaptation for Personal Data Stores

This section presents the blockchain-based security protocol for trading Intellectual Property originally introduced in [5]. The protocol enables buyers to preview digital assets before purchasing, thus establishing a higher level of trust, or meta-trust, that goes beyond

the trust already provided by blockchains. This meta-trust is crucial in contexts where buyers require quality verification before committing to a purchase, which applies both to intellectual property trading [5] and personal data exchanges facilitated through PDS. Current PDS implementations lack robust mechanisms for such verification, making this protocol adaptation particularly valuable for advancing personal data sovereignty.

While describing each component of the protocol, we simultaneously discuss how it can be adapted to address the specific requirements of PDS and enable personal data sovereignty. By examining both the original application and potential adaptations, we demonstrate how specialised security mechanisms developed for intellectual property trading can be repurposed to enhance individual control over personal data.

3.1. Prerequisites and System Architecture

The protocol relies on several key components to facilitate secure trading without intermediaries. It is implemented as a web-based decentralised application (dApp), but differs from traditional web applications that separate frontend and backend functionalities. Instead, our dApp connects the frontend directly to blockchain networks via wallets, which store blockchain addresses and cryptographic keys to authenticate users and execute smart contract transactions.

Once users link their blockchain wallets, they can submit requests through the frontend, which are processed by smart contracts on the blockchain, stored in the InterPlanetary File System (IPFS)[4], or handled directly within the frontend. IPFS serves as a decentralised storage system for file retrieval, while smart contracts enforce business logic on-chain. To enhance security, critical operations such as file encryption and decryption occur locally on the frontend, minimising unnecessary data exposure.

The system supports two primary user roles:

- **Sellers**: Individuals who provide digital assets for sale, along with necessary metadata such as price and encrypted content location.
- **Buyers**: Users who wish to purchase digital assets, with mechanisms to verify quality through randomly selected samples before committing to purchase.

In the context of PDS, these roles would translate to:

- **Data Subjects**: Individuals who offer their personal data for sharing or monetisation through the PDS.
- **Data Recipients**: Entities interested in accessing or purchasing personal data, who need verification of data quality and relevance before commitment.

This direct correlation between roles in both contexts demonstrates the initial feasibility of adapting the protocol. However, personal data exchange presents specific requirements that necessitate adaptations to the original protocol, as discussed in the following subsections.

3.2. Protocol Design and Adaptation

The security protocol consists of four stages, each with specific adaptation considerations for personal data contexts.

[4] A peer-to-peer network for sharing data using a distributed hash table. `https://ipfs.tech/`

Stage 1: Preparing Digital Assets for Sale. In the initial stage, sellers prepare their digital assets for secure trading. The seller must ensure that only authorised users (buyers who have completed the purchase) can view the content, protecting the value of the digital assets until a transaction is complete. To achieve this, the seller proceeds as follows:

Algorithm 1 Asset Preparation Protocol

```
procedure PREPAREASSETS(assets[])
    for each asset in assets[] do
        symmetricKey ← GenerateRandomKey()
        encryptedAsset ← EncryptAES256(asset, symmetricKey)
        Store symmetricKey securely
        Store encryptedAsset on IPFS
    end for
end procedure
```

As outlined in Algorithm 1 (Asset Preparation Protocol), the seller encrypts each digital asset using symmetric encryption (AES-256) and generates unique encryption keys. This encryption ensures that even if the assets themselves are accessible through IPFS, they remain confidential without the corresponding decryption keys.

Adaptation for PDS: For personal data contexts, the algorithm implementation organises data into structured records based on distinct categories, including demographic information, transaction history, and preference data, rather than treating each asset as a unique creative work. The PDS implementation maintains the same encryption approach while extending it with an authenticity metadata layer.

Unlike intellectual property, where value is largely subjective, personal data often requires verification of source and accuracy. The protocol extension incorporates authenticity indicators such as digital signatures from original data sources and verification timestamps into the metadata structure. This authentication mechanism directly addresses the verification requirements in personal data exchanges while preserving the security foundation of the original protocol.

Stage 2: Sample Request and Selection. To ensure buyers can verify product quality without gaining access to the complete collection, the protocol implements a secure sampling mechanism:

Algorithm 2 (Sample Request Protocol) implements a verifiable random selection process that is crucial for fair sample verification. This process prevents sellers from deliberately choosing high-quality samples that might not represent the overall collection quality. The smart contract randomly selects one encrypted key hash from all submitted hashes, and the blockchain's transparency ensures this selection process cannot be manipulated by either party.

Adaptation for PDS: For personal data exchange, the algorithm implementation provides meaningful sampling capabilities while preserving the critical randomness mechanism that prevents selective presentation. Personal data requires contextual understanding for effective evaluation, a feature missing from current PDS architectures that creates significant friction in personal data markets.

Algorithm 2 Sample Request Protocol

procedure REQUESTSAMPLE($symmetricKeys[]$)
  Buyer → Seller: Sample request with $buyerPublicKey$
  $encryptedKeys[] \leftarrow \emptyset$
  for each key in $symmetricKeys[]$ **do**
    $encryptedKey \leftarrow$ AsymmetricEncrypt(key, $buyerPublicKey$)
    $hashedKey \leftarrow$ Hash($encryptedKey$)
    Add $hashedKey$ to $encryptedKeys[]$
  end for
  Seller → Blockchain: Upload $encryptedKeys[]$
  $randomIndex \leftarrow$ SmartContract.SelectRandom(0, length($encryptedKeys[]$)-1)
  $selectedHash \leftarrow encryptedKeys[randomIndex]$
  return $selectedHash$
end procedure

The protocol implementation enhances the sample selection process with contextual metadata that characterises the selected data without compromising confidentiality. This contextual layer enables data recipients to properly interpret and evaluate the sample data within its broader framework. The implementation also supports multiple sample selection across different data categories, delivering a comprehensive cross-sectional view of data quality across various personal information types.

Stage 3: Sample Delivery. Once a sample has been selected, the seller must provide the corresponding key to the buyer:

Algorithm 3 Sample Delivery Protocol

procedure DELIVERSAMPLE($selectedHash$)
  SmartContract → Seller: Notification of selected hash
  Seller → SmartContract: $encryptedKey$ corresponding to $selectedHash$
  SmartContract: Verify $hash(encryptedKey) = selectedHash$
  if verification succeeds **then**
    SmartContract → Buyer: $encryptedKey$
    Buyer: $symmetricKey \leftarrow decrypt(encryptedKey, buyerPrivateKey)$
    Buyer: Access sample using $symmetricKey$
  else
    Transaction fails
  end if
end procedure

Algorithm 3 (Sample Delivery Protocol) manages the secure delivery of the selected sample to the potential buyer. Once a hash is randomly selected, the protocol verifies the corresponding encrypted key through hash comparison before delivery, ensuring that only the specific sample chosen by the smart contract is provided to the buyer.

Adaptation for PDS: The sample delivery mechanism translates directly to personal data contexts with minimal architectural changes. The core verification process maintains

the same cryptographic integrity while integrating the contextual metadata layer that positions each sample within the broader data collection framework.

This implementation preserves the security architecture of the original protocol while specifically addressing the challenges of evaluating personal data quality. The transparent verification process builds recipient confidence that they are examining representative samples of the actual data they may acquire, addressing a critical trust gap in current personal data exchange systems.

Stage 4: Purchase and Complete Delivery. If the buyer is satisfied with the sample quality, they can proceed with the purchase:

Algorithm 4 Purchase Completion Protocol

procedure COMPLETEPURCHASE($hashedKeys[]$)
  Buyer → SmartContract: Purchase request with payment
  SmartContract: Hold payment in escrow
  SmartContract → Seller: Purchase notification
  Seller → SmartContract: All $encryptedKeys[]$
  $verified \leftarrow$ TRUE
  for each $encryptedKey$ in $encryptedKeys[]$ **do**
    $hashedKey \leftarrow$ Hash($encryptedKey$)
    if $hashedKey$ not in $hashedKeys[]$ **then**
      $verified \leftarrow$ FALSE
      break
    end if
  end for
  if $verified$ **then**
    SmartContract → Seller: Release payment
    SmartContract → Buyer: All $encryptedKeys[]$
    Buyer: Decrypt and access all assets
  else
    Transaction fails
  end if
end procedure

Algorithm 4 (Purchase Completion Protocol) finalises the transaction while ensuring both parties fulfil their obligations. The smart contract holds payment in escrow until the seller provides all remaining encrypted keys, which are then verified against their previously stored hashes. This verification ensures the buyer receives all necessary keys to access their purchased content while the seller receives payment only after providing valid keys.

Adaptation for PDS: For personal data exchange, this stage implements expanded transaction terms while maintaining the core key delivery mechanisms. The smart contract extension incorporates detailed usage agreement parameters including permitted purposes, access duration rights, processing restrictions, and attribution requirements.

The implementation stores hashed representations of these agreement terms and requires explicit cryptographic acceptance via digital signatures. This architecture produces

an immutable record that the recipient acknowledged specific conditions prior to receiving access. While the smart contract cannot directly monitor post-delivery data usage, the blockchain record serves as definitive evidence of the agreed-upon terms for dispute resolution.

This implementation bridges a critical gap in current personal data exchange systems that lack robust mechanisms for formalising and recording usage agreements. The protocol extension gives individuals precise control mechanisms over how their personal data is used after sharing.

Public Key Recovery. Supporting all stages of the protocol, a critical technical component is the ability to recover a user's public key from their blockchain address:

Algorithm 5 Public Key Recovery Protocol

1: **procedure** RECOVERPUBLICKEY($ethereumAddress$)
2: $addressHash \leftarrow$ Keccak256($ethereumAddress$)
3: $signature \leftarrow$ Sign($addressHash$, $userPrivateKey$)
4: Extract r, s values from $signature$
5: $publicKey \leftarrow$ ECDSARecover($addressHash$, r, s)
6: **return** $publicKey$
7: **end procedure**

Algorithm 5 (Public Key Recovery Protocol) enables secure asymmetric encryption by recovering a user's public key from their Ethereum address. This recovery process allows for secure key exchange without requiring users to manually manage cryptographic keys.

Adaptation for PDS: This component functions identically in personal data contexts, as the cryptographic requirements for secure key exchange remain consistent across domains. The public key recovery mechanism operates in the PDS implementation with the same security architecture, maintaining consistent user experience and cryptographic protections.

3.3. Technical Implementation

These algorithms have been implemented in Solidity, the primary programming language for Ethereum smart contracts. The smart contract implementation oversees the entire transaction process, ensuring security without requiring trust between participants. The contract handles four critical functions: recording sample requests from potential buyers, implementing verifiable random sample selection, verifying the authenticity of provided keys via hash comparison, and securing the exchange of payment and keys.

In the Solidity implementation, we've incorporated important safeguards for both parties. For buyers, the contract includes an escrow mechanism that holds payments until valid keys are delivered and verified through hash comparison. If the seller fails to provide valid keys within a specified timeframe (24 hours in our implementation), the buyer can execute a function to retrieve their deposited funds. For sellers, the contract includes ownership verification that prevents unauthorised modifications and ensures only legitimate

buyers can access the encrypted keys. Additionally, the contract implements gas-efficient verification procedures that balance thorough validation with reasonable transaction costs.

This design creates a self-enforcing agreement that protects both intellectual property rights and buyer interests without the need for centralised intermediaries. Readers should refer to our previous work [5] for full implementation details.

Adaptation for PDS: The PDS implementation extends the smart contract with additional state variables and functions that handle the complex data usage agreements. The contract extension includes structured data types for agreement terms storage, cryptographic functions for verifying recipient acceptance signatures, and time-bound access control mechanisms that enforce specified duration limits.

The extended implementation maintains the core security architecture while incorporating the specific functionalities required for personal data contexts. The contract optimises these extensions for gas efficiency to ensure practical usability on public blockchains without compromising the enhanced functionality.

3.4. Security Analysis and Benefits

The security protocol provides several important guarantees that apply to both intellectual property trading and personal data exchange:

- **Seller/Data Subject Protection:** Complete confidentiality of digital assets or personal data is maintained, as all content is encrypted before being made available. Only authorised entities who have completed the transaction receive decryption keys.
- **Buyer/Recipient Protection:** The ability to verify quality through randomly selected samples before purchase prevents misrepresentation of assets or data. The blockchain enforced randomness ensures this verification is trustworthy.
- **Transaction Integrity:** The smart contract ensures that both parties fulfil their obligations, with buyers/recipients receiving valid keys only after payment and sellers/data subjects receiving payment only after providing valid keys.
- **No Trust Requirements:** The protocol eliminates the need for trust between parties. The cryptographic verification and blockchain-based enforcement ensure compliance without requiring trust in either the counterparty or a central authority.
- **Non-repudiation:** All transactions are recorded on the blockchain, providing an immutable record that neither party can later deny.

Additional Benefits for PDS: Adapting this protocol for PDS creates several important capabilities for personal data sovereignty that aren't present in existing PDS implementations:

- **Reduced Information Asymmetry:** Both parties can verify aspects of the exchange before commitment, addressing a critical weakness in current data marketplaces.
- **Enhanced Transparency:** The blockchain-based record creates transparency and accountability not typically found in conventional data sharing mechanisms.
- **Expanded Exchange Opportunities:** By minimising trust requirements through cryptographic verification and blockchain-based enforcement, the protocol potentially enables secure personal data exchange between individuals and organisations without established relationships.

– **Individual Empowerment:** The protocol provides individuals with greater agency in deciding how their personal data is shared and used, advancing the broader goal of data sovereignty.

The combination of symmetric encryption for content protection, asymmetric encryption for secure key exchange, and blockchain-based verification creates a comprehensive security framework that addresses the key challenges in both intellectual property trading and personal data exchange.

3.5. Key Requirements Addressed by Protocol Adaptation

To summarise, the protocol adaptation described in this section effectively addresses several critical requirements for secure personal data exchange:

– **Data Quality Verification:** By implementing the sample selection and delivery mechanisms (Algorithms 2 and 3), the protocol enables recipients to evaluate personal data before commitment, solving a key limitation of current PDS architectures.
– **Formal Usage Agreements:** The extended smart contract functionality creates cryptographically signed records of data usage terms, providing a foundation for accountability in personal data exchanges.
– **Secure Exchange Mechanism:** The trustless escrow system ensures fair exchange without requiring third-party intermediaries, protecting both data subjects and recipients.
– **Transparency and Non-repudiation:** The blockchain-based implementation provides an immutable record of all transactions, creating transparency that's often lacking in conventional data sharing approaches.

The theoretical protocol adaptation described in this paper has been complemented by a technical demonstration implementing key components of the system. This implementation successfully validates the sample selection and verification mechanisms in a personal data context, confirming that the cryptographic and blockchain-based verification approaches translate effectively from intellectual property to personal data sovereignty applications. The implementation particularly demonstrates the technical viability of trustless sample verification.

4. Conclusion and future works

This paper demonstrates how the blockchain-based security protocol developed for intellectual property trading could be utilised to address challenges in personal data sovereignty. The exploration of adapting a multi-stage verification protocol built on symmetric and asymmetric encryption, smart contracts, and blockchain verification reveals a practical framework for personal data exchange that preserves individual control while enabling participation in the data economy. The protocol creates a level of meta-trust that goes beyond the basic trust provided by blockchain technology alone, by allowing data buyers to verify quality before purchase while still protecting sellers' data sovereignty.

Rather than proposing a new PDS architecture, our research offers a complementary security protocol that addresses a critical gap in existing PDS implementations: the lack of

buyer-side trust mechanisms. The proposed adaptation maintains the core security mechanisms of the original protocol while introducing specific modifications to address the unique requirements of personal data contexts. By implementing structured data organisation, enhanced authenticity verification, and expanded usage agreements, the protocol creates a trustless environment where individuals can confidently share their personal information.

This research, including our technical demonstration, provides a foundation for further exploration. The implementation has validated our theoretical adaptations, particularly the critical sample selection and verification mechanisms. Building on this foundation, future work can focus on extending the implementation to more complex usage scenarios and evaluating its performance at scale.

Another promising direction for future work is exploring how smart contracts could be extended to enforce post-sharing obligations in personal data exchanges. While current PDS implementations typically focus on access control before sharing rather than enforcement after access is granted, our blockchain-based approach creates a foundation for implementing enforceable usage terms. Future research could investigate mechanisms for monitoring compliance with data usage agreements, implementing automated penalties for violations, and creating auditable records of data usage that extend beyond the initial exchange.

Additionally, future work should look at the legal and regulatory implications of the proposed approach, specifically how the protocol complies with GDPR standards and other data protection frameworks. This analysis would help identify any potential compliance gaps and determine how the protocol should evolve to fulfil regulatory standards.

The practical implementation faces certain technical challenges, particularly regarding smart contract extensions for complex usage agreements and robust authenticity verification. Nevertheless, the conceptual framework demonstrates significant promise in repurposing specialised security mechanisms to shift power dynamics in the digital data economy. This proposed adaptation represents a meaningful step toward enabling individuals to participate in and benefit from the value of their personal data while maintaining complete sovereignty over its access and use.

References

1. Abiteboul, S., André, B., Kaplan, D.: Managing your digital life. Communications of the ACM 58(5), 32–35 (2015)
2. Bader, S., Pullmann, J., Mader, C., Tramp, S., Quix, C., Müller, A.W., Akyürek, H., Böckmann, M., Imbusch, B.T., Lipp, J., et al.: The international data spaces information model–an ontology for sovereign exchange of digital content. In: International Semantic Web Conference. pp. 176–192. Springer (2020)
3. Baraku, V., Paraskakis, I., Veloudis, S., Yadav, P.: Personal data sovereignty in virtual enterprises: Implementing data capsules for enhanced privacy and compliance. In: Working Conference on Virtual Enterprises. pp. 447–461. Springer (2024)
4. Baraku, V., Paraskakis, I., Veloudis, S., Yadav, P.: Responsible information sharing in the era of big data analytics facilitating digital economy through the use of blockchain technology and observing gdpr. In: CLOSER. pp. 257–264 (2024)
5. Baraku, V., Veloudis, S., Paraskakis, I., Yadav, P.: Blockchain-based decentralised marketplace for secure trading of intellectual property. In: Bădică, C., Gušev, M., Iftene, A., Ivanović, M.,

Manolopoulos, Y., Xinogalos, S. (eds.) Advances in ICT Research in the Balkans, pp. 208–219. Springer Nature Switzerland, Cham (2025)

6. Berners-Lee, T., Sambra, A., Capadisli, S., Verborgh, R., Mansour, E., Hawke, S., Hess, C., Eling, N.: Solid: A platform for decentralized social applications based on linked data. Technical report, MIT CSAIL & Qatar Computing Research Institute (2018)
7. Bodó, B., Gervais, D., Quintais, J.: Blockchain and smart contracts: the missing link in copyright licensing? International Journal of Law and Information Technology 26(4), 311–336 (2018)
8. Buterin, V.: Ethereum white paper: A next-generation smart contract and decentralized application platform. Ethereum Foundation (2014), [Online]. Available: https://ethereum.org/en/whitepaper/
9. Crabtree, A., Lodge, T., Colley, J., et al.: Building accountability into the internet of things: the iot databox model. Journal of Reliable Intelligent Environments 4, 39–55 (2018)
10. De Hert, P., Papakonstantinou, V.: The new general data protection regulation: Still a sound system for the protection of individuals? Computer Law & Security Review 32(2), 179–194 (2016)
11. Fathullah, M.A., Subbarao, A., Muthaiyah, S.: A review of data breach cost in cloud computing. In: Proceedings of the International Conference on Technology and Innovation Management (ICTIM 2022). pp. 199–209. Atlantis Press (2022)
12. Janssen, M., Brous, P., Estevez, E., Barbosa, L., Janowski, T.: Data governance: Organizing data for trustworthy artificial intelligence. Government Information Quarterly 37(3), 101493 (2020)
13. Kirkham, T., Winfield, S., Ravet, S., Kellomäki, S.: The personal data store approach to personal data security. IEEE Security & Privacy 11(5), 12–19 (2013)
14. Ma, Z., Jiang, M., Gao, H., Wang, Z.: Blockchain for digital rights management. Future Generation Computer Systems 89, 746–764 (2018)
15. Mortier, R., Zhao, J., Crowcroft, J., Wang, L., Li, Q., Haddadi, H., Amar, Y., Crabtree, A., Colley, J., Lodge, T., Brown, A.: Personal data management with the databox: What's inside the box? In: Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking. pp. 49–54 (2016)
16. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review (2008)
17. Ng, I., Holtby, J., Ma, X., Alahmadi, A., Turoczy, S.: White paper on the hat (hub-of-all-things) and hatdex personal data exchange ecosystem. Working paper series, WMG Service Systems Research Group (2017)
18. Savelyev, A.: Copyright in the blockchain era: Promises and challenges. Computer Law & Security Review 34(3), 550–561 (2018)
19. Tawalbeh, L., Muheidat, F., Tawalbeh, M., Quwaider, M.: Iot privacy and security: Challenges and solutions. Applied Sciences 10(12), 4102 (2020)
20. Van Kleek, M., Liccardi, I., Binns, R., Zhao, J., Weitzner, D., Shadbolt, N.: Better the devil you know: Exposing the data sharing practices of smartphone apps. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. pp. 5208–5220 (2018)
21. Yadav, P., Moore, J., Li, Q., Mortier, R., Brown, A., Crabtree, A., Greenhalgh, C., McAuley, D., Amar, Y., Shamsabadi, A.S., Haddadi, H.: Providing occupancy as a service with databox. In: Proceedings of the 1st ACM International Workshop on Smart Cities and Fog Computing (CitiFog'18). pp. 29–34. Association for Computing Machinery, New York, NY, USA (2018)
22. Zuboff, S., Schwandt, K.: The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power. Profile Books (2019)

Vijon Baraku is a PhD student in the Department of Computer Science at the University of York, UK. His research interests focus on data sovereignty, data ownership, blockchain technology, and knowledge representation.

Simeon Veloudis is an Associate Professor in the Department of Computer Science at CITY College and a Senior Researcher at the South East European Research Centre (SEERC). He holds a PhD in Computer Science from the University of Reading. His research interests include Cloud Computing, Semantic Modelling, Knowledge Representation, Security, and Formal Methods.

Iraklis Paraskakis is a Professor of Information Systems in the Department of Computer Science at CITY College, Greece, and a Senior Research Officer at the South East European Research Centre (SEERC), where he coordinates the Information & Knowledge Management Research Group. He holds a PhD in Information Technology and Education from the Open University (UK) and an MSc from London School of Economics. His research focuses on Cloud Computing, Service Oriented Computing, Educational Informatics, and Knowledge Management.

Poonam Yadav is a Senior Lecturer in the Computer Science Department at the University of York, UK, and a visiting research fellow at Computer Lab, Cambridge University. Her research focuses on making IoT and edge computing-based distributed systems resilient, reliable, and robust, with particular emphasis on coordination and collaboration in resource-constrained environments. She leads the System and Network Interoperability (SYSTRON) Lab.

Rezon Baraku holds a BSc from CITY College, University of York. His research focuses on blockchain technology.

Received: April 10, 2025; Accepted: July 17, 2025.

Computer Science and Information Systems 23(1):621–655 https://doi.org/10.2298/CSIS241126032S

IoT based Personalized Online Learning Systems (Systematic Mapping Study)*

Edlir Spaho[1] and Betim Çiço[2]

[1] Epoka University
Tirana, Vore 1032, Albania
espaho@epoka,edu.al
[2] Faculty of Architecture and Engineering
Tirana, Vore 1032, Albania
bcico@epoka.edu.al

Abstract. The Internet of Things (IoT) has found a wide application in different domains. One domain in which IoT has considerable impact is in Education. In this context, IoT plays an important role in obtaining information about the environment and actors in which the educational processes take place. This paper presents a systematic review of the literature identifying how IoT is integrated in Education to enhance personalization of online learning processes. To achieve this, we conducted an extensive mapping study analyzing the different techniques and methods used to integrate IoT in Education with particular focus on Personalized Online Learning (POL). Our mapping study involved screening of more than 6400 research and academic articles, from which 65 papers were identified as having implemented experimental or solution-based research involving IoT. From preliminary results we determined that the use of IoT in Education to implement POL is in its early stages. This study highlights the possible gaps in implementing IoT in POL and suggests future work opportunities for researchers, educators, and practitioners.

Keywords: Personalized Online Learning, Internet of Things, Systematic Mapping Study.

1. Introduction

The number of objects connected to the internet has increased at an unprecedented rate, turning the concept of the Internet of Things (IoT) into a reality [4]. IoT is becoming more pervasive and it is used in several domains like healthcare, transportation, manufacturing, agriculture, commerce, and energy [75]. In addition to the aforementioned domains, IoT has also been implemented successfully in education domain where it has not only changed the traditional learning and teaching practices, but has brought changes to the infrastructure and teaching methodology of educational institutions [33]. Three main stakeholders that deliver services when IoT is implemented in education are instructors, students, and administrative staff [39].

Different authors have concluded that integration of IoT in education is shifting the structure of educational systems [52], [71]. With its ability to collect vast amounts of

* This study is an extended version of a study titled: "Personalized Online Learning based on IoT (Systematic Mapping Study)"

data, IoT can be a powerful technology in developing smart campus models. IoT provides enhanced learning tools, Personalized Learning (PL) experiences and smart education delivery [33], [59]. Use of IoT in education spans the e-learning ecosystem, provides new research opportunities, boosts engagement in learning and e-learning processes, and increases creativity and collaboration through its hyper-connectivity properties [1]. IoT is being integrated into education to enable PL experiences through smart classrooms equipped with sensors, cameras, beacons, and wearable devices. These technologies collect data on student behavior, learning preferences, and performance, allowing for dynamic adjustments to lesson pace and difficulty based on individual learning styles and real-time performance. [5], [100].

Despite increasing interest and acknowledgment of its potential advantages, the incorporation of IoT into Personalized Online Learning (POL) is still insufficiently underexplored. Existing literature predominantly focuses on distinct facets of integrating IoT into POL, with limited comprehensive studies addressing this intersection. A systematic mapping study within this domain is necessary to identify key trends, gaps, and future research directions. This study fills that gap by conducting a systematic literature review of how IoT technologies are being integrated into existing POL systems, enhancing the adaptability and effectiveness of learning environments.

The primary objective of this systematic mapping study is to investigate the current state of research on the integration of IoT into POL systems. To achieve this, we address the following key research question:

– How can IoT be effectively integrated into existing POL systems?

This study encompasses peer-reviewed journal and conference articles published within the last six years, specifically focusing on the integration of IoT technologies into POL systems. Articles that exclusively address IoT or POL, without examining their intersection, have been mostly omitted from the scope of this research. To systematically map the existing body of research, we adhered to a widely recognized protocol commonly used in engineering studies [68]. This approach ensures a rigorous, methodical, reliable, and reproducible process, incorporating a comprehensive literature search, selection of relevant studies based on predefined inclusion and exclusion criteria, as well as structured data extraction and synthesis. This study provides significant contributions to both academic literature and practical applications by exploring the integration of IoT in POL environments. Main contributions of the work include:

– Scoping of the research which use IoT within POL and identifying IoT integration technologies, methods, models and tools;
– Identifying the volume and nature of existing research within this field;
– Mapping the publication frequency trends over time in IoT-enabled POL;
– Assessing the current implementation and utilization of IoT in POL systems;
– Evaluating the application of IoT to personalize online learning across different educational levels;
– Highlighting research gaps in the implementation of IoT to personalize or adapt online learning to enable guidance for future studies.

The structure of the study is outlined below. Section 2 presents the research background. In Section 3, we describe the systematic mapping approach implemented for paper selection. Section 4 provides an analysis of the results and addresses the research

question (RQ). Finally, Section 5 concludes the study and outlines potential areas for future research. After this study-based review, we present the background of our research.

2. Research Background

Personalized Online Learning (POL) is an educational methodology that uses electronic and distance learning, computer algorithms, and artificial intelligence to adapt online learning services like content, path, exercises, and feedback to recommend appropriate educational content. The use of emerging technologies, such as the Internet of Things (IoT) in education bridges the gap between the physical and virtual worlds by integrating physical objects and environments with virtual platforms, making the learning process more interactive and immersive [41], [26], [81], [98], [95].

We define the Internet of Things (IoT) in education as an emerging technology that enhances POL by integrating data from learner physical environment, socio-psychological and behavioral attributes during their interactions with existing PL frameworks. IoT integration in education improves student health and safety monitoring, attendance tracking, and facilitates ubiquitous learning. It enhances learning by boosting retention, accelerating progress, fostering critical thinking, and enabling dynamic communication between students and teachers [50], [10].

Integrating IoT into Learning Management Systems (LMS) enhances flexibility, supports self-directed learning, enriches teaching and learning experiences, optimizes assessments, and promotes collaboration among stakeholders [54]. Integration of IoT in e-learning has significant impact on student efficiency, learning experience and educational productivity. It also offers high levels of flexibility, accessibility, adaptability, and scalability [71]. IoT can also be used to create more significant learning spaces by engaging physical objects in learner environment with digitally association within a particular learning subject [35]. It is assumed that IoT has tremendous potential in education, particularly with the implementation of the ”hypersituation” concept, which amplifies the information generated by multiple sources in real-time, thereby providing contextualized learning framework supported by the surrounding environment [59]. With the application of big data analytics, Adaptive Learning Environments (ALEs) can be significantly reshaped, reducing the risk of failure for online learning professionals and online learners [45].

IoT, Cloud Computing, and Learning Analytics (LA) enable personalized, interactive learning by replacing rigid curricula with data-driven, customizable approaches. These technologies provide insights into student and teacher behaviors, and supporting evidence-based decision-making in education. Although still in its infancy, analyzing behavioral patterns via IoT data shows promise for creating decision support systems tailored to the individual learner [58].

This study considers the possible methods, technologies, processes, and set of practices used in integrating IoT properties, technologies, protocols, and frameworks into personalized or adaptive online learning systems. Additionally, the study examines systematic reviews in IoT in education and future trends of IoT in education literature with a special focus on POL systems.

2.1. Existing systematic reviews in IoT in Education to Personalize Online Learning

Use of IoT in the Education domain with a special focus on POL is in its early stages. Sneesl et al. (2022) conducted a systematic review concerning the theoretical framework of the adoption of IoT technology related to smart campus adaptation [78]. Laksmi et al. (2022) carried out a systematic review with a focus on the IoT platform's use in educational processes especially in teaching and learning processes [46]. A state-of-the-art review on integrating Industrial Revolution 4.0 and IoT in academia to identify the factors that affect the level of motivation toward the integration of these technologies was performed by Butt et al. (2020) [15]. Asad et al. (2022) executed a review study on sophisticated insights regarding the impact of IoT-based smart laboratories on student academic performance in higher education in terms of conceptual and practical implementations [7]. Adi et al. (2020) conducted a critical review on how IoT-generated data are processed for Machine Learning (ML) analysis and current challenges in furthering intelligent solutions in the IoT environment with a proposed framework for ML and knowledge discovery for IoT to integrate adaptive learning techniques locally, at the edge, through a fog or in the cloud [3]. Chen et al. (2021) carried out a comprehensive review to understand the past, present, and future academic structure, major research topics, and contributors in smart learning research [17].

2.2. Use of IoT in Education to Personalize Online Learning

The use of wearable devices like smartwatches, fitness trackers, and biometric sensors is growing in education [9]. Schools and universities use these tools to collect data on student behavior, physiological responses, and health metrics [2]. This data personalizes learning by detecting focus loss and adjusting lessons or monitoring biometrics like heart rate and movement for learning insights [19]. In addition, IoT devices and LA provide data on student behavior and learning preferences, enabling PL plans to better achieve individual goals [101].

IoT in Education has proved to be effective in improving teaching and learning activities by analyzing the impact of physical environmental properties such as temperature, noise pollution and CO_2 level on student attention, and in observing student reactions to lectures with the use of sensoring and monitoring technologies which provide real-time feedback [8], [9], [79].

Adaptive learning platforms use data analytics and ML algorithms to personalize online learning by analyzing IoT-collected data to identify where students are struggling and provide additional support or resources as needed [30]. In addition these platforms can use sensors to monitor learner responses and adjust content or pace accordingly [80]. Of note, Yakoubovsky and Sarian (2021) combine IoT and Electroencephalography (EEG) device to monitor student brain activities to assess student levels of engagement to provide a better POL experience [94].

2.3. Challenges of integrating IoT in Educations

Integration of IoT and data mining in e-learning brings challenges and opportunities to educational institutions and students [52]. Some of the challenges identified by Alfoudari

et al. (2021) in their study were lack of engagement, interactivity and personalization of external factors, learning attitudes and teaching methods [5]; whereas some of the challenges addressed by Ennouamani and Mahani (2017) were finding the most influential characteristics of e-learners and proper integration of adaptive learning platforms modules [30]. Privacy and security are commonly identified challenges by authors. In addition, the size and heterogeneity of data that need to be handled were identified by Arishi and Mavaluru (2018). Ghallabi et al. (2020) highlighted challenges related to the high costs of sensors, the battery life of IoT devices, wireless coverage, adaptation of m-learning, and AR. Meanwhile, Kassab et al. (2020) identified scalability and humanization as key issues [39], [52]. In addition to identified challenges, typical responsibilities of students and teachers are fundamentally changed in IoT-based e-learning settings and there is an imminent need for innovative teaching methods to enable integration of IoT in education [71], [10]. E-learning entities and platforms have faced criticism due to completion rates, costs, employment, and career readiness, but integration of new technologies and IoT applications can help improve e-learning platforms to overcome these challenges [92]. Considering this background in the use of IoT and POL, this paper will address implemented research methodology.

3. Research Methodology

3.1. Motivation for conducting the systematic literature mapping

The objective of this study is to identify to what extent and how IoT is implemented into existing POL models. Its specific focus is on the intersection between IoT and POL, exploring this intersection on how IoT is implemented in POL. To achieve a complete overview of the area, we reviewed IoT in Education papers that address POL methods, techniques and frameworks identified in Section 2.1.

There are different reviews and mapping studies regarding POL or IoT as described in Section 2.1. Chweya and Ibrahim (2021) investigated implementation of IoT in learning institutes and concluded there was a lack of studies in respect to models and methodologies used to implement IoT in Education [20]. In their systematic literature review, Kassab et al. (2020) stated there were few, if any, consolidated and coherent reviews implementing IoT in education [39]. In addition, within the last decade there has been no systematic map addressing common ways of implementing IoT in POL. Our mapping study allows researchers and stakeholders to make informed decisions regarding implementation of IoT in online learning by taking into consideration present and future challenges and requirements.

3.2. Systematic mapping study

There are different guidelines that can be implemented to conduct systematic mapping; [42], [67], [43], however, Petersen et al. (2015), consolidated the best practices suggested by other researchers, which enabled a clearer structure for understanding the types of research reports and results that have been published by categorizing papers [68]. The process begins by formulating research questions (RQs) with the next steps involve screening papers based on their title, abstract, and keyword metadata specific to the RQs. The primary goal of a systematic mapping study is to identify the gaps in the research area being investigated. The conducted systematic mapping process is represented in Figure 1.

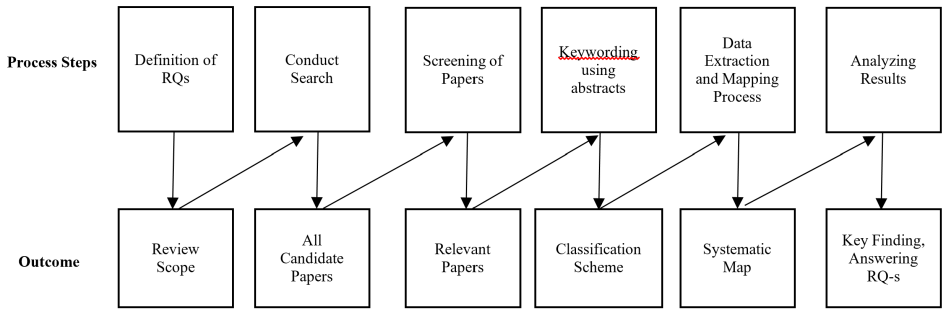


Fig. 1. The systematic mapping process [68]

3.2.1 Definition of RQs

Based on study motivation and objectives (Section 3.1), we proposed the following RQ to guide the search and selection process:

R.Q.1: How can IoT be integrated within existing Personalized Online Learning systems? The research question is composed of seven sub-questions (listed in Table 1) to analyze different aspects of integration of IoT in POL.

Table 1. R.Q.1. sub-questions

RQ.	Research Sub-question
RQ1.1	What are potential techniques used to integrate IoT in existing POL systems?
RQ1.2	What are the POL models or strategies that may be added or enhanced by using IoT?
RQ1.3	What are the POL system input components or parameters that may be added or enhanced by using IoT?
RQ1.4	What are the tools or methods used to measure POL components or parameters?
RQ1.5	What are the POL algorithms that may be added or enhanced by using IoT?
RQ1.6	What are the IoT tools that may be used in POL?
RQ1.7	Which research approaches are used in academic literature related to the integration of IoT into POL and what contribution has been achieved?

3.2.2 Conducting the search

Identifying the search string of publications was the first step in conducting a systematic research process. To identify keywords and formulate search string from research questions we followed PICO (Population, Intervention, Comparison and Outcomes) criteria developed by Petersen et al. (2015) [68]. These criteria are defined in Table 2.

To explore how IoT is integrated in POL systems in primary studies we formulated a search string that reveals combination of IoT and POL systems. On this study we extracted information on different aspects of IoT integration to POL systems. The extracted information consisted of IoT integration techniques (RQ1.1), IoT POL models (RQ1.2) and personalization components (RQ1.3). In addition, tools or methods for measuring person-

alization components (RQ1.4), algorithms used to process data providing personalization services (RQ1.5), and IoT tools used (RQ1.6), are also extracted.

Table 2. PICO of the study

Dimension	Description
Population	Primary studies which integrate IoT in POL systems (both theoretical and empirical studies).
Intervention	Techniques, models, components, algorithms, and tools used to integrate IoT in POL systems.
Comparison	Comparing different techniques, models, components, algorithms and tools used to integrate IoT in POL systems.
Outcome	Evaluation of collected empirical studies which integrate IoT in POL systems.

The details of generating keywords and classification scheme are described in Section 3.2.4.

There are three main concepts of the search string "Internet of Things", "Personalized", and "Online Learning" derived from research questions. Our basic string was "Internet of Things" AND "Personalized" AND "Online Learning". To extend the scope of research we conducted a literature investigation for synonyms and interchangeable terms which identified main concepts. Various terms related to the "Internet of Things" have been used by different authors, standardization bodies, alliances, projects, and industries, often with similar or different meanings. Main terms extracted from literature for the first concept are: Internet of Everything, Machine to Machine, Cloud of Things, Web of Things, and Network of Things. Regarding the second concept, we identified Adaptive, Individualized, Tailored, Customized, Context-aware and Learner-centric keywords. Meanwhile, for the third concept, we retrieved e-Learning, Elearning, Distance Learning, Virtual Learning, Web-Based Learning, Electronic Learning, Digital Learning, Distance Education, Computer-Supported Learning and Internet-Based Learning terms [39], [30], [20], [47], [103], [36]. To construct the full string, we used AND Boolean operator to combine main concepts and OR Boolean operator to express synonyms or interchangeable terms of main concepts. The constructed full search string was ("IoT" OR "Internet of Things" OR "IoE" OR "Internet of Everything" OR "Industry 4.0" OR "Web of Things" OR "WoT" OR "Machine to Machine" OR "M2M" OR "CoT" OR "Cloud of Things" OR "NoT" OR "Network of Things") AND ("Personalized" OR "Adapted" OR "Individualized" OR "Tailored" OR "Customized" OR "Context-aware" OR "Learner-centric") AND ("Online Learning" OR "Distance Learning" OR "Distance Education" OR "e-learning" OR "Elearning" OR "Electronic Learning" OR "Virtual Learning" OR "Digital Learning" OR "Computer-Supported Learning" OR "Internet-Based Learning").

Various systematic literature reviews regarding IoT or POL used several options for electronic library and database selection such as ACM Digital Library, IEEE Xplore, Education Resources Information Center (ERIC), Scopus, Springer Link, Web of Science, Science Direct, and DBLP bibliography. To conduct our research, we implemented the search string in ACM and IEEEXplore digital libraries and on Education Resources Information Center (ERIC) and Springer Link databases. We selected ERIC as the main education research and information databases, while ACM, IEEEXplore digital libraries

and Springer Link database as main sources for engineering related articles. To determine latest trends we limited the research from January 2017 to March 2023. Detailed results of implementation of search string and results are show in Table 3.

Table 3. Number of unique studies and search strings (Time interval: 2017 - 2023)

Source	Basic string	Full String	Invalid Resources	Duplicates within Search Strings	Duplicates between Databases	Library Total
ACM	305	165	79	13	23	355
ERIC	8	898	7	0	0	899
IEEEXplore	38	150	1	9	0	178
Springer Link	2363	2514	1	565	4	4307
Total	**2714**	**3727**	**88**	**587**	**27**	**5739**

3.2.3 Screening of paper

Based on the RQ and Inclusion and Exclusion criteria of this mapping study, the most relevant papers were identified during the screening process. Based on study title, abstract and keywords, we included or excluded each study found within the search string. From the database search, we identified a total of 6441 papers published during 2017 – 2023 time span. Most of the publications are in the Springer Link digital database.

Phase 1 involved the automatic removal of 88 invalid sources, such as workshop programs, keynotes, book covers, speeches, retracted articles, PhD theses and unpublished works. Furthermore, with the help of spreadsheet software we automatically removed 614 duplicate papers. For Phase 2, we relied on 5739 remained references.

In Phase 2, we applied additional filtering, based on the inclusion and exclusion criteria, Table 4, to titles and keywords, producing 2510 references, and then we manually excluded 2405 articles based on titles and then based on abstracts, as recommended by [47] resulting on 96 candidate studies.

In the last Phase, after full text reading of candidate studies we excluded 31 papers as not totally related to our systematic mapping study, resulting in 65 accepted studies. The number of included and excluded papers for each Phase is presented in Figure 2.

3.2.4 Keywording using abstracts and classification scheme generation

The goal of keywording in the context of systematic mapping studies, according to [67], is to facilitate the classification and organization of research papers. This process, which involves identifying key terms from abstracts, creates a structured overview of a research area. This systematic approach aids in categorizing the studies, making it easier to identify trends, research gaps, and the distribution of research efforts across different topics within the field.

To classify the papers, we followed the process documented in Fig 3, as proposed by [41], who rated classification schemes based on the basis of a set of quality attributes: **Scheme Definition:** The taxonomy/classification is created on the basis of an exhaustive

Table 4. Criteria used for including and excluding research studies

Inclusion Criteria	Exclusion Criteria
– Included content of IoT that investigates Educational perspectives to Personalize Online Learning i.e., methods, frameworks and use cases; – Published between January 2017 and March 2023; – Written in English with full-text available; – Peer-reviewed journal and conference literature;	– Duplicate papers from the same study in different databases. – Mention of IoT is tangential with different scopes not directly related to POL; – Publications not written in English; – Publications not directly related to our topic; – Full-text is inaccessible; – Books and gray literature;

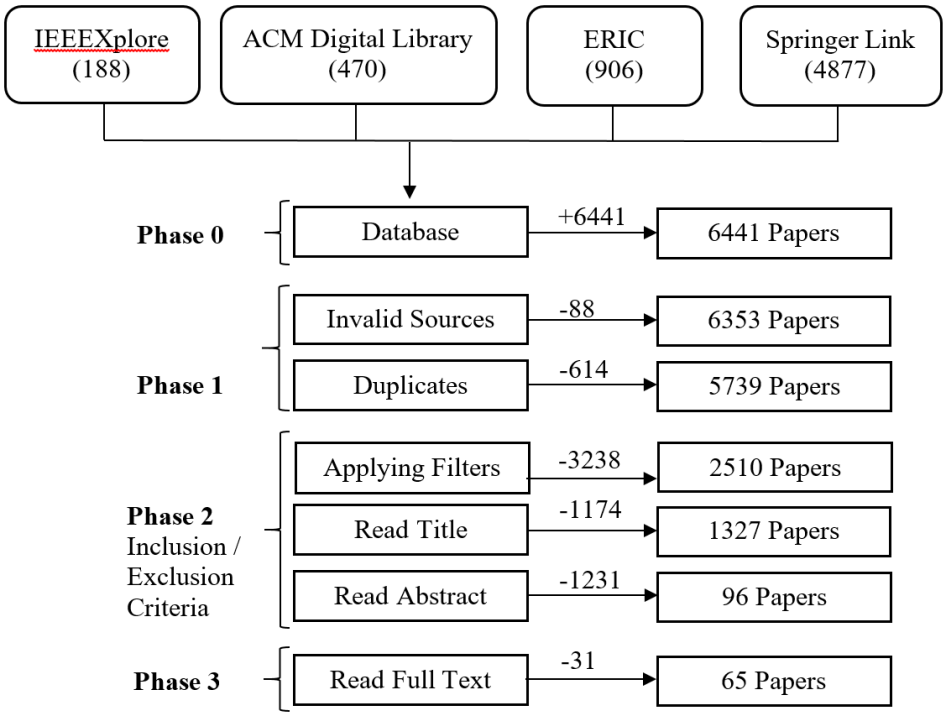


Fig. 2. Selection of primary studies [68]

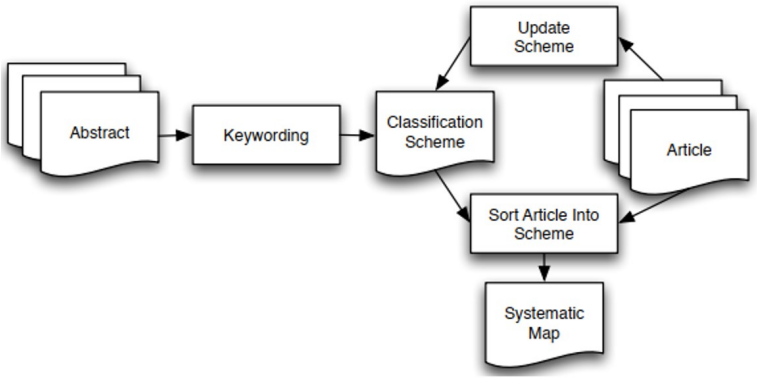


Fig. 3. Classification Scheme Generation Process [41]

analysis of existing literature in the field.
Scheme Terminology: The taxonomy uses terms that are used in existing literature.
Orthogonality: There are clear boundaries between categories which makes it easy to classify.
Complete: No categories are missing, allowing existing articles to be classified.
Accepted: The community accepts and knows the classification/taxonomy.

In Figure 4 we represent the classification scheme where all facets are derived from the existing literature on the subject, as well as the results of our own study. To build the classification scheme we conducted the following steps as described in [67]:
Step 1: Reviewing the abstracts and keywords of primary studies identified the main concepts for these studies. When abstracts were not sufficient, the introduction and conclusion sections of the papers were also analyzed. Extracted keywords are strictly related to the RQ to classify and map papers accordingly.
Step 2: Combining together the set of keywords and concepts from different papers, determined the categories representative of the research area of the relevant primary studies. Additionally, these categories were refined and updated to accommodate new data.
Step 3: Categories formed by final set of clustered keywords are classified into higher hierarchical levels to develop a high level of understanding about the nature and contribution of selected primary studies. The following section presents key findings of the systematic mapping study, offering insights that directly reflect our objectives. To outline the results, we followed the structure and layout of a systematic mapping study conducted by Cico et al. [21].

4. Results

As indicated above, our main research questions was: “How can IoT be integrated within existing POL systems?”. From an initial sample of 6441 papers, 65 primary relevant studies were selected for answering RQs. In order to analyze the main dimensions of research question, seven other sub-questions were included (e.g. R.Q.1), with the results allowing for each article to be classified into categories pertinent to specific sub-questions.

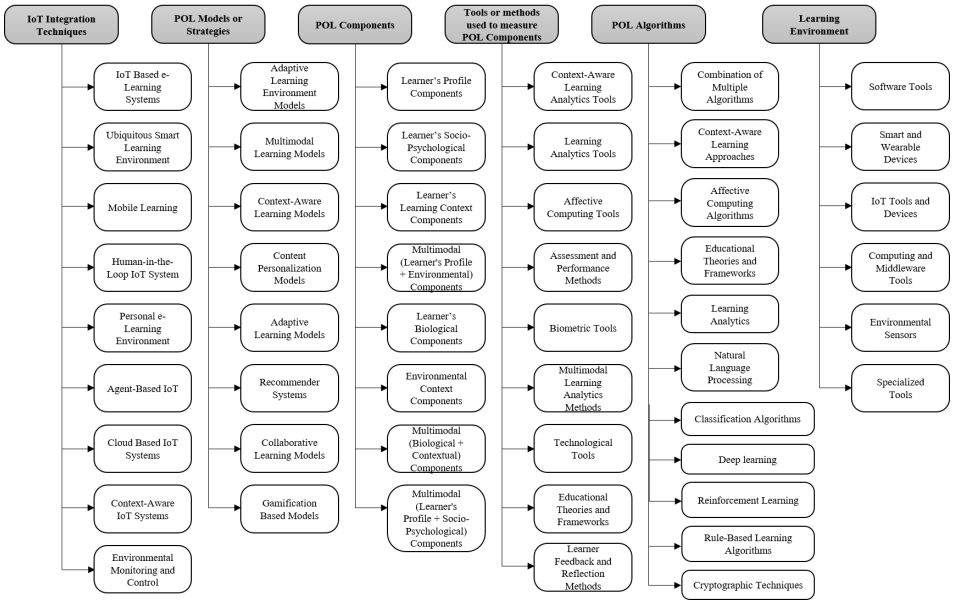


Fig. 4. Classification scheme derived from keywording using abstracts

4.1. Descriptive analysis of results

This section presents a comprehensive descriptive analysis of the selected studies focusing on distribution of articles over years and the databases where they were retrieved from.

Table 5 presents the distribution of publications from January 2017 to March 2023 and their dissemination across different digital databases and libraries.

Table 5. Distribution of selected studies over years and digital libraries or databases

Publication Year	Number of Publications	Percentage	ACM	ERIC	IEEE	Springerlink
2017	8	12%	3	5	0	0
2018	8	12%	0	3	1	4
2019	15	23%	2	6	4	3
2020	11	17%	1	6	1	3
2021	10	15%	2	4	2	2
2022	8	12%	2	3	1	2
2023	5	8%	0	1	1	3

The integration of IoT into POL systems is an interdisciplinary field between information technology and education. ERIC database contains scholarly articles related to education and the integration of different technologies into education to improve the quality of learning. Given this context, it makes sense that nearly half of the primary studies,

specifically 43% of the total, are found in it. The Springer link digital database is the second most utilized source, with 17 publications, representing 26% of the total. In addition, 69% of selected studies were journal articles and 31% of were published in conference proceedings.

4.2. Answering Research Questions

In the next section we provide the results of research questions and each research subquestion by clustering and analyzing them according to main classification categories.

4.2.1 What are potential techniques used to integrate IoT in existing POL systems?

To answer this question, we created the "IoT Integration Techniques" classification categories of the selected studies, as shown in Table 6. From summarized results we can

Table 6. Number of papers according to used IoT Integration Technique

IoT Integration Technique	Number of Studies	Percentage
IoT Based e-Learning Systems	12	18%
Ubiquitous Smart Learning Environment	11	17%
Mobile Learning	10	15%
Human-in-the-Loop IoT System	9	14%
Personal e-Learning Environment	9	14%
Agent-Based IoT	6	9%
cloud-based IoT Systems	3	5%
Context-Aware IoT Systems	3	5%
Environmental Monitoring and Control	2	3%

observe that about 35% of the papers have as main focus IoT Based e-Learning Systems and the creation of Ubiquitous Smart Learning Environment.

The most commonly used IoT integration technique into POL is the development of **IoT Based e-Learning Systems** that focus on delivering personalized or adaptive learning experiences by leveraging IoT technologies. These systems are used to recommend appropriate content to learners using multimedia recommendation schemes powered by IoT [41]. They combine content-based information with collaborative user preferences [25] and utilize ML algorithms to analyze IoT collected data [3] to provide personalized recommendations. In addition, IoT based tools for emotion detection adapt learning paths based on facial expressions [40], [80], while sensor-driven activities enable hands-on, environment-specific learning experiences [62]. Additionally, deep learning and multi-classification algorithms personalize content by classifying students based on their learning data [86]. IoT technologies further enable real-time data mining to analyze student engagement [31] and support smart laboratories that foster interaction and creativity in learning environments [7]. Lastly, an innovative IoT system categorizes students by learning styles (visual, auditory, kinesthetic) using fingerprint analysis, while tracking their academic performance [74].

The second most commonly used IoT integration technique, nearly 17% of selected papers, is the creation of **Ubiquitous Smart Learning Environment** (USLE). Several educational strategies and technologies are used in designing USLE, such as Augmented Reality (AR), Virtual Reality (VR) [51], [89], [88], [65], [38], ubiquitous computing and learning [53], [76], PL spaces [102], and an ambient and pervasive PL ecosystem [12] to create a comprehensive and immersive learning experience. Ubiquitous Computing, Mobile Computing, and IoT are integrated to develop learning scenarios for middle school education, although without detailed IoT-based personalization techniques [53]. The VILLAGE project explored the use of virtual environments, chatbots, and time machines to enhance the immersion of language learners and personalize experience through interactive digital artifacts [88]. AR techniques are employed in Personalized Learning (PL), allowing interaction with paper-based learning sheets that are enhanced by AR. This supports the content with online resources, such as YouTube videos [89]. Authors in [65] focused on the use of virtual laboratories as a complement to traditional laboratories for teaching chemistry concepts. A novel generic architecture for context-aware ubiquitous learning systems, based on the IoT computing paradigm, is implemented using common IoT edge devices. Four variants of the architecture are presented, each based on different protocols [76]. Authors in [102] used IoT devices to create and measure the effectiveness of PL space in the educational metaverse. Meanwhile, [13] focused on the implementation and testing of an IoT Dataflow system in classroom settings, primarily to enhance K-12 science education through hands-on experiments. Furthermore, authors in [51] experimented on advanced techniques like 3D presentations using AR, VR, smart assessments, digital libraries, and cloud integration for effective data management in educational frameworks. Additionally, game-based systems and educational data mining were employed to provide customized experiences with AR and VR to enhance content delivery and retention [12].

Mobile Learning (m-learning) techniques use mobile devices mainly to extract the context where the learning takes place [83], [96], [60], [48], [22] and as a learning device tool [64], [14], [56], [85], [66]. In context-aware m-learning systems authors used Global Positioning System (GPS) and other mobile phone embedded sensors to provide adaptive learning content based on the learner location and environmental conditions [96], [48], [83] and to adapt the learning path in mobile and ubiquitous learning environment [60]. Authors in [64] outlined a m-learning framework specifically tailored for low-income economies, emphasizing the importance of aligning theoretical approaches with practical m-learning strategies in resource-constrained environments. Additionally, authors in [14] proposed an adaptive m-learning environment that adjusts educational content and strategies to fit the learner cognitive, behavioral, emotional, psychological, and sociological profiles. Meanwhile, advanced technologies like wearable devices and smart glasses are explored by [85] to enhance interactivity, assessment, and POL experiences. Moreover, embedded biosensors (noninvasive, low-cost, and distraction-free) used in smartphones and smartwatches are investigated by [22] to implement the adaptivity within the Education 4.0 processes.

Human-in-the-loop IoT systems incorporate human feedback and interactions into the decision-making process of POL. These systems are explored in [82], where wearable EEG devices are used to infer a learner cognitive state and detect changes in learning performance. In [28], a Reinforcement Learning (RL)-based framework is proposed to

adapt IoT applications according to human variability, utilizing RL to account for dynamic learning behaviors. Additionally, an Intelligent Music Selection System is introduced by [18], utilizing wearable sensing devices to measure Heart Rate Variability (HRV) and predict user emotions. This enables automatic song recommendations based on physiological responses. Meanwhile, authors in [27] propose a Domain-Specific Modeling Language (DSML) designed to integrate IoT data sources and model human emotions to personalize online learning experiences based on emotional states. Lastly, authors in [77], used smart glasses and Raspberry Pi to explore real-time student behavioral analysis through facial recognition algorithms, meanwhile, authors in [63] assessed the feasibility of wearable physiological sensing technologies such as Electro-Dermal Activity (EDA), Photoplethysmography (PPG), and Skin Temperature (ST) to measure student engagement levels.

Personal e-Learning Environment mainly consists on personalized recommendation systems [72], [26], [11], [99] and on adaptive e-learning environments [90], [91], [55]. These systems encompass a variety of approaches to enhance e-learning experiences. These include course recommendations tailored to a learner profile and preferences [66], the use of collaborative and content-based filtering techniques to adapt learning materials to individual student needs [11], Technique Feature Analysis (TFA) for recommending personalized word-learning tasks [99], and a learning path recommendation system that generates PL paths for e-learners, based on learning styles, knowledge levels, and learning goals [26].

Adaptive e-learning environments integrate a variety of innovative methods. Authors in [90] discussed the adaptation of the e-learning environment based on the learner personality and emotional state. Meanwhile, [91] focused on the use of apps that adapt to the learner specific situation and needs. Additionally, a dynamic and student-controlled learning space is achieved by integrating Personal Learning Environments with LMS through a service-based framework, as outlined in [55]. In addition, a system that combines Explainable AI and Internet of Behavior technologies to assess student behavior and personalize learning materials is developed by [29].

Agent-based IoT integration techniques commonly incorporate Educational Chatbots (ECs) to deliver POL services. In [44], authors developed an EC focused primarily to support project-based learning and examined its impact on learning outcomes. Meanwhile, in [24], an EC was designed to respond to student queries by detecting questions using Natural Language Processing techniques and domain ontologies. Additionally, virtual assistants in the educational sector, developed with Google Dialogflow as described by [73], concentrate on abstracting knowledge and generating responses. This functionality facilitates effective interactions by analyzing course materials and managing entities and intents. Lastly, intelligent agents are employed by [84] to track student knowledge states by analyzing their interactions to recommend personalized content.

Cloud-Based IoT Systems include a conceptual framework proposed by [69] to enhance analytical thinking and support PL by adapting to individual learner needs through problem-solving and cloud-based resources. Additionally, an approach to integrate personalization parameters as a collection of web services uploaded to cloud databases to model the learner is proposed by [32].

In **Context-Aware IoT Systems**, contextual parameters such as location, noise level, and time of day are used by [97] to build user profiles for selecting PL materials. Ad-

ditionally, the impact of environmental variables like temperature, humidity, air quality, light, and sound on attention, motivation, and attitudes is assessed through a prototype and a Perception of Academic Achievement questionnaire [8].

IoT integration in POL enables adaptive and context-aware learning experiences through IoT-based e-learning systems, USLEs, and m-learning. VR, AR, and context-aware systems enhance immersion and personalization. Meanwhile, cloud-based ECs and intelligent agents promote improved interaction and adaptive content delivery.

Key Findings:

- IoT-Based Personalized Learning Systems integrate IoT and ML algorithms to classify students based on their learning data in order to provide adaptive learning paths or content based on individual needs and preferences;
- Ubiquitous Smart Learning Environments integrate IoT, mobile computing, augmented and virtual reality to create immersive learning experiences;
- Mobile and Context-Aware IoT Systems use mobile devices and wearable technologies to provide adaptive learning content based on learner context and preferences.

4.2.2 What are the POL models that may be added or enhanced by using IoT?

In the realm of educational technology, various models and strategies enhance learning experience through personalization and adaptability, as shown in Table 7. Adaptive Learning Environments and Multimodal Learning Models are among the most commonly identified models in POL systems.

Table 7. Number of papers according to used POL Models or Strategies

POL Models or Strategies	Number of Studies	Percentage
Adaptive Learning Environment Models	14	22%
Multimodal Learning Models	13	20%
Context-Aware Learning Models	11	17%
Content Personalization Models	8	12%
Adaptive Learning Models	7	11%
Recommender Systems	6	9%
Collaborative Learning Models	3	5%
Gamification Based Models	3	5%

Adaptive Learning Environment Models includes PL spaces that are cognitive load-aware, particularly in the context of the metaverse [102], as well as dynamically adjustable and student-controlled learning environments [55]. Learning paths are personalized through recommendation systems [26], and further optimized using models that sequence learning objects and automatically construct course structures [60]. Additionally, authors in [91] introduced the concept of an educational ecosystem, where technology and pedagogy fill specific niches, supported by applications and delivered through PLEs. Authors in [56] suggested creating a personalized cloud environment for m-learning to

make education more accessible. Additionally, an empirical study conducted by [90] explored the impact of adaptive e-learning environments based on learner personality and emotion.

Multimodal Learning Models integrate multiple aspects simultaneously to deliver an enhanced, personalized, and adaptive learning experience. These models utilize Multimodal LA to assess and classify student engagement in e-learning environments [16], [31], enabling adaptive learning scenarios [80] and automatically recommending songs based on the user physiological responses [18]. Additionally, wearable devices are used in [82] to monitor and adjust to the learner cognitive state in real-time by adapting content delivery according to the learner engagement and understanding levels. In [94], authors incorporated Digital Dynamic Modulation, which modulates video content with brain-stimulating frequencies to improve cognitive functions, such as attention and memory retention. Furthermore, an immersive learning model is implemented by [57] using multisensory effects such as olfactory cues, vibrations, and airflow to enhance the overall learning experience.

Context-Aware Learning Models can use learner socio-cultural and socio-economic contexts to support PL experiences [64]. These models can enhance the Felder-Silverman Learning Style Model (FSLM) by incorporating context-awareness, gathered from sensor data, into each component of the FSLM framework [83]. Additionally, they can provide personalized, context-aware recommendations using GPS technology [96]. These models can increase learner attention and motivation by integrating data from environmental sensors into existing POL algorithms or systems [8]. Furthermore, they can utilize low-level multi-domain context data to generate high-level contextual information [37].

Content Personalization Models adapt learning materials based on various factors, including the learner environmental context and knowledge level [97], as well as learner environmental context and behavioral analytics [3]. These models may also personalize content according to the learner characteristics, preferences, background, behaviors, and learning style [14]. Additionally, educational content can be tailored into a structured chatbot format [73], or recommendations can be personalized using knowledge tracing techniques [84].

Adaptive Learning Models dynamically adjust learning materials and experiences to address the unique needs, abilities, and preferences of individual learners. An experiment on adaptive learning through learner profiling in a cloud-based approach is conducted by [32]. A problem-solving-based PL model, emphasizing PL through problem posing to deliver tailored content and resources, is implemented by [69]. The authors in [89] explored a blended learning strategy that combines AR-based and online-based content delivery to adapt to individual learning speeds and preferences. Additionally, an adaptive teaching model, where instructors modify class activities based on student feedback, is proposed by [51].

Several **recommendation techniques** are identified from selected studies to provide tailored educational resources including online courses recommendations [66] news and related multimedia recommendations [41], learning resources recommendations [79], collaborative and content-based filtering recommendations [87], and social recommendations [34]. A unique hybrid and semantic recommendation system that combines content-based information with collaborative user preferences is proposed by [25].

Collaborative Learning Models explore ECs to facilitate project-based learning, where they provide personalized feedback, simulate peer interactions, and enhance reflection and metacognition [44]. Furthermore, the Student - Instructor - Technology Environment model is employed by [66], focusing on the interaction between these elements to enhance collaborative e-learning experiences.

Gamification Based Models focus on increasing student engagement and personalization through game-like elements. For instance, the GAIA Challenge game adapts the learning experience by progressively revealing new sections of the game map as student complete different stages, offering a PL trajectory [62]. Additionally, the integration of gamified elements like badges and biofeedback methods are implemented by [22] to boost student motivation and support academic success.

Key Findings:

- Adaptive Learning Models play a central role in personalized education by dynamically adjusting learning materials and experiences to meet individual learner needs, abilities, and preferences;
- Multimodal Learning Models provide a highly adaptive learning experience by integrating multiple forms of inputs and LA to adapt content, enhancing engagement and cognitive performance;
- Context-aware learning models enhance existing personalized or adaptive online learning frameworks by integrating environmental, socio-cultural, and socio-psychological data through sensor-based inputs.

POL models include ALEs, primarily used to create dynamic, user-controlled PL spaces or cloud-based PLEs. Multi-modal learning models integrate multiple aspects simultaneously, delivering an enhanced, personalized, and adaptive learning experience. Context-aware learning models are utilized to improve existing POL models by integrating contextual data or to develop new context-driven POL models. Content personalization models adapt content based on the learner's profile and environmental context. Recommendation-based POL systems suggest courses or learning resources to enhance POL services. Additionally, models such as Collaborative Learning and Gamification foster engagement and provide tailored resources through dynamic, interactive, and game-based approaches.

4.2.3 What are the POL system components or parameters that may be added or enhanced by using IoT?

POL systems integrate various components to adapt to individual learner needs and preferences. As depicted form Table 8, the integration of different types of components is the most commonly implemented approach in the selected studies to provide POL services.

Nearly 40% of studies offer personalization services by either focusing solely on the **learner profile** or by combining it with environmental or socio-psychological components. The main elements used to construct the learner profile include prior knowledge, learning style, and learning preferences. Additional factors considered are the learner personality, emotions, cognitive state, intentions, and stress levels. Demographic information such as age, academic major, and academic performance is also incorporated. Further-

Table 8. Number of papers according to used POL Components

POL Components	Number of Studies	Percentage
Learner Profile Components	18	28%
Learner Socio-Psychological Components	12	18%
Learner Learning Context Components	11	17%
Multimodal (Learner Profile + Environmental) Components	11	17%
Learner Biological Components	5	8%
Environmental Context Components	3	5%
Multimodal (Learner Profile + Socio-Psychological) Components	3	5%
Multimodal (Biological + Contextual) Components	2	3%

more, the learner profile includes data on the learner activities and performance, as well as intra-human, inter-human, and multi-human variabilities.

The subsequent most frequently utilized personalization elements are **Learner Socio-Psychological Components**. The main socio-psychological components used in POL are the learner behavioral patterns [16], [14], [29], combined with user emotions [90], [18], [27] and other factors such as learner motivation and engagement [16], learning styles, and cognitive load [14]. Additionally, anxiety levels in virtual learning [38], immersive tendencies [88], social relationships [34], affective interactions [44], intention inference [24], and engagement [63] are considered to provide POL experiences.

Learner Learning Context Components are utilized to extract the learner context, enabling the delivery of POL services. High-level context information is derived from low-level, multi-domain data sources [37]. In the m-learning framework [64], sociocultural and socioeconomic factors are also considered to tailor learning experiences. In addition, **Environmental Context Components**, such as the learner location [96], temperature, humidity, light [8], noise level, and time of day [97], along with graphical elements like layout and background texture [49], are utilized to create an optimal learning environment. These components foster immersive, collaborative, and PL experiences.

Learner Biological Components are critical elements used to build and enhance POL systems. These components primarily include physiological signals such as heart rate, brain waves, galvanic skin response, and visual attention metrics. For instance, the heart rate signal [102], [23], [80], [40], captured through PPG (Photoplethysmogram) sensors or smartwatches, is utilized to gauge learner emotional engagement and cognitive levels. Brain waves, monitored via EEG [23], provide real-time feedback on learner brain activity. Galvanic skin response along with heart rate [94], aids in detecting emotions with high accuracy, thereby allowing the system to adapt to the learner emotional state. Visual attention metrics, such as the number of faces, eye count, and eye status [102], combined with affective state monitoring, including indicators like frustration or engagement [80], collectively facilitate the development of ALEs. By tailoring content and interactions to align with each learner's physiological and emotional states, these systems aim to enhance overall educational outcomes.

POL systems utilize various components to tailor educational experiences to individual learners. These components include learner profiles, which comprise prior knowledge, learning styles, preferences, emotions, and demographic data, along with socio-psychological factors such as behavioral patterns, motivation, engagement, and emotional states. Additionally, biological signals like heart rate and brain waves, combined with environmental factors such as location, noise levels, and lighting, further enhance personalization by creating adaptive and immersive learning environments. Collectively, these elements enable POL systems to deliver responsive and customized learning experiences that cater to the diverse needs of learners.

Key Findings:

- Learner Profile components play a crucial role in providing POL services;
- Integration of IoT into existing POL systems incorporates biological and environmental context components;
- Integration of IoT into existing POL systems increase significance of Multimodal components.

4.2.4 What are the tools or methods used to measure POL components or parameters?

Integration of IoT into existing POL systems enable incorporation of contextual components generated by IoT sensors data. The main category of the tools or methods used to measure POL components or parameters, as shown in Table 9, are **Context-Aware Learning Analytics Tools**. These tools include IoT Edge Devices equipped with various sensors to represent the state of physical things and their context in ubiquitous learning environments [76] and to adapt and measure learning components based on the learner physical context and environmental conditions [48]. Additionally, contextual mapping techniques

Table 9. Tools or methods used to measure POL Components

Tools or methods to measure POL Components	Number of Studies	Percentage
Context-Aware Learning Analytics Tools	12	18%
Learning Analytics Tools	10	15%
Affective Computing Tools	9	14%
Assessment and Performance Methods	8	12%
Biometric Tools	6	9%
Multimodal Learning Analytics Methods	6	9%
Technological Tools	6	9%
Educational Theories and Frameworks	4	6%
learner Feedback and Reflection Methods	4	6%

are employed to measure and adjust learning paths according to the learner context, while a context-aware mechanism, integrates GPS, QR code interfaces, and learning logs to adapt the content [60], [96]. Furthermore, ML approaches are used to develop a context

fusion model, infering multi-domain context information from various IoT sensors [37], and to process data intelligently [3]. In addition, personalized m-Learning applications are used as context-aware measurement tools to provide POL services [83], [97].

Learning Analytics Tools and ML techniques, are employed to track and analyze learner engagement using wearable devices [16]. Additionally, TFA is used for personalized word-learning tasks, grounded in a comprehensive word learning theory [99]. Graph Neural Network (GNN) methods are implemented to encode user and item feature spaces for social recommendations, addressing correlations between user preferences and item features [34]. In another approach combination of metadata analysis (views, likes, comments) and readability assessments (Coleman-Liau index, Flesch-Kincaid grade level), along with the use of scoring mechanisms (time, popularity, difficulty) are used to measure and personalize online learning content [41].

The primary **Affective Computing Tools** used to understand and respond to learner behavior, personality, and emotional states include biometric-based tools such as Facial Recognition Algorithms and Raspberry Pi Cameras [77], Facial Emotion Detection using Webcams and Emotion Detection APIs [40], Emotion Recognition and Usability Assessment Tools [27], and Eye-Tracking Measurement Method [85]. Additionally, IoT devices including cameras, microphones, and inertia sensors [80] are employed to capture and analyze the learner affective state by tracking facial expressions, eye-gaze, heartbeat, and speech cues. Another commonly used approach involves questionnaires, such as the Myers-Briggs Type Indicator (MBTI) model for personality and the Ortony, Clore and Collins (OCC) model for emotion modeling [90], as well as the Immersive Tendencies Questionnaire, Igroup Presence Questionnaire [88], and the Attention-Scoring Model (ASM) [31]. These tools help to better understand and respond to learner affective states, enhancing the personalization and effectiveness of POL systems.

Assessment and Performance Methods primarily rely on data-driven performance evaluation and interaction analysis, utilizing data mining techniques [51] and ML algorithms [79], [70]. In addition, performance-based and observational assessment methods [7], along with pre- and post-tests [65], [22], are commonly employed. Qualitative methods to measure engagement metrics include student reflections, feedback, and interaction analysis [66].

The primary **Biometric Tools** and methods utilized in measuring components of POL systems include HRV for Physiological Data Analytics [102], [18], [23], EEG for Neuropsychological and Cognitive Measurement [82], [94], and the combination of EDA, PPG and ST as tools for engagement measurement [63].

Multimodal Learning Analytics Methods encompass a mixed-methods evaluation approach that integrates both quantitative and qualitative techniques. These methods assess various facets of student learning, including engagement, interactions, perceptions of learning, motivational beliefs, creative self-efficacy, and perceptions of teamwork to measure the impact of PL interventions [64], [44]. Behavioral and performance-based measurement methods assess learner performance using a variety of approaches, including tests, monitoring of engagement levels, tracking of progress, and interactions with IoT-enabled devices [13], [14]. Additionally, learner profiling and cognitive assessment tools include questionnaires that capture learner preferences, interests, domain-specific knowledge, and the Recall Serial Information test used to measure learner memory capacity [11].

Various **Technological Tools** are exploited to measure components of POL. Main tools include secure access and authentication mechanisms, such as the Zero Knowledge Proof system to ensure personalized and safe access to educational resources [61]. In addition, Human-Computer Interaction is evaluated through authoring system for creating PL scenarios, particularly in the context of using robots and IoT-based toys for language learning [93]. Lastly, a service-based framework links PLEs with LMS [55], while federated learning metrics and models like MAML and REPTILE ensure user privacy [87].

Educational Theories and Frameworks Methods employ standardized questionnaires such as the VAK (Visual, Auditory, Kinesthetic) Theory [74], Bloom's Four Agents of Change [12], and the National Survey of Student Engagement [91]. These tools are used to assess student learning styles and to analyze individuals, processes, data, and devices. This comprehensive evaluation supports the creation of PL experiences within the context of IoT.

Learner Feedback and Reflection Methods are essential for understanding and enhancing the PL experience by collecting data on student perceptions, attitudes, self-efficacy, and emotional states. These tools include the Perceptions of Academic Achievement Questionnaire [8], which measures student attention, motivation, attitudes, and perceptions of academic achievement in relation to physical classroom variables. The Foreign Language Virtual Classroom Anxiety Scale (FLVCAS) [38] is used to quantitatively assess anxiety levels in virtual language classrooms. Additionally, Quality of Experience questionnaire [57] evaluates PL components, while another questionnaire measure the impact of PL supports, learner self-evaluation perceptions, and their self-perceived progress [89].

A variety of tools and methods are utilized to measure POL components. Context-Aware LA Tools, such as IoT edge devices and sensors, adapt learning environments based on learner physical context and environmental conditions. Affective states are measured using biometric tools such as HRV, EEG, and emotion detection systems, alongside questionnaires that capture learner preferences and emotional responses. Educational theories and feedback tools further enhance PL experiences by enabling detailed learner profiling and improving system effectiveness.

Key Findings:

- Context-Aware LA Tools including IoT edge devices equipped with sensors, are frequently used to adapt and measure POL components by tracking the learner physical context and environmental conditions in real-time, providing data-driven personalization;
- Biometric Tools, such as Heart Rate Variability, Electroencephalography, and facial emotion detection systems, are commonly used to monitor learners' emotional and physiological states, improving the responsiveness of POL environments to individual needs.

4.2.5 What are the POL algorithms that may be added or enhanced by using IoT?

There are various approaches and algorithms used when integrating IoT into POL systems to deliver PL services. The most common approach, identified from selected primary studies as shown in Table 10, involves **combining multiple ML algorithms**.

Table 10. Algorithms used in POL Systems

POL Algorithms	Number of Studies	Percentage
Combination of Multiple Algorithms	18	28%
Context-Aware Learning Approaches	10	15%
Affective Computing Algorithms	9	14%
Educational Theories and Frameworks	8	12%
Learning Analytics	6	9%
Natural Language Processing	5	8%
Classification Algorithms	2	3%
Deep Learning	2	3%
Reinforcement Learning	2	3%
Rule-Based Learning Algorithms	2	3%
Cryptographic Techniques	1	2%

Classification algorithms, including C4.5, the J48 algorithm, and instance-based classifiers, combined with rule-based methods such as OneR, PART, are employed to classify and analyze student engagement levels. K-means clustering, alongside collaborative filtering techniques and genetic algorithms are utilized to personalize learning experiences by leveraging learner data, behavioral patterns, and individual preferences [72], [16], [37]. The combination of Variable Length Genetic Algorithm, Collaborative Filtering Methods, Ant Colony Optimization and Particle Swarm Optimization are used to generate and recommend PL path [26], [60]. Furthermore, collaborative and content-based filtering algorithms are used to provide PL materials [11]. ML models, such as Dynamic Key-Value Memory Networks and their augmented versions [84], trace student knowledge states and predict performance. ASM and behavioral analysis algorithms measure student engagement during video lectures by analyzing facial and eye movements, contributing to personalized engagement metrics [31], meanwhile RL, Rule Based Learning, and Deep Spatio-Temporal Convolutional LSTMs model dynamically adjust learning content based on cognitive states inferred from EEG data [82]. Furthermore, multi-armed bandit algorithms and decision trees [70] recommend the most effective support to students.

Context-Aware Learning Approaches employ an algorithmic approach in which data from sensors and IoT devices is processed to deliver personalized educational content based on real-time environmental data and user input. Adaptive sensor-based learning algorithms adjust m-learning content based on both predefined learning style models and real-time inputs from physical sensors [83], [96]. Context-aware user profiling focuses on creating personalized user profiles by incorporating context factors to recommend relevant learning objects [97], [48]. IoT technology is also employed to adapt learning experiences according to physical classroom conditions [8]. Furthermore, PhET simulation laboratories, combined with Inquiry-Based Learning and the Technological Pedagogical Content Knowledge framework, are used as interventions in virtual laboratories [65].

In **Affective Computing Algorithms**, emotional and physiological data are utilized through techniques such as facial landmark detection, eye-gaze tracking, and heartbeat monitoring, combined with data aggregation methods to tailor the learning experience to the learner emotional and physiological state [77], [40], [80]. The MBTI model is used to assess learner personality, while the OCC model is applied for emotion modeling, and

the Desirability Level Prediction model helps predict student desirability levels [90]. Affective learning assessments focus on measuring emotional components, such as foreign language anxiety, using tools such as FLVCAS [38]. In addition, PPG signal analysis, LA, and Cognitive Load Measurement, assess emotional engagement in educational metaverse environments [102]. The Wilcoxon signed-rank test is employed as a statistical tool to analyze differences in engagement levels based on physiological metrics [63].

The **Educational Theories and Frameworks** applied in POL emphasize pedagogical strategies to enhance PL through conceptual approaches rather than specific algorithms. These include Social Constructivism and Flow Theory for m-learning personalization, fostering interaction and engagement [64], and Blended Learning, which integrates m-learning to offer self-directed learning at learner pace [64], [66]. Additionally, Instructional Scaffolding and Project-Based Learning help learners grasp complex concepts in Ubiquitous Computing environments [53]. Problem-Based Learning in Cloud Learning Environments further tailors learning experiences by leveraging cloud technologies for flexible and collaborative learning [69]. Finally, Arts-Based Learning strategies, such as Poetweet and Reflective Mosaic, enhance interaction and social presence in m-learning environments [66].

Learning Analytics plays a crucial role in POL by utilizing various algorithms and models to improve educational outcomes and adapt learning processes. An IoT-based system categorizes students based on their learning styles using the VAK theory and fingerprint analysis, assigning personalized educational content tailored to these styles [74]. Furthermore, LA and educational data mining are employed to create pervasive and adaptive learning ecosystems that personalize learning experiences [12], [7]. In addition, ML-enhanced LA integrates models such as Random Forest for activity recognition and predictive analytics to enhance personalized and adaptive learning processes [23].

Natural Language Processing (NLP) algorithms are commonly utilized in the development of ECs within POL systems. Bidirectional Encoder Representations from Transformers model is used to build proactive chatbot framework [6]. Additionally, the integration of 3D virtual worlds, such as OpenSimulator, enhances immersion and presence in language learning environments. These virtual environments utilize tools like chatbots and ”time machines” to create engaging, context-rich learning experiences [88]. Furthermore, Latent Dirichlet Allocation is utilized to process user queries and map learning object metadata into a structured ontology, contributing to a more PL environment [24]. Lastly, NLP algorithms [79] process feedback from instructors and students to optimize the educational framework by analyzing course outcomes.

Classification Algorithms such as the Support Vector Machine algorithm is used for learner modeling by analyzing learner traces in Cloud Computing environment [32]. Additionally, decision tree methods are used to create classifiers that predict user emotional states and activity levels based on HRV, to provide PL experiences based on learner physiological responses [18].

Deep Learning algorithms are used to provide POL by leveraging various neural network models and federated learning techniques. For instance, a deep GNN-based Social Recommendation framework encodes user and item feature spaces into graph networks for personalized recommendations in IoT scenarios. This framework uses matrix factorization in conjunction with encoded graph spaces to infer missing rating values, which is crucial for delivering PL experiences [34]. Similarly, Convolutional Neural Networks

combined with attention networks are employed to enhance the personalization of content based on learner interactions [41].

Reinforcement Learning (RL) frameworks, including hierarchical RL, utilize a Governor RL agent based on a Markov Decision Process to adjust parameters in IoT applications by addressing human variability [28]. Alongside RL, data mining techniques are employed to analyze student learning behaviors and preferences to recommend learning strategies and content [86].

In **Rule-Based Learning Algorithms** RecRules combines content-based and collaborative filtering using semantic reasoning to personalize recommendations [25], while ECs use rule-based, retrieval-based, and generative models for adaptive content delivery and interaction design [44]. In addition, **Cryptographic Techniques** like Hadamard Matrix-based Zero Knowledge Proof System implements authentication secure access roles, which is critical to protect PLEs [61].

These ML algorithms collectively contribute to the personalization of online learning by accurately classifying learners, generating adaptive learning paths, managing human variability, and providing tailored recommendations and interactions. Each algorithm addresses specific aspects of learner behavior and preferences, creating a more adaptive and effective learning environment.

Key Findings:

- The combination of traditional ML algorithms, such as C4.5, J48, and Genetic Algorithms, with IoT-generated data demonstrates the ability to create individualized learning experiences based on learner engagement, preferences, and environmental context;
- ML algorithms like Support Vector Machine, K-Means Clustering, and graph neural networks are employed to analyze learner profiles, classify data, and offer personalized recommendations;
- The reliance on predefined rules and classification parameters can limit the flexibility needed for truly dynamic and real-time personalization;
- The combination of rule-based algorithms with clustering techniques like K-Means can be seen as an effective strategy for handling large-scale data.

4.2.6 What are the IoT tools that may be used in POL?

The integration of IoT tools in POL systems utilizes a variety of protocols, devices, and sensors, Table 11, to enhance the learning experience.

Software Based IoT Tools typically include Chatbots, Virtual Immersive Learning Tools, and Intelligent Agent-based Tools. These tools integrate a range of software technologies to enhance learning personalization and optimize educational processes. Examples of software-based tools include Google Dialogflow, combined with Google Sheets, Python, and JavaScript scripts automates data manipulation processes [73]. Additionally, Textit, a no-code platform for chatbot development, is employed to design and deploy chatbots through messaging platforms such as Telegram [44]. Furthermore, conversational workflows are managed using tools such as Context Dimension Trees and Petri Nets, to develop highly interactive and adaptive chatbot systems [24]. Virtual Immersive Learning Tools, such as PhET Simulations, are utilized in virtual chemistry labs to facilitate

Table 11. IoT tools integrated into POL Systems

IoT Tools	Number of Studies	Percentage
Software Tools	23	35%
Smart and Wearable Devices	17	26%
IoT Tools and Devices	10	15%
Computing and Middle-ware Tools	6	9%
Environmental Sensors	5	8%
Specialized Tools	4	6%

conceptual understanding [65]. In addition, CloudSim, an IoT-based simulator, supports learner modeling in cloud environments, offering flexibility in the delivery of educational content [32].

Furthermore, **Tools** such as Microsoft Cognitive Services-API1, Google Cloud Vision-API2, CLMtrackr3, ADOxxR are utilized for recognizing emotions [27], meanwhile a video camera, Visual C Programming Language and Computer Vision Library are used to measure engagement levels by analyzing video data [31]. Lastly, Audacity Software, JavaScript Templates, LEDs, Sensors, Actuators, and Educational Micro-controllers are used as IoT Educational Tools for creating Ubiquitous Computing, Mobile Computing, and IoT learning scenarios [53].

IoT-enabled Personal Smart Devices based platforms focus on utilizing mobile devices both as IoT tools and learning tools. These devices, embedded with sensors such as accelerometers, proximity sensors, ambient light sensors, and GPS, are used to create ubiquitous learning environments [60], to build context-aware mechanisms [96], and to optimize content delivery based on the learner environment and device capabilities [48]. Additionally, these devices play a significant role in the development of m-learning frameworks [64], [83], [66].

Wearable devices have become integral components of modern IoT-based personalized e-learning environments. These devices include the Mio® Alpha heart rate sport watch, used for emotion prediction [18], and devices such as Google Glass, Oculus Rift, Apple Watch, and Android Wear to provide seamless and PL experience [56]. Other examples include the MKB0805 Dynamic PPG Heart Rate and Blood Pressure Module, as well as Oculus Quest 2 Glasses used to assess learner emotional engagement and cognitive levels [102]. Additionally, wearable devices equipped with NFC readers, often worn on the wrist, are used to track learner progress and assess motivation [16]. Smart Glasses, featuring both outward- and inward-facing cameras, accurately determine visual focal points, enabling the tracking of learner attention and interests [85]. Moreover, Empatica™ E4 device is used measure EDA, PPG, and ST to capture physiological responses to infer engagement levels [63].

IoT data collection and networking tools in personalized e-learning environments encompass a variety of innovative technologies and devices designed together to process and analyze real-world data. Key communication technologies such as Zigbee, Bluetooth, Wireless Sensor Networks, Cloud Computing services, and Internet facilitate connectivity between environmental sensors (e.g., sensor nodes, RFID, GPS) and edge computing tools [61], [7]. Data sources include real-world IoT datasets like Epinions, Yelp, and Flixster [34]. In addition, tools such as Raspberry Pi, RFID, Smart Boards, Biometric Scanners,

Electronic Whiteboard Screens, and Bluetooth Beacons facilitate comprehensive data collection [77], [12].

IoT Prototyping and Development Devices include tools that facilitate the creation and testing of IoT-based e-learning environments. Devices like Raspberry Pi with different sensors, Arduino Uno Board, and Mobile Devices combined with platforms such as Node-RED, GrovePi sensors, and IoT Lab Kits form the basis for IoT-based e-learning environments prototyping [41], [62], [74], [57].

Environmental sensors and tools play a crucial role in enhancing POL by creating context-aware ALEs that respond to physical conditions. An IoT-based system integrates temperature, humidity, air quality, light, and sound sensors to collect environmental data in order to enhance personalization of learning environments [8]. Another application uses GPS to capture location context and deliver personalized and location-based learning content [97]. Furthermore, the dataset titled "Sensor Signal Dataset for Exploring Context Recognition of Mobile Devices" which includes environmental sensors data is utilized to demonstrate the use of mobile device sensors as critical IoT tools for context-aware learning environments [37].

Middle-ware Tools contribute in developing PLEs by integrating IoT with Cloud Technologies [91] LMS such as Moodle [55]. Additionally, middle-ware facilitates data collection and interaction within the educational framework [51]. Edge computing tools facilitate the deployment of Deep Spatio-Temporal Convolutional LSTM models on edge-based prototypes for processing EEG signals to recognize human emotions [82]. Additionally, edge devices such as black globe thermometers and skin temperature monitors are used to assess human thermal comfort [28].

The integration of IoT tools into POL systems enhances educational experiences through a variety of protocols, devices, and sensors. Software-based tools such as chatbots, virtual immersive learning tools, and intelligent agents optimize personalization by using different platforms. In addition, wearable devices such as smartwatches and smart glasses monitor physiological responses to assess learner engagement. Environmental sensors and edge computing tools further personalize learning by adapting to physical conditions. Lastly, middle-ware tools seamlessly integrate IoT with cloud technologies and LMS to optimize data collection and interaction within the learning framework.

Key Findings:

- IoT Software Tools like chatbots, immersive virtual tools, and intelligent agents enhance POL by automating processes and providing personalized interactions;
- Wearable and environmental devices along with sensors for temperature, light, and sound, monitor learner physiological and environmental data to personalize learning experiences;
- Tools like Raspberry Pi, Arduino, and cloud technologies facilitate real-time data collection and creation of adaptive learning environments.

4.2.7 Which research approaches are used in academic literature related to the integration of IoT into POL and what contribution has been achieved?

Table 12 presents the research contribution types and their distribution related to the integration of IoT in POL. Among the various types of contributions, papers proposing

frameworks or methods are the most prevalent, accounting for 68% of selected papers. These papers introduce innovative IoT and AI-based systems that use wearable sensors and biosensors, to personalize and enhance educational experiences. Extracted main frameworks include IoT for Education 4.0 [23], ERUDITE [82], and APPLE [91], which utilize tools like smartwatches and EEG signals to track emotions. Additionally, FaiR-IoT framework addressed human variability in order to provide adaptive, sustainable, and learner-centered education [28].

Table 12. Number of publication by contribution type

Contribution Type	Number of Studies	Percentage
Framework/methods	44	68%
Tool	11	17%
Model	7	11%
Guidelines	2	3%
Lesson learned	1	2%

Another collection of research papers introduces various educational **tools** to enhance learning experiences. These tools include an iOS application for personalized course content based on learning styles, WIoTED system for improving student engagement through wearable IoT devices, and an authoring system for robots and IoT-based toys to support EFL teaching [83], [96], [93]. Highlighted other tools include an intelligent music selection system based on HRV, an IoT-based system for supervising online English teaching through intelligent face perception, and a personalized NIE content recommendation system [18], [41], [77].

The collection of papers under the **Model** contribution type introduces various innovative models to enhance educational experiences. Extracted main models include the Customized X-Learning Environment model, which integrates multiple learning elements and technologies to create a student-centered virtual education environment [55]. PrivRec and DP-PrivRec, privacy-preserving federated learning-based recommender systems, models are also highlighted [87]. Additionally, a ML-based context fusion model infer high-level context information from low-level multi-domain IoT data [37]. Moreover, Decision Tree Thompson Sampling is introduced as a novel model for PL within the Automatic Personalized Learning Service framework [70].

Table 13 presents the results of the study based on the research types of the selected papers. The majority of primary studies (66%) focus on **proposing solutions**, most of which aim to develop and test new approaches that have not yet been fully evaluated in real-world practical scenarios. These solution-oriented papers introduce various frameworks and systems. Highlighted main solutions include an m-learning framework designed for resource-constrained environments [64], a Cloud of Things-based online education framework [61], a Personalized Context-Aware Recommendation learning system [96], and an IoT-based educational systems [93]. Other solutions include an IoT-based personalized Newspaper in Education content recommendation system [41], an IoT-based interaction framework combined with an attention-scoring algorithm [31], and an adaptive learning models utilizing ML to track student knowledge states [84]. Experimental results and

empirical studies validate the effectiveness of these solutions, demonstrating improved learning outcomes, and increased user satisfaction across diverse educational contexts.

Table 13. Number of publication by research type

Research Type	Number of Studies	Percentage
Solution research	43	66%
Evaluation research	18	28%
Validation research	3	5%
Experience paper	1	2%

Evaluation research is the second most common research type implemented in IoT-based POL, comprising 23% of selected studies, and focuses on assessing the effectiveness and performance of various educational technologies and interventions in real-life settings. These studies span over multiple aspects, including context-aware ubiquitous learning systems [76], adaptive e-learning systems based on personality and emotion [64], and a PL through an iOS application that improves student comprehension [61]. IoT technology impact on student attention and motivation through physical variables [8], the influence of virtual classrooms on foreign language anxiety [38], and vocational student self-efficacy and attitudes toward m-learning [90] have also been evaluated. Lastly, the Personalized Learning Space in the Educational Metaverse [83] using physiological and cognitive metrics, and IoT-enabled smart labs to assess their impact on academic performance [7] are also highlighted.

The rest of papers falls into different categories emphasysing different experiences and validations of IoT-based POL systems [62], [82], [94].

Key Findings:

- Research contributions of integrating IoT into POL primarily focus on developing different frameworks and methods;
- Solution-based studies dominate the integration of IoT into POL, though often not yet tested extensively in real-world environments;
- Evaluation research studies assesses real-life effectiveness of IoT-based POL systems across various educational technologies.

5. Conclusions and Future Works

This paper conducted a systematic mapping study exploring the integration of IoT technologies in POL, and identified the current landscape, limitations, and potential research gaps. We systematically considered various dimensions of IoT integration within existing POL systems, providing a comprehensive understanding of techniques, models, components, tools, and algorithms involved.

Our analysis reveals that 35% of the reviewed studies integrate IoT to create ubiquitous smart learning and IoT based e-learning systems, however as few as 3% of studies

used IoT to integrate and control environmental parameters into POL systems. Furthermore, 40% of these studies highlight learner profile as a primary factor in personalizing educational experiences where IoT is expressly used to extract socio-psychological, and biological context. Additionally, 42% of the studies utilized LA and a combination of multiple algorithms to process data and delivered tailored online learning services. 66% of selected studies focused on solution research, while 63% of the latter selection were solution proposals, frameworks or methods.

Use of IoT in education is shifting the structure of educational system by bringing changes to educational institutions infrastructure, and involving innovative teaching methodologies and learning practices. Typical responsibilities of students and teachers are fundamentally changed in IoT-based e-learning settings and there is an imminent need for innovative teaching/learning methods to enable integration of IoT in education.

The use IoT in POL bridges the gap between the physical and virtual worlds and significantly reshapes ALEs. The integration of IoT in personalizing environmental factors, intelligent tutoring systems, and reinforcement learning algorithms remain underexplored, additionally, there is a lack of utilizing IoT to seamlessly integrate the adaptation of physical and virtual learning environments.

Additional analysis of psychological and behavioral patterns through IoT data holds great promise for the development of decision support systems that cater to the unique needs of individual learners. In addition, challenges of privacy, security, and personalization of external factors, proper integration of adaptive learning platform modules, and heterogeneity of data need to be handled in IoT-based POL systems.

Based on our findings we conclude that integration of IoT into existing POL systems increases personalization and enhances academic outcomes. Furthermore, we also conclude that there is a need for new pedagogical approaches, personalization models, and standardizations to accommodate IoT innovations in Education. Lastly, our study shows that there is a lack of longitudinal studies examining the long-term impacts of IoT-enhanced learning environments on educational outcomes.

These findings offer valuable insights for educators, policymakers, and technology developers to enhance the design and implementation of PLEs.

Our future study will propose a solution framework that seamlessly integrates biological, psychological, and environmental components into existing POL systems. In addition, the impact of these components on educational outcomes will be assessed.

References

1. Abbasy, M.B., Quesada, E.V.: Computers & education understanding socio-technological challenges of smart classrooms using a systematic review. International Journal of Information and Education Technology 7(12), 914–920 (2017)
2. Adams, B.S., Cummins, M., Davis, A., Freeman, A. Hall, G.C., Ananthanarayanan, V.: Nmc horizon report: 2017 higher education edition. The New Media Consortium, Austin, Texas (2017), [Online]. Available: https://library.educause.edu/-/media/files/library/2017/2/2017horizonreporthe.pdf
3. Adi, E., Anwar, A., Baig, Z., Zeadally, S.: Machine learning and data analytics for the iot. Neural Computing and Applications 32(20), 16205–16233 (2020)
4. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys ‘and Tutorials 17(4), 2347–2376 (2015)

5. Alfoudari, A.M., Durugbo, C.M., Aldhmour, F.M.: Predictable influence of iot (internet of things) in the higher education. Computers & Education Journal 173(12) (2021)
6. Almada, A., Yu, Q., Patel, P.: Proactive chatbot framework based on the ps2clh model: An ai-deep learning chatbot assistant for students. In: Intelligent Systems and Applications. pp. 751–770. Springer International Publishing, Cham (2023)
7. Asad, M., Naz, A., Shaikh, A., Alrizq, M., Akram, M., Alghamdi, A.: Investigating the impact of iot-based smart laboratories on students' academic performance in higher education. Universal Access in the Information Society 23, 1135–1149 (2024)
8. Aydin, A., Göktaş, Y.: Examining the effects of physical variables in classrooms on students' attention via the internet of things. Participatory Educational Research (PER) 10(1), 160–177 (2023)
9. Bagheri, M., Movahed, S.H.: The effect of the internet of things (iot) on education business model. In: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). pp. 453–441. IEEE (2016)
10. Bajracharya, B., Gondi, V., Hua, D.: Iot education using learning kits of iot devices. Information Systems Education Journal 19(6), 40–44 (2021)
11. Benhamdi, S., Babouri, A., Chiky, R.: Personalized recommender system for e-learning environment. Education and Information Technologies 22, 1455–1477 (2017)
12. Betts, A., Thai, K.P., Gunderia, S., Hidalgo, P., Rothschild, M., Hughes, D.: An ambient and pervasive personalized learning ecosystem: "smart learning" in the age of the internet of things. In: Adaptive Instructional Systems: Second International Conference, AIS 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings. pp. 15–33. Springer-Verlag, Berlin, Heidelberg (2020)
13. Bondaryk, L.G., Hsi, S., Van Doren, S.: Probeware for the modern era: Iot dataflow system design for secondary classrooms. IEEE Transactions on Learning Technologies 14(2), 226–237 (2021)
14. Bourekkache, S., K., O.: Mobile and adaptive learning application for english language learning. International Journal of Information and Communication Technology Education (IJICTE) 16(2), 36–46 (2020)
15. Butt, R., Siddiqui, H.and Soomro, R., Asad, M.: Integration of industrial revolution 4.0 and iots in academia: a state-of-the-art review on the concept of education 4.0 in pakistan. Interactive Technology and Smart Education 17(4), 337–354 (2020)
16. Camacho, V.L., de la Guía, E., Olivares, T., Flores, M.J., Orozco-Barbosa, L.: Data capture and multimodal learning analytics focused on engagement with a new wearable iot approach. IEEE Transactions on Learning Technologies 13(4), 704–717 (2020)
17. Chen, X., Zou, D., Xie, H., Wang, F.L.: Past, present, and future of smart learning: a topic-based bibliometric analysis. International Journal of Educational Technology in Higher Education 18(2), 2365–9440 (2021)
18. Chiu, M.C., Ko, L.W.: Develop a personalized intelligent music selection system based on heart rate variability and machine learning. Multimedia Tools and Applications 76(14), 15607–15639 (2017)
19. Chu, S.L., Garcia, B.M., Rani, N.: Research on wearable technologies for learning: A systematic review. Frontiers in Education 8 (2023)
20. Chweya, R., Ibrahim, O.: Internet of things (iot) implementation in learning institutions: A systematic literature review. SCIENCE & TECHNOLOGY 29(1), 471–517 (2021)
21. Cico, O., Jaccheri, L., Nguyen-Duc, A., Zhang, H.: Exploring the intersection between software industry and software engineering education - a systematic mapping of software engineering trends. Journal of Systems and Software 172 (2021)
22. Ciolacu, M.I., Binder, L., Popp, H.: Enabling iot in education 4.0 with biosensors from wearables and artificial intelligence. In: 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME). pp. 17–24. IEEE (2019)

23. Ciolacu, M.I., Binder, L., Svasta, P., Tache, I., Stoichescu, D.: Education 4.0 – jump to innovation with iot in higher education. In: 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME). pp. 135–141. IEEE (2019)
24. Clarizia, F., Colace, F. andLombardi, M., Pascale, F., Santaniello, D.: Chatbot: An education support system for student. In: Cyberspace Safety and Security. pp. 291–302. Springer International Publishing, Cham (2018)
25. Corno, F., De Russis, L., Monge R., A.: Recrules: Recommending if-then rules for end-user development. ACM Transactions on Intelligent Systems and Technology 10(5), 1–27 (2019)
26. Dwivedi, P., Kant, V., Bharadwaj, K.K.: Learning path recommendation based on modified variable length genetic algorithm. Education and Information Technologies 23(2), 819–836 (2018)
27. Elkobaisi, M., Al Machot, F.: Human emotion modeling (hem): an interface for iot systems. Journal of Ambient Intelligence and Humanized Computing 13, 4009–4017 (2022)
28. Elmalaki, S.: Fair-iot: Fairness-aware human-in-the-loop reinforcement learning for harnessing human variability in personalized iot. In: Proceedings of the International Conference on Internet-of-Things Design and Implementation. p. 119–132. Association for Computing Machinery, New York, NY, USA (2021)
29. Embarak, O.H.: Internet of behaviour (iob)-based ai models for personalized smart education systems. Procedia Computer Science 203(C), 103–110 (2022)
30. Ennouamani, S., Mahani, Z.: An overview of adaptive e-learning systems. In: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS). pp. 342–347. IEEE (2017)
31. Farhan, M., Jabbar, S., Aslam, M., Ahmad, A., Iqbal, M., Khan, M., Maria, M.E.A.: A real-time data mining approach for interaction analytics assessment: Iot based student interaction framework. International Journal of Parallel Programming 46(5), 886–903 (2018)
32. Ghallabi, S., Essalmi, F., Jemni, M., Kinshuk: Learner modeling in cloud computing. Education and Information Technologies 25(6), 5581–5599 (2020)
33. Gul, S., Asif, M., Ahmad, S., Yasir, M., Majid, M., Malik, M.S.A.: A survey on role of internet of things in education. IJCSNS International Journal of Computer Science and Network Security 17(5), 159–165 (2017)
34. Guo, Z., Wang, H.: A deep graph neural network-based mechanism for social recommendations. IEEE Transactions on Industrial Informatics 17(4), 2776–2783 (2021)
35. Gómez, J., Huete, J.F., Hoyos, O., Perez, L., Grigori, D.: Interaction system based on internet of things as support for education. Procedia Computer Science 21, 132–139 (2013)
36. Jando, E., Meyliana, Hidayanto, A.N., Prabowo, H., Warnars, H.L.H.S., Sasmoko: Personalized e-learning model: A systematic literature review. In: 2017 International Conference on Information Management and Technology (ICIMTech). pp. 238–243. Morgan Kaufmann, Santiago, Chile (2017)
37. Jia, B., Liu, S., Guan, Y., Li, W., Ren, W.: The fusion model of multidomain context information for the internet of things. Wireless Communications and Mobile Computing 2017(1) (2017)
38. Kaisar, M.T., Chowdhury, S.Y.: Foreign language virtual class room: Anxiety creator or healer? English Language Teaching 13(11) (2020)
39. Kassab, M., DeFranco, J., Laplante, P.: A systematic literature review on internet of things in education: Benefits and challenges. Journal of Computer Assisted Learning 36(2), 115–127 (2017)
40. Kassab, M., Mazzara, M.: Towards designing smart learning environments with iot. In: Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment: Second International Workshop, DEVOPS 2019, Château de Villebrumier, France, May 6–8, 2019, Revised Selected Papers. pp. 152–166. Springer-Verlag, Berlin, Heidelberg (2019)

41. Kim, Y., Jung, S., Ji, S.and Hwang, E., Rho, S.: Iot-based personalized nie content recommendation system. Multimedia Tools and Applications 78(3), 3009–3043 (2019)
42. Kitchenham, B. (ed.): Guidelines for performing Systematic Literature Reviews in software engineering. EBSE Technical Report EBSE-2007-01. Keele, UK, Keele Univ. (2007)
43. Kitchenham, B., Brereton, P.: A systematic review of systematic review process research in software engineering. Information and Software Technology 55(12), 2049–2075 (2013)
44. Kumar, J.: Educational chatbots for project-based learning: investigating learning outcomes for a team-based design course. International Journal of Educational Technology in Higher Education 18(65) (2021)
45. Kusuma, S., Kasi Viswanath, D.: Iot and big data analytics in e-learning: A technological perspective and review. International Journal of Engineering & Technology 7(1.8), 164–167 (2018)
46. Laksmi, I., P., H., E., W.: A systematic review of iot platforms in educational processes. In: Proceedings of the 12th Annual International Conference on Industrial Engineering and Operations Management. Istanbul, Turkey (2022)
47. Littell, J.: Systematic reviews in the social sciences: A review. Evidence & Policy: A Journal of Research, Debate and Practice 2, 535–537 (2006)
48. Louhab, F.E., Bahnasse, A., Talea, M.: Smart enhanced context-aware for flipped mobile learning: Seca-fml. In: EAI International Conference on Technology, Innovation, Entrepreneurship and Education. pp. 95–107. Springer International Publishing, Cham (2019)
49. Ma, J., Li, J., Lv, Z.: English web-based teaching supervision based on intelligent face image perception and processing for iot. Complexity 2021 (2021)
50. Madhavi, V., Sanjeev, W.: An architectural approach of internet of things in e-learning. In: 2015 International Conference on Communications and Signal Processing (ICCSP). pp. 1773–1776 (2015)
51. Mahapatra, S.K., Pattanayak, B.K., Pati, B.: Flip learning: A novel iot-based learning initiative. In: Intelligent and Cloud Computing. pp. 59–67. Springer Singapore, Singapore (2021)
52. Mavaluru, D., Arishi, H.A.: Internet of things (iot) technology for smart e-learning using data mining: Review and challenges. Journal of Advanced Research in Dynamical and Control Systems 10(13), 626–632 (2018)
53. Mavroudi, A., Fragou, O., Goumopoulos, C.: Design concepts, principles and patterns in the curriculum of the new computing education era. Designs for Learning 11(1), 141–153 (2019)
54. Mershad, K., Wakim, P.: A learning management system enhanced with internet of things applications. Journal of Education and Learning 7(3), 23–40 (2018)
55. Mesquita, A., Moreira, F., Péres, P.: Customized x-learning environment. In: ICERI2016 Proceedings. pp. 835–843. IATED, Seville, Spain (2016)
56. Mitra, S., Gupta, S.: Mobile learning under personal cloud with a virtualization framework for outcome based education. Education and Information Technologies 25(3), 2129–2156 (2020)
57. Mohana, M., da Silveira, A.C., Subashini, P., Santos, C.A.S., Ghinea, G.: Technology enhanced mulsemedia learning: Insights of an evaluation. In: Computer-Human Interaction Research and Applications. pp. 24–42. Springer Nature Switzerland, Cham (2023)
58. Moreira, F., Ferreira, M.J., Cardoso, A.: Higher education disruption through iot and big data: A conceptual approach. In: Learning and Collaboration Technologies. Novel Learning Ecosystems. pp. 389–405. Springer International Publishing, Cham (2017)
59. Moreira, F.T., Magalhães, A., Ramos, F.and Vairinhos, M.: The power of the internet of things in education: An overview of current status and potential. In: Citizen, Territory and Technologies: Smart Learning Contexts and Practices. pp. 51–63. Springer International Publishing, Cham (2018)
60. Muhammad, A., Shen, J., Beydoun, G., Xu, D.: Sbar: A framework to support learning path adaptation in mobile learning. In: Frontier Computing. pp. 655–665. Springer Singapore, Singapore (2018)

61. Mukherjee, K.and Maurya, S., Mandal, R.K., Thakur, M., Khatoon, F.: A novel methodology for cloud of things-based secure higher education framework using zero knowledge proof system. International Journal of Web-Based Learning and Teaching Technologies (IJWLTT) 17(4), 1–21 (2022)
62. Mylonas, G., Paganelli, F., Cuffaro, G., Nesi, I., Karantzis, D.: Using gamification and iot-based educational tools towards energy savings - some experiences from two schools in italy and greece. Journal of Ambient Intelligence and Humanized Computing 14(12), 15725–15744 (2023)
63. Ojha, A., Jebelli, H., Sharifironizi, M.: Understanding students' engagement in learning emerging technologies of construction sector: Feasibility of wearable physiological sensing system-based monitoring. In: Proceedings of the Canadian Society of Civil Engineering Annual Conference 2021. pp. 269–281. Springer Nature Singapore, Singapore (2023)
64. Okai-Ugbaje, S., Ardzejewska, K., Imran, A.: A mobile learning framework for higher education in resource constrained environments. Education and Information Technologies 27(8), 11947–11969 (2022)
65. Penn, M., Ramnarain, U.: A comparative analysis of virtual and traditional laboratory chemistry learning. Perspectives in Education 37(4) (2019)
66. Perry, B., Edwards, M.: Innovative arts-based learning approaches adapted for mobile learning. Open Praxis 11(3), 303–310 (2019)
67. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. pp. 68–77. BCS Learning & Development Ltd., Swind, GBR (2008)
68. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology 64, 1–18 (2015)
69. Phunaploy, S., Chatwattana, P., Piriyasurawong, P.: The problem-based learning process with a cloud learning environment to enhance analysis thinking. International Journal of Higher Education 10(6), 45–56 (2021)
70. Prihar, E., Haim, A., Sales, A., Heffernan, N.: Automatic interpretable personalized learning. In: Proceedings of the Ninth ACM Conference on Learning @ Scale. pp. 1–11. Association for Computing Machinery, New York, NY, USA (2022)
71. Rahmani, A.M., Ali, N.R., Hussain, M.M., Malik, T.S., Sadrishojaei, M., Hosseinzadeh, M., Al-Musawi, A.: E-learning development based on internet of things and blockchain technology during covid-19 pandemic. Mathematics 9(24) (2021)
72. Rawat, B., Dwivedi, S.K.: Discovering learners' characteristics through cluster analysis for recommendation of courses in e-learning environment. International Journal of Information and Communication Technology Education (IJICTE) 15(1) (2019)
73. Reyes, R., Garza, D., Garrido, L., De la Cueva, V., Ramirez, J.: Methodology for the implementation of virtual assistants for education using google dialogflow. In: Advances in Soft Computing. pp. 440–451. Springer International Publishing, Cham (2019)
74. Saxena, A., Shinghal, K., Misra, R., Agarwal, A.: Automated enhanced learning system using iot. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). pp. 1–5. IEEE press (2019)
75. Shanmukha, S.Y., Kiran, K.K.: Internet of things and its applications. International Journal of Engineering & Technology 7(2), 422–427 (2018)
76. Shapsough, S., Zualkernan, I.: A generic iot architecture for ubiquitous context-aware assessments. In: Proceedings of the 10th International Conference on Education Technology and Computers. p. 69–73. Association for Computing Machinery, New York, NY, USA (2018)
77. Shrestha, S.K., Furqan, F.: Iot for smart learning/education. In: 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA). pp. 1–7. IEEE (2020)

78. Sneesl, R., Jusoh, Y.Y., Jabar, M.A., Abdullah, S.: Revising technology adoption factors for iot-based smart campuses: A systematic review. Sustainability 14(4840), 23–40 (2022)
79. Soui, M., S.Karthik, A., A.: Intelligent personalized e-learning platform using machine learning algorithms. In: Lokulwar, P., Verma, B., Thillaiarasu, N., Kumar, K., Bartere, M., Singh, D. (eds.) Machine Learning Methods for Engineering Application Development, pp. 110–126. Bentham Science Publishers (2022)
80. Spyrou, E., Vretos, N., Pomazanskyi, A., Asteriadis, S., Leligou, H.C.: Exploiting iot technologies for personalized learning. In: 2018 IEEE Conference on Computational Intelligence and Games (CIG). pp. 1–8. IEEE Press (2018)
81. Svrcek, M., Kompan, M., Bieliková, M.: Towards understandable personalized recommendations: Hybrid explanations. Comput. Sci. Inf. Syst. 16(1), 179–203 (2019)
82. Taherisadr, M., Al Faruque, M.A., Elmalaki, S.: Erudite: Human-in-the-loop iot for an adaptive personalized learning system. IEEE INTERNET OF THINGS JOURNAL 11(8), 14532–14550 (2024)
83. Tortorella, R., Graf, S.: Considering learning styles and context-awareness for mobile adaptive learning. Education and Information Technologies 22(1), 297–315 (2017)
84. Trifa, A., Hedhili, A., Chaari, W.: Knowledge tracing with an intelligent agent, in an e-learning platform. Education and Information Technologies 24, 711–741 (2019)
85. Tsai, Y.T., Yu, S.J., Chen, X.Y., Chen, O.T.C., Sun, J.C.Y., Huang, C.C.: Personalized mobile learning system via smart glasses. In: IoT as a Service. pp. 136–144. Springer International Publishing, Cham (2018)
86. Wang, M., Lv, Z.: Construction of personalized learning and knowledge system of chemistry specialty via the internet of things and clustering algorithm. The Journal of Supercomputing 78(8), 10997–11014 (2022)
87. Wang, Q., Yin, H., Chen, T., Yu, J., Zhou, A., Zhang, X.: Fast-adapting and privacy-preserving federated recommender system. The VLDB Journal 31(5), 877–896 (2021)
88. Wang, Y.F., Petrina, S., Feng, F.: Village—virtual immersive language learning and gaming environment: Immersion and presence. British Journal of Educational Technology 48(2), 431–450 (2017)
89. Wang, Y.H.: Using augmented reality to support a software editing course for college students. Journal of Computer Assisted Learning 33(5), 532–546 (2017)
90. WFatahi, S.: An experimental study on an adaptive e-learning environment based on learner's personality and emotion. Education and Information Technologies 24(4), 2225–2241 (2019)
91. Whalley, B., France, D., Park, J., Mauchline, A., Welsh, K.: Developing active personal learning environments on smart mobile devices. In: Proceedings of the Future Technologies Conference (FTC) 2019. pp. 871–889. Springer International Publishing, Cham (2020)
92. Woodside, J.M., Amiri, S., Sause, W.: The internet of things in e-learning. In: Proceedings of eLearn: World Conference on EdTech 2015. pp. 1207–1210. Association for the Advancement of Computing in Education (AACE), Santiago, Chile (2015)
93. Ya-Wen, C., Yuping, W., Yu-Fen, Y., Zih-Kwan, Y., Nian-Shing, C.: Designing an authoring system of robots and iot-based toys for efl teaching and learning. Computer Assisted Language Learning 34(1-2), 6–34 (2021)
94. Yakoubovsky, R., Sarian, V.: Iot in effective distance learning process. In: 2021 1st International Conference on Technology Enhanced Learning in Higher Education (TELE). pp. 311–314. IEEE (2021)
95. Yan, Z., Du, H., Zhang, L., Zhao, J.: Personalization exercise recommendation framework based on knowledge. Comput. Sci. Inf. Syst. 20(2), 857–878 (2023)
96. Yao, C.B.: Constructing a user-friendly and smart ubiquitous personalized learning environment by using a context-aware mechanism. IEEE Transactions on Learning Technologies 10(1), 104–114 (2017)
97. Yau, J.Y.K., Hristova, Z.: Evaluation of an extendable context-aware "learning java" app with personalized user profiling. Technology, Knowledge and Learning 23(2), 315–330 (2018)

98. Yu, J., Zhao, L., Yin, S., Ivanovic, M.: News recommendation model based on encoder graph neural network and bat optimization in online social multimedia art education. Comput. Sci. Inf. Syst. 21(3), 989–1012 (2024)
99. Z., D., X., H.: Personalized word-learning based on technique feature analysis and learning analytics. Journal of Educational Technology & Society 21(2), 233–244 (2018)
100. Zhang, M., Li, X.: Design of smart classroom system based on internet of things technology and smart classroom. Mobile Information Systems (2021)
101. Zhang, M., Saeed, R., Saeed, S., Stankovski, S., Zhang, X.: Wearable technology and applications: A systematic review. Journal of Mechatronics, Automation and Identification Technology 5(3), 5–16 (2020)
102. Zhao, Z., Zhao, B., Ji, Z., Liang, Z.: On the personalized learning space in educational metaverse based on heart rate signal. International Journal of Information and Communication Technology Education 18(2), 1–12 (2018)
103. Čolaković, A., Hadžialić, M.: Internet of things (iot): A review of enabling technologies, challenges, and open research issues. Computer Networks 144, 17–39 (2018)

Edlir Spaho is a seasoned computer engineer with extensive experience in education management and IT services. Based in Tirana, Albania, he serves as an assistant lecturer at "Bedër" University College and "Epoka" University, where he teaches a range of courses, including C++ Programming, Artificial Intelligence, Data Mining, and Internet of Things (IoT) Fundamentals. With a Master of Science in Computer Engineering from "Epoka" University and ongoing PhD studies focused on developing a personalized e-learning algorithm leveraging IoT, Edlir combines technical expertise in programming and AI with a deep commitment to advancing educational technology. His professional background also includes roles in IT management, curriculum development, and authoring educational materials. He has received several awards, including recognition at the Balkan Olympiad of Informatics. Beyond teaching, he is passionate about research and innovation in personalized and adaptive learning systems.

Betim Çiço is a full professor in the Computer Engineering Department at Epoka University. His current research interests include high-performance computing, digital design, methodologies and tools, e-learning, and FPGA applications in nuclear electronics, among others. He received his Diploma Engineer degree in Electrical and Electronic Engineering from the Polytechnic University of Tirana, Albania (PUT), and his PhD in Nuclear Electronics from the Institute of Nuclear Physics (INP, Research Institute), Tirana, Albania, in 1983. He served as a scientific researcher at the INP from 1972 to 1998 (Head of the Nuclear Electronic Department) and as a part-time professor at PUT. Since 1998, he has been a full professor at PUT (until 2012, Head of Computer Engineering Department), South East European University in Tetovo, Macedonia (Dean of Faculty of Contemporary Sciences and Technology, 2 years), and at Epoka University in Tirana since 2016. He participated as a partner in many European Scientific Activities, like FP5, FP6, FP7, and Horizon 2020. He is also a member of many program committees and stirring committees (ACM, IEEE, Springer Fergal) and on the editorial board of journals.

Received: December 26, 2024; Accepted: February 16, 2025.

Computer Science and Information Systems 23(1):657–685 https://doi.org/10.2298/CSIS250405084D

A Knowledge Graph based Approach for Credit Card Fraud Detection*

George Konstantinos Dimou and Georgia Koloniari

University of Macedonia, Thessaloniki, Greece
{dai18069,gkoloniari}@uom.edu.gr

Abstract. Credit Card Fraud Detection remains an evolving and difficult challenge due to its complexity, class imbalance, and scale of transactional data. In this work, we propose a novel graph-based approach that constructs a Knowledge Graph (KG) from transactional data to model relationships between different entities. We calculate a variety of centrality measures, both in an unweighted and in a weighted KG, where edge weights represent transaction-specific features like amount, in order to capture structural importance. These centrality measures are then used to enrich the feature space for multiple Machine Learning (ML) models. Our experimental evaluation assesses the performance of the proposed approach with respect to both accuracy and efficiency. Various experiments are conducted, comparing multiple centrality measures, feature combinations and resampling, showcasing how the addition of the centrality measures as classification features significantly improves the performance of our classification models compared to relying only on the original transactional attributes.

Keywords: Knowledge Graphs, Centrality Measures, Machine Learning, Weighted Graph, Credit Card Fraud Detection.

1. Introduction

Fraud, defined as deliberate deception with the intention of obtaining personal or financial gain, poses a diachronic problem which nowadays emerges as more prevalent than ever. Focusing specifically on fraud for financial gain, its effects amount to substantial losses both to individuals and businesses. With the widespread of card-based payments *Credit Card Fraud* becomes a particularly prominent type of fraud, and comprises the most common type of identity theft, while the attempts at fraudulent credit card transactions rise by 46% every year with a projected $43 billion of losses by 2026 [34]. Credit card fraud can be distinguished between card-present and card-not-present fraud. The latter type accounts for 65% of all credit card losses [34] due to in part the ever-growing number of online transactions. While technology offers avenues for improved preventive measures (such as secure embedded microprocessor chips) these rely mainly on mistakes on the fraudsters part, and most often new means to circumvent these precautions keep emerging.

From the above, it becomes clear, that preventive measures are not sufficient, and of particular importance becomes the challenge of timely detection of a fraud attempt so as to minimize any potential losses. Therefore, *Fraud Detection*, defined as identifying

* This paper is an extended version of the work presented in [13].

activities suspicious of being fraudulent, is one of the fundamentals of fraud risk management along with prevention and investigation. *Credit Card Fraud Detection* (CCFD) is a subcategory of fraud detection that deals with fraudulent activities involving the use of credit cards, i.e., targeting credit card fraud, particularly.

Towards the development of effective fraud detection approaches both data analysis relying on statistics and AI-based methods, and in particular, Machine Learning (ML) methods have been widely deployed [2,21,31,39,41]. All such methods rely on feature engineering, which is not a trivial task, requiring to select appropriate features that will enable accurate and effective detection of suspicious transactions. However, though the advances in AI and ML offer significant promise for further improvements, one dimension that must not be ignored is the correlations that fraudulent transactions often have. Traditional approaches mostly ignore such hidden relationships, considering each transaction in isolation, and thus disregarding important information that can be garnered by identifying and exploiting the interconnections between fraudsters and fraudulent transactions.

To address this issue, in our work, we propose utilizing *Knowledge Graphs* (KGs) for modeling transactional data and exploiting KG-based features for CCFD. Knowledge Graphs are used to represent entities and their relationships and are particularly appropriate when we are interested in studying the inter-workings of any complex system such as a transaction network. Instead of only studying each transaction separately, modeling the data through a KG, enables us to embed it in a network that facilitates the revelation of hidden patterns and relationships that otherwise would be very difficult, if not impossible, to detect. The goal of this work is to design an appropriate KG for modeling transactional data that can be utilized to extract enriched features that, in turn, can be leveraged in ML approaches to achieve accurate fraud detection.

In particular, we explore two alternative KG models, one unweighted and one weighted, that exploits transactional properties to define appropriate edge weights. Network analysis is then utilized to extract additional network features that can act in a way complimentary to the traditional tabular data that conventional ML methods usually employ. As network features, we consider network *centrality measures*, that is measures that are used to quantify the importance of different nodes in a network. Centrality measures consider different aspects of network structure and thus identify nodes with different roles of importance in the network, acting as central figures, appearing in many network paths and so on. To address efficiency, we implement our KG models in a graph database and make use of its network analysis libraries to extract network features. We model CCFD as a binary classification problem and experimentally study the performance of various ML classifiers and various combinations of centralities with respect to both detection accuracy and efficiency.

The work presented here extends previous work that used KGs for CCFD presented in [13]. In this work, we consider a second weighted KG in addition to an unweighted KG model, and deploy additional centrality measures as well as weighted versions of them that exploit the corresponding weighted KG model. Furthermore, we have conducted a very thorough experimental evaluation comparing different combinations of centrality-based features and traditional features to determine the most successful combinations along with more classifiers. We have also conducted new experiments showcasing the scalability and time efficiency of the proposed approach.

Our approach is presented in the next sections as follows. Section 2 briefly summarizes work related to both CCFD and FD and KG-based approaches. Section 3 introduces our proposed KG data modeling, presenting both the unweighted and weighted KG model. Section 4 defines the CCFD classification problem, explains our ML pipeline, and focuses on the different centrality measures that we study. Section 5 includes our experimental evaluation, and Section 6 offers conclusions and directions for future work.

2. Related Work

Advancements in Artificial Intelligence (AI) and ML, have made them irreplaceable in CCFD. A plethora of studies [10,19,22,24,35] showcase how even the simplest ML models can perform noticeably better than conventional rule-based systems. Most such approaches utilize the tabular transactional data as their learning features, while most recently there are some attempts that explore the use of KGs to improve detection performance.

2.1. Fraud Detection Using Tabular Data Features

A characteristic that limits the performance of AI and ML-based methods is the high class imbalance that FD in general and CCFD as well exhibit. Most transactions are legitimate transactions and only a very small percentage consists of fraudulent transactions, making their classification as such extremely difficult. Therefore, many works particularly target this class imbalance issue by deploying different sampling schemes such as SMOTE [8], towards improving FD and CCFD accuracy. Varmedja et al. [40] employed a pipeline utilizing a feature selection tool, SMOTE and normalization to address the issue. Similarly, Dhankhad [12] used multiple algorithms and ensemble learning in tandem with SMOTE, observing the clear superiority of the stacked models. In another study, Rahman et al. [33] examined the Chinese stock market and financial research databases and utilized boosting algorithms that incorporated random under-sampling and clustering based under-sampling, called RUSBoost and CUSBoost, respectively. Those models clearly outperformed other popular boosting algorithms such as AdaBoost and XGBoost in terms of accuracy, showcasing the need of those specially-engineered algorithms. Even though, resampling can be beneficial and effectively reduce class imbalance, it frequently results in significant computing overhead and possible model bias. Furthermore, there are also instances where using resampling techniques even worsens performance, as also seen later in our work, offering no real value. Thus, alternative approaches are also explored, such as in [15], where instead of sampling, they opted to use multiple classifiers for different user groups and proposed clustering but along with substantial preprocessing increasing overhead as well.

Another direction towards dealing with class imbalance is methods based on fuzzy theory. In their stacking architecture, Hussein et al. [36] combined fuzzy set theory, rough set-based classification and SVM with fuzzy Rough Nearest Neighbor and Sequential Minimal Optimization. Although this approach showcased potential in handling imbalanced datasets, it suffered from significant computational overhead, rendering it impractical for larger datasets, as their evaluation was limited to fewer than 1000 samples. Similarly, Charizanos et al. [7] presented an online fuzzy FD system that used reasoning to

accurately identify questionable credit card transactions. However, their method lacked generalization ability due to its focus on a single type of fraud - transactional fraud - and the need of fuzzy number types fine-tuning required for every different fraud type. Due to their computational requirements and limited applicability, fuzzy methods were judged inappropriate for our study's goals.

2.2. Knowledge Graphs for Fraud Detection

Although the idea of KGs dates back to the 1980s [38], it was not until DBpedia was launched in 2007 and Google's use in 2012 that they gained the popularity they have today. Despite their potential, KGs are rarely used in FD. A number of works represent the transactional data with graphs where nodes represent transactions, and focusing on unsupervised learning, model the problem of fraud detection as anomaly detection, aiming to identify anomalous nodes, aka fraudulent transactions [1,14,20]. Another direction of research more closely related to our approach focuses on the construction of domain-specific Knowledge Graphs and the use of graph embeddings or network measures and supervised ML for fraud detection. Zhang et al. [44] used an Auto Insurance Knowledge Graph to model transaction correlations. Their method included creating an ontology using ML-based feature selection and having domain experts confirm it. Using KG embeddings, they managed to enhance gang FD, a type of fraud that involves links between claimants and repair shops by integrating geographic data with string similarity. They evaluated using XGBoost, SVM and Neural Networks, showing notable performance gains in detecting individual fraud cases. Employing four centrality measures - Degree, Betweenness, Closeness and Eigenvector each providing insights into corporate networks - Wen et al. [42] built a KG centered on managers and their associations with firms. They evaluated their KG using SVN, KNN, Decision Tree (DT) and Random Forest (RF) trained on data from Chinese A-share market from 2011 to 2017, with PCA and SMOTE for preprocessing. However, they did not explore combinations of centralities potentially missing valuable contextual information.

More recently, Graph Neural Networks (GNNs) [37] have also been deployed to address fraud detection along with the use of graph-based data representations. Cherif et al. [9] presented an encoder-decoder GNN model, that effectively captures complex interactions between customer and merchants by framing FD as an edge prediction task. Their method achieves strong performance in a large dataset by integrating geographical data and engineering transaction features into a heterogeneous network structure. The model's reliance on distance-based features may however restrict its usage in environments with restricted access in real-time geolocation. Also, in [26], fraud detection in blockchain is addressed using Variational Graph Autoencoders based on a KG that is build based on the blockchain transactions. These methods also struggle with class imbalance and efforts to address the issue in this context as well have appeared, such as GraphSMOTE [45]. GraphSMOTE applies over-sampling in an embedding space it creates, while it trains both the GNN and the over-sampling module to guarantee the integration of the generated data into the learning process.

3. Knowledge Graph Based Modeling

Traditional state-of-the-art ML approaches primarily focus on individual entities when trying to detect fraudulent transactions. Typically, they examine transactions independently, ignoring any possible connections between financial transactions. For instance, the model would assess a new transaction executed by a user with a rich fraudulent record similarly to someone who solely engages in lawful activities. Recognizing that, KGs can be used to efficiently tackle this limitation, due to their natural treatment of transactions as interconnected entities.

By organizing transactions as linked entities rather than discrete events, KGs offer a practical approach, with a more comprehensive modeling of financial activities and the different relationships between transactions, the actors performing the transactions and other related properties. In this section, we introduce two distinct KG-based architectures: an *unweighted* model that captures relationships without any additional numerical characteristics and a *weighted* model that incorporates financial disparities as edge weights to improve FD.

3.1. Unweighted KG Model

They key for the creation of a KG comes down to effectively describing data that has been extracted from a data source, such as a relational database. It is crucial to select the most appropriate entities and the relationships that connect them. These characteristics would be representative of the various aspects present in a transaction, such as its temporal dimensions, transactional data and finally related actor-user information necessary to construct our KG. Since KGs by definition mandate the existence of multiple entities [16,43], a method to produce them is vital.

Following the *property graph model*, we represent the transactional data by discerning between nodes with different labels representing semantically related entities, directed, named edges that represent the different types of relationships existing between the defined entities, and finally properties that are used to describe both nodes (entities) and edges (relationships).

This is accomplished by representing each *Transaction* as a separate entity and establishing additional entities for *Users* and *Time*. The sender (Originator) and receiver (Destination) of a Transaction, are mapped using User nodes, which ensure the appropriate representation of all financial actors. A Time entity is employed to incorporate the temporal dimension, preserving the Transactions chronological order. Finally, *amount* and *type* are two distinct examples of Transaction-specific attributes, which are maintained as properties directly connected to the Transaction entity.

Maintaining the integrity of the graph structure requires the establishment of relationships that are understandable and straightforward. For instance, a *MADE_A* relationship can be used to link the Originator User to the Transaction entity and similarly a *WITH_* relationship can link the Transaction to the Destination User. Furthermore, a temporal *AT* relationship links the Transaction to the corresponding Time entity.

Careful consideration must be given to attributes which are closely tied to account balances, for both User nodes involved, before and after the transaction transpires. Based on the realistic assumption that a User can be directly involved into multiple transactions

happening simultaneously, attributing such features to User nodes could lead to redundancy. A more efficient alternative handles those features as relationship properties between Users and Transactions. By preserving similar relationship properties, not only unnecessary duplication is avoided, but also the usage of weighted centrality measurements is enabled.

Figure 1 presents a potential structure, with minimal number of features as described previously, that demonstrates how transactions and their associated entities are cohesively represented within the knowledge graph.

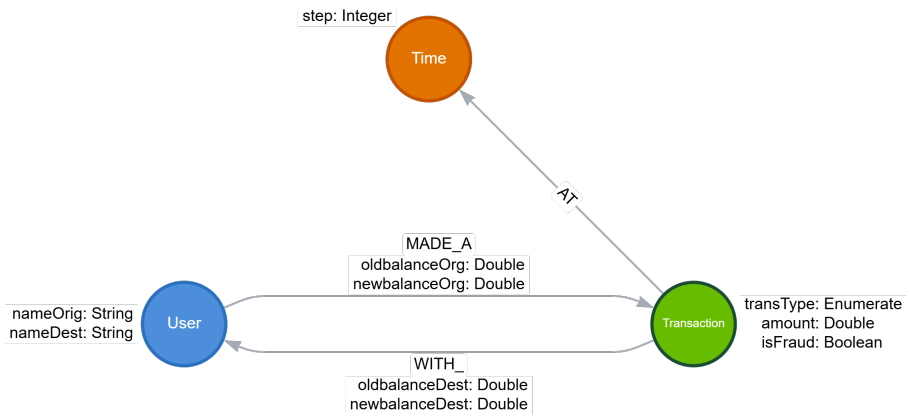


Fig. 1. Visualization of a potential unweighted Knowledge Graph structure following the property graph model. We discern between nodes labeled as: User, Transaction and Time along with their associated relationships

The naming conventions are consistent with our particular dataset, however depending on the application context they can be adapted, replaced or in some occasions omitted. By using the property graph model, our KG can be easily extended to incorporate additional properties, and even relationships or entities if these are available.

As stated, an unweighted graph is a type of graph which contains no numerical values assigned to the relationships between entities. In the context of FD, this could represent relationships between Users, Transactions and Time solely based on their presence alone. Due to the fact, that it eliminates the possibility of biases introduced from numerical properties, this kind of representation proves essential for applying structural analysis techniques such as centrality measures.

3.2. Weighted KG Model

In contrast to unweighted graphs, a weighted graph assigns numerical values, weights, to edges, so as to differentiate between relationships of varying strength or intensity. As such, a weighted graph may offer a more accurate depiction of transactional interactions by incorporating numerical values into the relationships between the entities that quantify the strength of the corresponding relationships. In other words, providing edges with

meaningful values, relationship weights provide another layer of information by capturing patterns beyond simple connectivity.

To this end, we propose the use of a weighted KG model in which we introduce three different edge weights, specific to the edge type of the specific edge they characterize. In particular, we define the *sender weight* w_{sender} representing the change in balance for the originator of the transaction, the *receiver weight* $w_{receiver}$ representing the change in balance for the recipient of the transaction, and the time weight w_{time} which captures the net transfer, as formally defined in Eq. (1), Eq. (2) and Eq. (3), respectively.

$$w_{sender(i)} = |newbalanceOrg_i - oldbalanceOrg_i| \tag{1}$$

$$w_{receiver(j)} = |newbalanceDest_j - oldbalanceDest_j| \tag{2}$$

$$w_{time(i,j)} = |w_{receiver(j)} - w_{sender(i)}| \tag{3}$$

where:

$newbalanceOrg_i$ = final balance of the Originator after the transaction.
$oldbalanceOrg_i$ = initial balance of the Originator before the transaction.
$newbalanceDest_j$ = final balance of the Destination after the transaction.
$oldbalanceDest_j$ = initial balance of the Destination before the transaction.

While in most cases one would expect the receiver and sender weights to be equal, especially in cases of fraudulent transactions this is not always the case, which is what motivated us to define the weights based on these differences instead of the transaction amount.

In Fig. 2, we present an instance of a weighted KG. Bold edges indicate a greater difference in balances between users following a transaction, while edges with normal width represent smaller balance variations. All other properties are omitted from the figure to avoid redundancy, retaining only the weights with toy values for clarity.

4. Our Approach

The application of conventional ML techniques for the problem of CCFD has long been a topic of interest for researchers, resulting in a plethora of works being published annually. These initiatives employ a variety of techniques, ranging from unsupervised learning intended for real-time FD scenarios to supervised approaches utilizing publicly available datasets. In this paper, we focus on supervised learning and propose leveraging features derived from an appropriately constructed KG to improve its performance.

4.1. Problem Definition

Adopting supervised learning to address the problem of fraud detection, we model CCFD as a problem of binary classification where the goal is to predict whether a new transaction is legitimate or fraudulent. Any ML classification method relies on known labeled historical data, called the training set, in order to learn a function (or model) that will be able to accurately predict the label of new, unknown data. A significant step in this process is feature engineering and/or selection, that is the process of constructing and/or selecting the

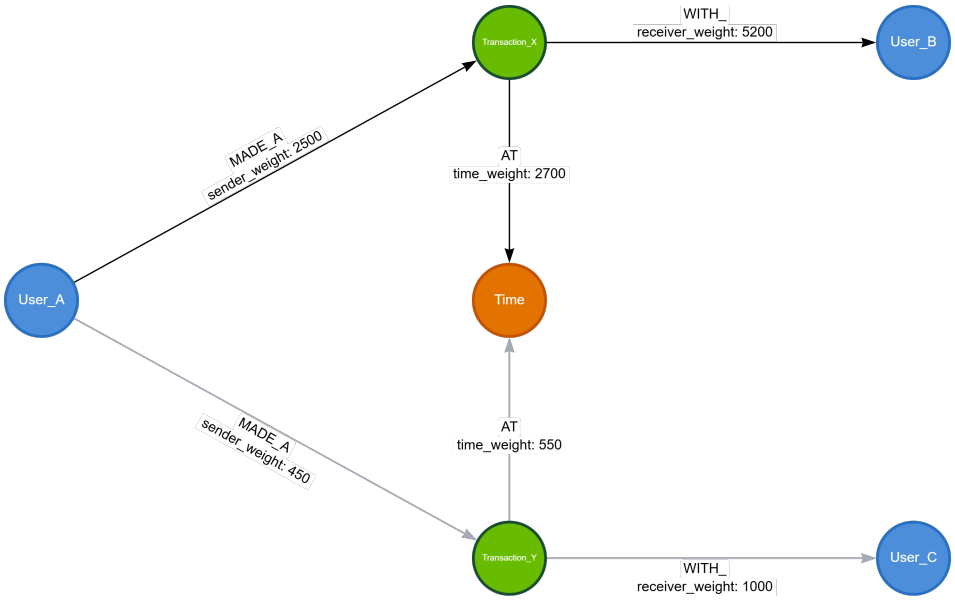


Fig. 2. Weighted KG representation. User_A has engaged in two transactions, with Transaction_X involving a larger balance difference, indicated by a bold relationship edge. Previously mentioned node and relationship features are removed to avoid redundancy

appropriate characteristics (features) of the data that will achieve the construction of an effective model with high predictive accuracy. In CCFD, this task is usually relegated to just utilizing all available transactional data attributes as input into various classification algorithms.

Problem 1 (Credit Card Fraud Detection as Binary Classification). Given a set of historical transaction records $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$, where each transaction t_i is represented by a feature vector $\mathbf{x}_i \in \mathbb{R}^d$ and an associated binary label $y_i \in \{0, 1\}$ indicating whether the transaction is legitimate (0) or fraudulent (1), the objective is to learn a classification function $f : \mathbb{R}^d \to \{0, 1\}$ that can accurately predict the label $\hat{y}$ of an unseen transaction $\mathbf{x}_{\text{new}}$.

The main objective of this work is to further enhance performance by incorporating new transactional features that frequently go undetected in the traditional aforementioned models. To that end, we propose leveraging the KG that can be derived from the same transactional data and use its structural properties as supplementary features for the ML classifiers. Let $\mathcal{G} = (V, E)$ represent our KG modeling entities such as users, transactions and temporal information as we have described in the previous section. Using network analysis, we derive additional structural features from $\mathcal{G}$ and augment the original feature vectors of our transactional data before using them as input for the ML classifiers. In particular, the feature vector $\mathbf{x}_i$ of each transaction t_i is extended as follows:

$$\mathbf{x}'_i = [\mathbf{x}_i \,\|\, \mathbf{c}_i],$$

where $\mathbf{c}_i$ denotes the vector of network features associated with t_i. As network features, we rely on the use of different node *centrality* measures, which we describe in detail in the next section.

Thus, given the set of transaction records $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ and the derived KG, $\mathcal{G} = (V, E)$, constructed from $\mathcal{T}$, our goal is to learn a classification function $f' : \mathbb{R}^{d'} \rightarrow \{0, 1\}$ based on the enhanced feature vectors $\mathbf{x}'_i$ that will predict whether a new transaction is fraudulent (1) or not (0).

The entire data analysis pipeline is shown in Fig. 3. The raw data, after any preprocessing, such as cleaning and sampling, is additionally transformed into two KG representations (weighted and unweighted). Features are extracted both from the clean transaction data and the KGs using graph analysis that yields centrality-based features. Both types of features are combined and used as input to train the ML classifiers.

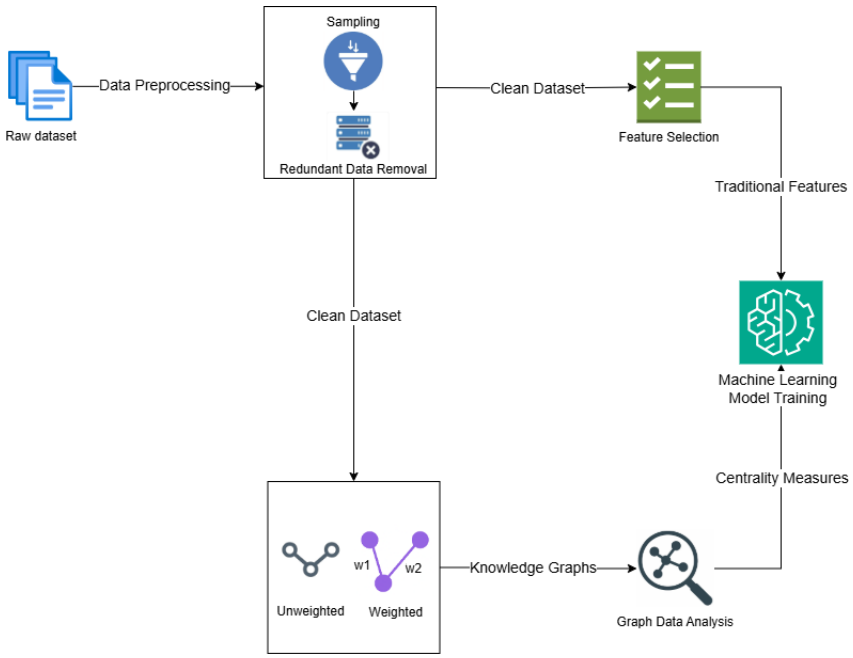


Fig. 3. Overview of the ML pipeline using graph-based features. The workflow begins with the preprocessing of the raw data. Two parallel paths are then followed: one for traditional feature extraction via feature selection and another for Knowledge Graphs construction, for both the unweighted and weighted variants. The graphs are then analyzed to extract centrality measures, which as a final step will be utilized alongside the original features to enhance machine learning model training

4.2. Graph-Based Features

Incorporating a KG into an ML pipeline comes down to effectively providing extra structural information that improves the learning process. This is usually achieved either through the usage of KG *embeddings* or *centrality measures*. The focus of this study is on the latter. Centrality measures serve as quantitative indicators of a node's significance or impact within the graph, capturing topological properties which are normally inaccessible through conventional tabular data representations. Different centrality measures assess the importance of a node in the network's structure through different perspectives, and therefore the use of several such measures can provide additional information about the role a node plays in the overall network.

Thus, in our approach, after constructing the KG based on the cleaned transactional data, we evaluate several centrality measures using both the unweighted and weighted KG model, as most of these measures are defined for both unweighted and weighted graphs. This approach enables us to experimentally study how the presence of relationship weights affects the importance of each node when incorporated into the centrality measure by the weighted model.

Let $G = (V, E)$ be an unweighted graph, where V denotes the set of vertices and $E \subseteq V \times V$ represents the set of edges connecting the pair of nodes. All edges are regarded as equal in this formulation and no numerical values are assigned to any. A weighted graph extends upon this definition to $G = (V, E, w)$, where $w : E \to \mathbb{R}$ is a weight function that assigns each edge $e \in E$ a positive real-valued weight. Various characteristics of the relationships between nodes, such as intensity, frequency or as in our case, the difference of balances between the involved users can be represented by them so as to further enrich the information conveyed by each edge in the graph with additional semantic meaning.

Based on the two definitions above, we explore the use of the following centrality measures.

Degree centrality is a simple centrality that counts the number of direct connections a node has. It shows how frequently a node interacts with other nodes in the network and nodes with high Degree are regarded as influential. Degree centrality can be expanded to a weighted version by summing the weights of all edges incident to a given node, as shown in Eq.(4). Furthermore, in directed graphs it can be further separated into in-degree and out-degree centrality, which take into consideration the quantity of incoming and outgoing edges, respectively.

$$CD(v) = \begin{cases} \deg(v), & \text{if the graph is unweighted} \\ \sum_{u \in N(v)} w_{vu}, & \text{if the graph is weighted} \end{cases} \tag{4}$$

where:

$\deg(v)$ = number of edges connected to node v.
$N(v)$ = set of neighboring nodes of v.
w_{vu} = weight of the edge between nodes v and u.

Eigenvector [3,23] centrality expands upon the idea of simply counting a node's direct links by taking into account the importance of the nodes to which said node is connected. A node linked to other highly influential nodes will attain a higher eigenvector

score. The measure assumes unweighted relationships by default but this can be easily modified to consider relationship weights. In this case, the score of a node in the previous iteration sent to its neighbors is multiplied by the scaled relationship weight. Negative weights are also ignored in this computation. Detailed formulation of the unweighted and weighted approaches are presented in Eq.(5) and Eq.(6), respectively.

$$EC_i = \frac{1}{\lambda} \sum_{j \in N(i)} A_{ij}\, EC_j \tag{5}$$

$$EC_i = \frac{1}{\lambda} \sum_{j \in N(i)} w^*_{ij}\, EC_j \quad \text{with} \quad w^*_{ij} = \frac{\max(w_{ij}, 0)}{\sum_{k \in N(i)} \max(w_{ik}, 0)} \tag{6}$$

where:

λ = largest eigenvalue of the adjacency matrix.
A_{ij} = (i, j) element of the adjacency matrix.
$N(i)$ = set of neighbors of node i.
w_{ij} = weight of the edge from node i to j.
w^*_{ij} = normalized weight of the edge from node i to j.

Acting as extensions of the Eigenvector centrality, **PageRank** [6] and **ArticleRank** [28] were developed to assess a node's relevance according to its relationships as well. PageRank, which was originally developed to rank web pages, assigns scores by considering a random-walk based approach and assigning higher scores to nodes that have a high probability of appearing in any random walk in the graph. Following this approach, higher PageRank values are given to nodes that receive links from nodes with a high ranking as well. ArticleRank follows a similar principle when it comes to academic citations, where each citation is treated as a vote of importance. Citations from influential articles are considered of greater importance. Eq.(7) and Eq.(8) present the formal equations for PageRank and ArticleRank respectively.

$$PR(u) = \frac{1-d}{|V|} + d \sum_{v \in N_{in}(u)} \frac{PR(v)}{out_deg(v)} \tag{7}$$

$$\mathrm{AR}_i(u) = (1-d) + d \sum_{w \in N_{in}(v)} \frac{\mathrm{AR}_{i-1}(w)}{|N_{out}(w)| + \overline{out_deg}} \tag{8}$$

where:

d = damping factor $\in [0, 1]$
V = set of nodes (pages) in the graph.
$N_{in}(u)$ = set of nodes linking to u.
$N_{out}(u)$ = set of nodes that node u links to.
$out_deg(u)$ = number of outgoing edges from node u.
$\overline{out_deg}$ = average out-degree.

Both can also be adapted to weighted versions similarly to Eigenvector centrality.

Another popular centrality metric in network analysis is the **Hyperlink-Induced Topic Search (HITS)** algorithm [25]. Designed also initially to support web search, it however has considerable differences with PageRank. As mentioned, PageRank assesses the overall significance of a node, while HITS differentiates between two specialized roles: hubs and authorities. Nodes with high hub scores are treated as efficient aggregators that link authoritative sources, while nodes with high authority ratings are considered reliable sources of information. By evaluating roles separately, HITS provides a more sophisticated understanding of structural functionality in the graph. Eq. (9) and (10) show the results of the hub and authority score computations for the weighted graph, respectively. They can easily be applied to the unweighted graph if we consider all edge weights equal to 1.

$$h_v = \sum_{u \in N_{out}(v)} a_u \cdot w(v,u) \tag{9}$$

$$a_v = \sum_{u \in N_{in}(v)} h_u \cdot w(u,v) \tag{10}$$

where:

$w(v,u) =$ weight of the link between node v and node u.

Betweenness [17,4,5] centrality measures a node's importance based on how frequently it appears on the shortest paths between any given pair of nodes. It showcases to what extent a node acts as a bridge in the graph, enabling communications between nodes. Nodes with high betweenness scores are thought to be essential to the flow of information. Eq.(11) presents the formula for calculating the betweenness centrality score.

$$C_B(v) = \sum_{s,t \in V \;\; s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{11}$$

where:

$\sigma_{st}(v) =$ number of those paths that pass through node v.
$\sigma_{st} \quad =$ total number of shortest paths from node s to node t.

The measure can be adopted for weighted graphs if, instead of just counting the number of shortest paths a node belongs to, we add the total edge weight of each such path, so as to account for its importance.

Closeness [18] centrality evaluates how close a node is to all other nodes in the graph. In its simplest form, it is defined as the inverse of the sum of the shortest path length from the given node to all other reachable nodes. However, normalizing this score, as presented in Eq.(12), to reflect the average of length of the shortest paths rather than their sum is more common.

$$C_C(v) = \frac{n-1}{\sum_{u \in V \setminus v} d(v,u)} \tag{12}$$

where:

$d(v,u) =$ shortest path length between node v and node u.

Similarly to betweeness, we can also adjust closeness centrality for weighted graphs by considering the total path weight instead of its length in our calculations.

The **Cost-Effective Lazy Forward (CELF)** [27] algorithm is an optimization-based greedy approach often used in influence maximization problems. It ranks nodes according to their marginal gain in a specific influence function, such as spread in a social network. CELF score at each step, given a monotonic submodular function f, is determined by Eq.(13) for both unweighted and weighted graphs.

$$\Delta f(v \mid S) = f(S \cup v) - f(S) \tag{13}$$

where:

$\Delta f(v \mid S)$ = marginal gain from adding v to S.
f = influence spread function or centrality objective.
S = current set of selected nodes.
v = candidate node.

4.3. Implementation

To support efficient KG construction and graph analysis, we implement both KG models in a graph database so as to exploit its querying and indexing capabilities that offer scalability and high performance. In particular, we use Neo4j[1], which is one of the most popular graph databases. Neo4j uses the property graph model that aligns with the proposed model for our KGs and furthermore, offers the Graph Data Science Library (GDSL) that is a library designed to support network analysis and provides methods for the efficient evaluation of several centrality measures including the ones we have described above that we use in our experimental evaluation.

5. Experimental Evaluation

We experimentally evaluate the proposed approach both with respect to its effectiveness, i.e., the detection accuracy, and its efficiency in terms of time efficiency. We first establish a baseline approach using only tabular transactional data as features and compare the performance of our approach against it.

5.1. Experiment Setup

For the evaluation of the effectiveness of our approach, three well-known metrics suited for our problem are employed: Precision, Recall and F1-Score, Eq.(14), Eq.(15) and Eq.(16), respectively. In addition, the Receiver Operating Characteristic (ROC) Curve is utilized to visually assess the trade-off between the True Positive Rate (TPR) and the

[1] Neo4j

False Positive Rate (FPR) across various classifiers.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

where:

TP = true positives.
FP = false positives.
FN = false negatives.

The primary objective of CCFD is to minimize the amount of fraudulent transactions that go undetected, with emphasis on the accurate determination of such cases. Among the different metrics, recall is the most important, since it quantifies the percentage of correctly detected fraudulent transactions. Due to the nature of our problem, reducing false negatives - cases in which fraud activity is overlooked - is of most importance. On the other hand, precision illustrates the percentage of illegitimate transactions correctly identified, among all of the observations flagged by our model. As a result, it focuses on the number of false positives, which are valid transactions wrongfully classified as fraud. Given the context, we prioritize algorithms that maximize recall, even at the slight cost of precision. It is crucial to note that, because of the dataset's imbalance we opted not to include Accuracy in our analysis.

We utilized 5-fold cross-validation, a common approach that divides the dataset into training and testing subsets. However, to reduce the effects of the aforementioned class imbalance, especially the under-representation of fraudulent cases, we used stratified sampling. This method ensures equal proportional representation of all classes in all different folds.

Dataset One of the biggest challenges, in the domain of CCFD and FD in general, is the scarcity of publicly available real-world datasets, which is predominantly the result of growing privacy concerns. Due to that reason, research has shifted towards the usage of creative solutions, such as generating synthetic datasets for ML model training. Eliminating any possible identifiable personal information, these synthetic datasets adhere with major privacy protection frameworks, including the European Union's General Data Protection Regulation (GDPR) [32,11] and the California Consumer Privacy Act (CCPA). Since synthetic data is completely anonymized with no real-world identifiers, it can be used and distributed without breaking privacy laws.

For this study, we use the Synth Financial Dataset for Fraud Detection [29], generated using the PaySim mobile money simulator [30]. Paysim attempts to generate synthetic transaction data that realistically mirror the behavior of real-world mobile money systems. It replicates different transaction types and simulates user-agent interactions using probabilistic modeling based on real transaction records from a mobile service provider in an African country. Those agents perform one of five financial actions: *CASH-IN*, *CASH-OUT*, *DEBIT*, *PAYMENT*, or *TRANSFER*, depending on behavior rules and restrictions,

Table 1. Overview of the synth dataset attributes. In the table, we define abbreviations using (), which are to be used later when referring to those features

Attribute	Description	Possible Values	Example
Step (9)	Simulation time step (in hours)	1 – 744	0
Type (8)	Transaction type	`CASH-IN`, `PAYMENT`, `TRANSFER`, `etc.`	CASH-OUT
Amount (7)	Value of the transaction	Any positive float	4782.71
NameOrig (1)	ID of transaction initiator	Alphanumeric string	B90293651
OldbalanceOrg (2)	Sender's balance before transaction	Any float	75631.32
NewbalanceOrg (3)	Sender's balance after transaction	Any float	12398.65
NameDest (4)	ID of transaction receiver	Alphanumeric string	X239613845
OldbalanceDest (5)	Receiver's balance before transaction	Any float	31231.12
NewbalanceDest (6)	Receiver's balance after transaction	Any float	93841.43
isFraud	Fraud label (ground truth)	0 = No, 1 = Fraud	0
isFlaggedFraud	Flag if over threshold (e.g., 200K)	0 = No, 1 = Yes	0

such as daily withdrawal limits that an account may have. This framework enables PaySim to produce transactional data that is incredibly realistic and flexible, closely resembling genuine financial transactions.

The synthetic dataset is composed of 11 attributes, as described in [30]. The *Step* feature represents hour simulation over the course of a 30-day period. Transaction category is identified by its *Type*, with the 5 aforementioned actions. *Amount* records the monetary value of the transaction. The originator (sender) and recipient (destination) are denoted by the *NameOrig* and *NameDest* respectively. *OldbalanceOrg*, *NewbalanceOrg*, *OldbalanceDest*, and *NewbalanceDest* provide the account balances of the originator and destination prior and after each transaction occurs. Fraudulent transactions are identified with the binary *isFraud* flag, whereas *isFlaggedFraud* marks transactions tagged as suspicious based on specific heuristics. Table 1 provides a sample entry and synopsis of these fields.

To improve computational efficiency, the original dataset was reduced to 25% of its original size. Of these, only 1142 entries were identified as fraud, accounting for just 0.108%, which indicates a notable class imbalance. Thorough examination of the dataset was conducted in order to tackle this issue. It was observed that, out of all transactional activities only in two, namely *CASH-OUT* and *TRANSFER*, fraudulent transactions occurred. Therefore, the remaining transaction types which did not exhibit any fraudulent behavior, were excluded. This resulted in a reduction of the downscaled dataset to about 50% while maintaining all fraudulent cases. Finally, attribute *isFlaggedFraud* which had a value of only 0, providing no real value for classification, was removed.

The Knowledge Graph (KG) constructed from the preprocessed dataset consisted of 1,006,918 nodes and 1,381,182 relationships, with the resulting database (including the KG structure and metadata) occupying approximately 4 GB of storage.

Table 2. Hyperparameter configurations for the evaluated ML models

Model	Hyperparameters	Values / Ranges
K-Nearest Neighbors	Number of neighbors (n) Weights	{4, 5, 6} {uniform, distance}
Logistic Regression	Regularization parameter (C) Maximum iterations	{1e-6, 1e-5, ..., 1e-4} {500, 1000, 1500}
Support Vector Machine	Kernel type Regularization parameter (C)	{linear, polynomial} {1, 2}
Naïve Bayes	Variance smoothing Class priors	{1e-9, 1e-8, 1e-7} {[0.4, 0.6], [0.3, 0.7], None}
Decision Tree	Split criterion Maximum leaf nodes	{gini, entropy} {4, 5, 6}
Random Forest	Number of estimators Split criterion	{50, 100, 150} {gini, entropy}
XGBoost	Number of estimators	{50, 100, 150}

All experiments were carried out on a Windows 10 with 16GB of RAM and AMD Ryzen 5 processor. The implementation was developed in Python 3.10, utilizing essential libraries such as scikit-learn and Neo4j. The source code needed to replicate our approach is publicly available on GitHub[2].

Classification Algorithms To expand upon existing research, this study leverages seven well established ML algorithms, each known for its efficiency in previous FD attempts. These include *K-Nearest Neighbors (KNN)*, which groups data points based on the majority class among their nearest neighbors; *Logistic Regression (LR)*, a linear model that calculates the likelihood of a binary result; *Support Vector Machine (SVM)*, which finds the best hyperplane which separates classes as much as possible; *Naïve Bayes (NB)*, a probabilistic classifier which assumes independence based on the Bayes theorem; *Decision Tree (DT)*, a tree-like structure of decisions dividing data into branches based on a criterion and feature values; *Random Forest (RF)*, an ensemble of DTs utilizing a technique called Bagging, averaging the results of those DTs to enhance prediction performance; *XGBoost*, a highly effective gradient boosting algorithm that builds DTs sequentially to correct previous errors and *Bagging* with a custom architecture which combines XGBoost, KNN and LR algorithms. Table 2 highlights the specific hyperparameters used for each model.

5.2. Baseline Performance

The establishment of a baseline, which serves as a reference point upon which we will compare our results is imperative and essential for all ML problems. In Table 3, we depict the performance metrics of the assessed classification models using only the original preprocessed dataset excluding the *isFlaggedFraud* column.

[2] KG-Fraud-Detection

Table 3. Baseline performance of ML models using the original features

Algorithm	Precision	Recall	F1-Score
K-Nearest Neighbors	0.830	0.436	0.572
Logistic Regression	0.798	0.382	0.516
Support Vector Machine	0.806	0.420	0.552
Naïve Bayes	0.044	0.322	0.074
Decision Tree	0.850	0.460	0.594
Random Forest	0.980	0.778	0.870
XGBoost	**0.984**	**0.826**	**0.898**
Bagging	0.982	0.812	0.889

Table 4. Confusion matrix for the XGBoost classifier

		Prediction Outcome	
		Legitimate	Fraudulent
Actual Value	Legitimate	91847	3
	Fraudulent	42	187

The classifiers are generally accurate on the positive predictions with few false positives, achieving much higher precision scores than recall, indicating that they miss a significant chunk of real fraud cases, which is far more important given the nature of our problem. This is particularly apparent in models such as KNN, SVM and LR, which maintain acceptable precision around 80% but exhibit relatively low recall from 0.38 to 0.43. NB underperforms significantly when compared to all other algorithms, especially in precision with a value as low as 0.044, indicating weak performance for the task.

Tree-based models on the other hand, show much more promising results with RF and XGBoost being the best among them. More specifically, XGBoost achieves the best scores across all metrics, with an impressive F1-Score of 0.898, correctly identifying the majority of fraudulent and legitimate transactions as indicated by the confusion matrix shown in Table 4, which also highlights the strong class imbalance between the two classes.

5.3. Evaluation of Detection Effectiveness

The impact of adding graph centrality measures into the FD pipeline is explored in this section. Our goal is to evaluate the influence these structural features have on model performance under various configurations. The analysis is divided into multiple subsections, each focusing on a different experimental dimension. We first start by assessing the different centrality metric combinations, followed by the effects of removing features initially in our dataset while keeping our centralities. Finally, we examine how resampling affects the outcomes and explore weighted centralities. In the following, centrality measures are abbreviated as: (D) Degree (unweighted), (E) Eigenvector (unweighted), (H)

Table 5. F1-Score performance using single centralities

Centrality	KNN	LR	SVM	NB	DT	RF	XGBoost	Bagging
(D)	0.669	0.678	0.515	0.210	0.626	0.873	**0.902**	0.898
(H)	0.576	0.528	0.459	0.076	0.659	0.876	**0.903**	0.9
(E)	0.577	0.516	0.461	0.075	0.659	0.876	**0.897**	**0.897**

Table 6. F1-Score performance using pairs of centralities

Combination	KNN	LR	SVM	NB	DT	RF	XGBoost
(D) + (E)	0.674	0.68	0.75	0.14	0.802	0.876	0.91
(D) + (H)	0.674	0.68	0.75	0.14	0.628	0.872	**0.92**

HITS, (P) PageRank, (B) Betweenness, (HC) Harmonic Closeness, (C) CELF, (WD) Degree (weighted), (WE) Eigenvector (weighted).

Combinations of Centralities In this set of experiments, we examine the performance of different unweighted centrality measure combinations when used as additional features in the ML pipeline. We assess the results for individual centralities and various combinations (pairs, triplets and quadruplets).

Initially, we examine performance for single centralities. Table 5 shows the results for Eigenvector, HITS and Degree centrality as these were the best performing centrality measures. First of all, we observe that even the addition of one centrality measure has a positive effect compared with the performance of the baseline approach, as both Degree and HITS reach an F1-Score higher than 90% for the best performing classifier XGBoost. We find that HITS and Degree alone yield good results, especially when paired with tree-based models, while Eigenvector individually lags behind the other two. Other metrics showed poor standalone performance, which did not justify any additional review and for that reason we omit their results.

Following the same idea, we also examined the performance when adding pairs of centrality measures into the classification features. We observed that most of the combinations did not result into a significant performance improvement. However, both combinations of Degree with Eigenvector or HITS returned promising results, as seen in Table 6. With relatively little variation in performance when it comes to simpler algorithms such as KNN, LR, SVM and NB, both combinations produce similar F1-Score. Interestingly, in DT, Degree with Eigenvector showcases a substantial difference of approximately 18%, suggesting that even though both centrality pairs are generally good predictors, model architecture may affect how effective they are. XGBoost with the pair of Degree and HITS slightly outperformed that of Degree and Eigenvector reaching the highest F1-Score of 0.92 compared to 0.91.

We then investigate several triplets of centralities. Table 7 provides a summary of the findings. Interestingly, combinations including HITS and Eigenvector showcase improved performance, particularly when combined with PageRank or CELF. XGBoost with HITS, Eigenvector and PageRank (or CELF) outperform all single centralities, achieving an F1-Score of 0.91.

Table 7. F1-Score performance using triplet combinations of centralities

Combination	KNN	LR	SVM	NB	DT	RF	XGBoost	Bagging
(D) + (H) + (P)	0.67	0.682	0.52	0.212	0.658	0.856	0.906	0.896
(D) + (H) + (B)	0.67	0.682	0.52	0.212	0.628	0.872	0.904	0.9
(D) + (H) + (HC)	0.67	0.684	0.522	0.212	0.658	0.856	0.906	0.9
(D) + (H) + (C)	0.67	0.682	0.52	0.212	0.628	0.868	0.904	0.896
(H) + (E) + (P)	0.578	0.662	0.518	0.076	0.658	0.854	**0.91**	0.9
(H) + (E) + (B)	0.578	0.642	0.514	0.076	0.658	0.864	0.902	0.9
(H) + (E) + (HC)	0.578	0.642	0.516	0.076	0.658	0.862	**0.91**	0.9
(H) + (E) + (C)	0.578	0.642	0.514	0.076	0.658	0.866	0.902	0.9
(E) + (B) + (C)	0.578	0.642	0.514	0.076	0.688	0.876	0.908	0.898

Table 8. F1-Score performance using quadruplet combinations of centralities

Combination	KNN	LR	SVM	NB	DT	RF	XGBoost	Bagging
(D) + (E) + (B) + (C)	0.67	0.682	0.518	0.212	0.688	0.864	**0.906**	0.902
(D) + (E) + (B) + (HC)	0.67	0.682	0.52	0.212	0.688	0.858	0.902	0.9
(D) + (E) + (B) + (P)	0.67	0.684	0.52	0.212	0.688	0.862	0.902	0.898
(H) + (E) + (B) + (C)	0.578	0.642	0.514	0.076	0.658	0.866	0.902	0.9
(D) + (H) + (E) + (B)	0.67	0.68	0.514	0.212	0.658	0.852	0.902	0.894

Furthermore, we also assessed four centrality combinations, which are outlined in Table 8. We note that the quadruplet of Degree, Eigenvector, Betweenness and CELF exhibits consistent performance across all classification models, and performs very well when used with XGBoost. The slight losses, of 1%, compared with the best triplet combinations, however, suggest a diminishing return with additional centralities, as seen when comparing the best performing algorithm XGBoost using triplets (HITS, Eigenvector and PageRank or Harmonic Closeness) and the aforementioned quadruplet.

To get a better understanding of the dynamics between precision and recall we present the scores for the top-performing single centrality (Degree), pair (Degree and HITS), triplet (HITS, Eigenvector and PageRank) and quadruplet (Degree, Eigenvector, Betweenness and Closeness) in Fig. 4. Across the top-performing models, precision remains consistently high ($> 0.98\%$). Recall on the other hand, tends to be quite lower but improves when combinations of centralities are used instead of solo measures. The gap between metrics narrows slightly when more centralities are added, enhancing FD without sacrificing either metric. Figure 5 additionally depicts the corresponding ROC curve for each best centrality combination for all classifiers, showing that the models with multiple centralities in the feature set tend to have more steep rises in the True Positive Ratio (TPR) early on, suggesting faster identification of fraudulent transactions at lower False Positive Rates (FPR).

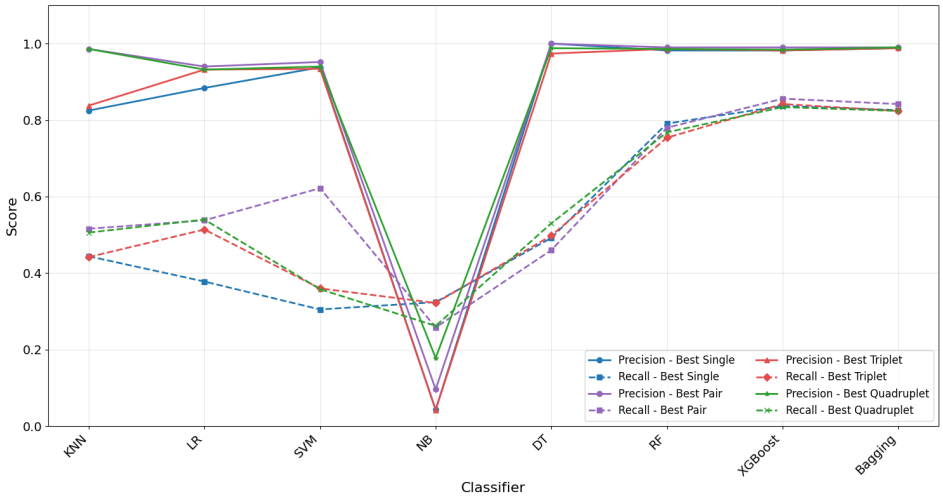


Fig. 4. Precision and recall for best single, pair, triplet, and quadruplet of centralities across all classifiers

Removal of Features In this experiment, we study feature selection with respect to the original transactional features of our dataset. We utilize the best pair of centralities as our network-based features, i.e., Degree with HITS, and evaluate the performance when successively removing attributes from the features vector. The impact of sequential initial feature removal on model performance, as measured by the F1-Score, is summarized in Table 9. Features are abbreviated using the numbers: NameOrig (1); OldbalanceOrg (2); NewbalanceOrg (3); NameDest (4); OldbalanceDest (5); NewbalanceDest (6); Amount (7); Type (8) and Step (9), defined previously in Table 1.

Initially, most models retain relatively good performance up until feature (4) is eliminated (around 0.1 decline). Almost all models show a progressive decline in performance as more features are removed on (5) through (7). Having said that, the ensemble techniques (RF, XGBoost, Bagging) demonstrate exceptional resilience, maintaining F1-Scores above 0.7 even after the removal of six features (7), which may be significant when considering that the cost of training the models with less features becomes more efficient.

Sampling To counter the effect of class imbalance, we evaluate the effect of deploying sampling techniques. As highlighted in Table 6, we include the results based on the best two centrality pairs: Degree with Eigenvector and Degree with HITS. As seen in Fig. 7, we use SMOTE [8] to evaluate the effects of data resampling and compare the outcomes before (Fig. 6) and after resampling. Even though comparable results can be seen between the two figures, we observe slightly lower performance, around 1%, in terms of F1-Scores after applying SMOTE on KNN, RF and XGBoost. On the other hand, a substantial difference can be observed in SVM with a 24% and 23% drop in performance on Degree with Eigenvector and Degree with HITS, respectively. Surprisingly, NB is the only algorithm which benefits from the usage of resampling with 7% increase on both combinations, but its overall low performance does not justify the use of sampling in our pipeline.

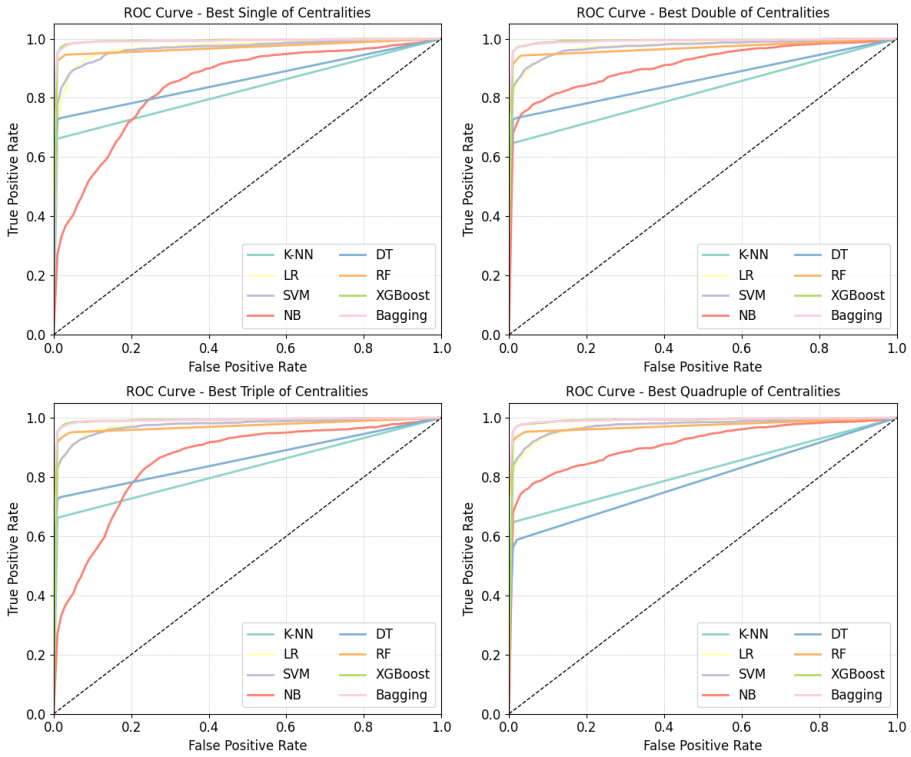


Fig. 5. ROC Curves for best centrality combinations across all classifiers. From left to right: best (1) single, (2) pair, (3) triple, and (4) quadruplet of centralities. The curves represent the mean true positive rate (TPR) across 5-fold cross-validation for the different classifiers, while the diagonal dashed line indicates the performance of a random classifier

Table 9. F1-Scores across models after sequential removal of features. Deleted feature sets are labeled (1) - (9). Empty cells indicate models not evaluated or not applicable

Deleted Feature(s)	KNN	LR	SVM	NB	DT	RF	XGBoost	Bagging
(1)	0.668	0.686	0.52	0.212	0.628	0.872	0.904	0.902
(1) - (2)	0.646	0.452	0.172	0.184	0.68	0.84	0.848	0.846
(1) - (3)	0.646	0.44	0.168	0.208	0.68	0.828	0.844	0.85
(1) - (4)	0.646	0.588	0.166	0.208	0.768	0.828	0.85	0.844
(1) - (5)	0.642	0.588	0.138	0.184	0.644	0.782	0.8	0.798
(1) - (6)	0.65	0.528	0.088	0.164	0.628	0.71	0.72	0.722
(1) - (7)	0.64				0.802	0.646	0.662	0.656
(1) - (8)	0.64				0.628	0.648	0.666	0.664
(1) - (9)	0.64				0.606	0.618	0.664	0.668

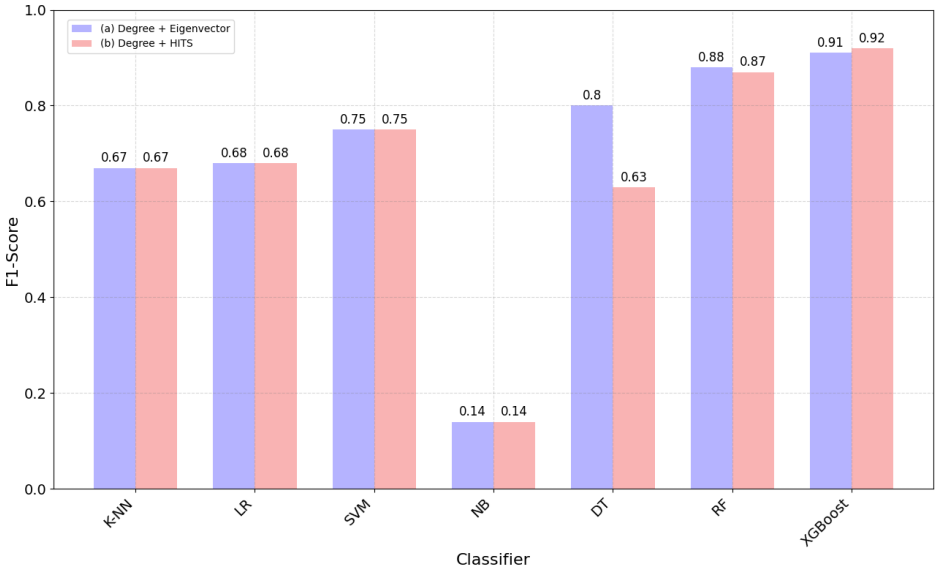


Fig. 6. F1-Scores without SMOTE for each centrality combination across classifiers

Table 10. F1-Scores across classifiers for different **weighted** centrality combinations

Method	K-NN	LR	SVM	NB	DT	RF	XGBoost	Bagging
(WD)	0.66	0.68	0.50	0.25	0.63	0.88	0.91	0.90
(WE)	0.58	0.53	0.51	0.08	0.71	0.88	0.90	0.91
(WD) + (WE)	0.66	0.66	0.51	0.24	0.71	0.89	0.90	0.91
(WD) + (H)	0.66	0.67	0.51	0.25	0.66	0.87	0.90	0.89
(WD) + (WE) + (H)	0.67	0.67	0.52	0.25	0.66	0.86	**0.92**	0.91

Weighted Model The introduction of weights into our graph shows promising results, as most classifiers benefit from them to a certain degree (Table 10). Among the different combinations of weighted centralities explored, only Weighted Degree (WD) and Weighted Eigenvector (WE) were retained for reporting, as they significantly outperformed the rest. WD already provides strong performance, notably for XGBoost with an F1-Score of 0.91. WE individually performs slightly worse than WD for simpler models such as KNN and LR, while SVM and DT show an increase of 1% and 8%, respectively. Both weighted metrics perform slightly better than their unweighted counterparts with around 1% increase, suggesting that the introduction of transaction-based weights assists in the identification of fraudulent from legitimate transactions. The addition of HITS (H) to the weighted centralities, enhances even further the performance achieving the best overall F1-Score of 0.92. Finally, it is worth noting that our custom Bagging architecture provides results comparable with XGBoost and in some cases exceeding them ((WD) + (WE)).

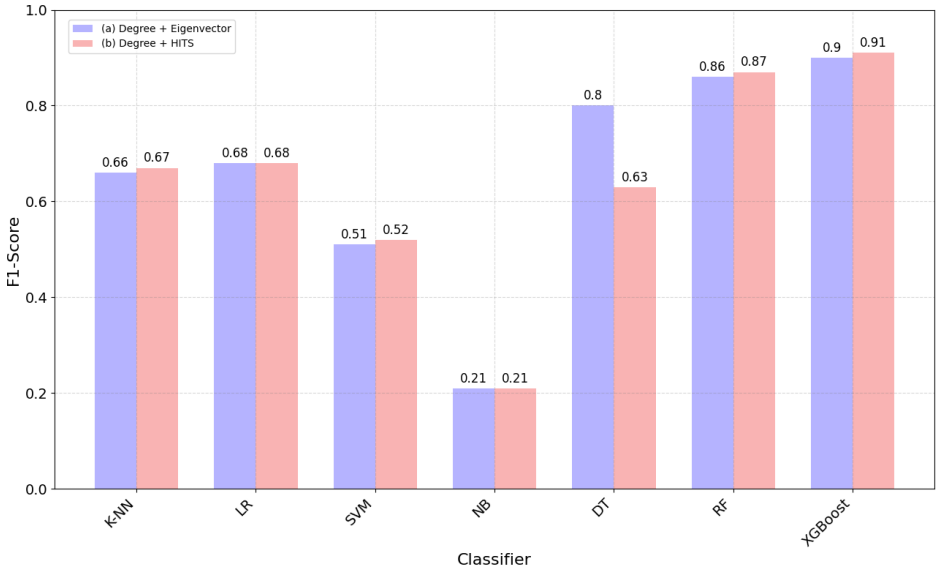


Fig. 7. F1-Scores after SMOTE for each centrality combination across classifiers

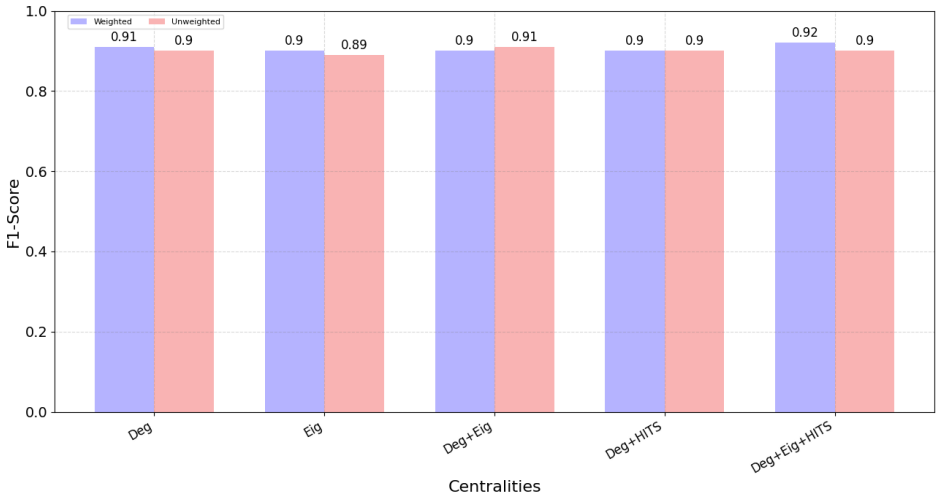


Fig. 8. Comparison of F1-Scores for XGBoost: weighted vs unweighted centralities

In Fig. 8, we provide a side-by-side comparison of the F1-Scores for the highest performing algorithm XGBoost, for both the weighted centralities and their unweighted counterparts, showing the slight superiority of the weighted model that leads us to conclude that the exploration of alternative weighting schemes could also be another research direction worth investigating in the use of KGs for fraud detection.

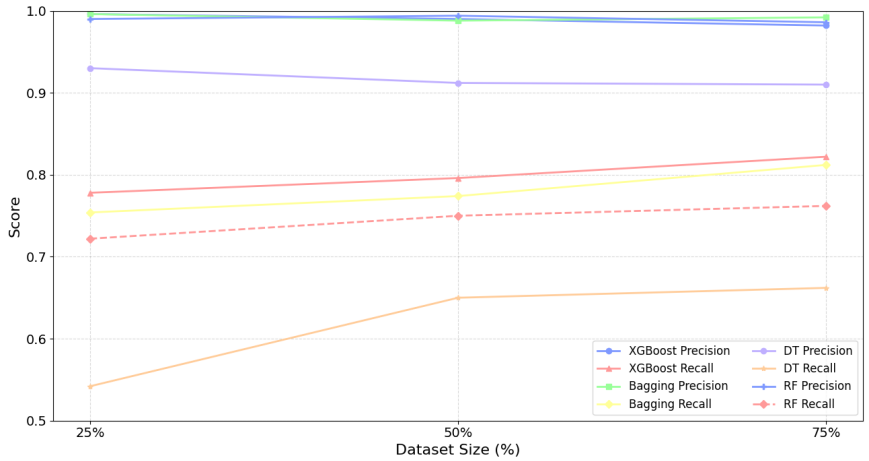


Fig. 9. Precision and Recall of various classifiers across different dataset sizes

5.4. Evaluation of Efficiency and Scaling

Finally, in this section, we evaluate the scaling abilities and time efficiency of the proposed approach. In particular, we examine the behavior of the suggested methods with a constantly increasing dataset size, with emphasis on both predictive performance and scalability. Figure 9 illustrates how various algorithms exhibit excellent resilience to data volume increase while maintaining high precision and recall, for the pair of Degree and HITS, our best pair combination as exhibited in the previous experiments. Recall somewhat improves with increasing data availability (around 0.05), indicating improved generalization for larger dataset sizes on ensemble algorithms such as RF, XGBoost and Bagging. Interestingly, DT benefits even further with an increase of approximately 0.12.

Analysis of execution time is presented in Fig. 10, which reveals significant variations among classifiers. Bagging and RF exhibit notable increases of computational cost, around 55% and 46% from 25% to 75% of dataset size increase, respectively. On the other hand, all other algorithms remain relatively close in terms of additional overhead w.r.t dataset size increase.

Figure 11 illustrates the change in execution time as individual features are removed based on the corresponding experiment we have described. Notably ensemble methods such as RF and Bagging display noticeably decreases in execution times, with Bagging's runtime decreasing from more than 200 to less than 40 minutes in the last stages of feature removal. In contrast, lightweight classifiers such as LR, NB and DT show minimal runtime variations. Thus, while feature removal does not seem to be worth the decrease in results quality for simpler classifiers, for more complicated ones, the small deterioration in F1-Score may be worth it considering the significant gains in efficiency when considering less features for training.

Finally, Table 11 shows that among the different computations in Neo4j, Betweenness centrality is by far the most expensive with 49 minutes, while Degree, Eigenvector and HITS are computed almost instantly, showing that the cost of exploring these additional features does not significantly impact the total approach performance as using a graph

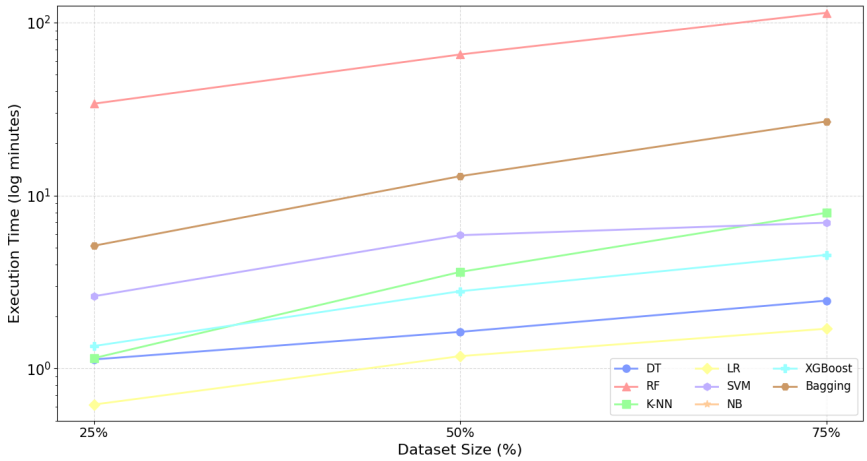


Fig. 10. Execution time (log-scaled minutes) of all classifiers across increasing dataset sizes

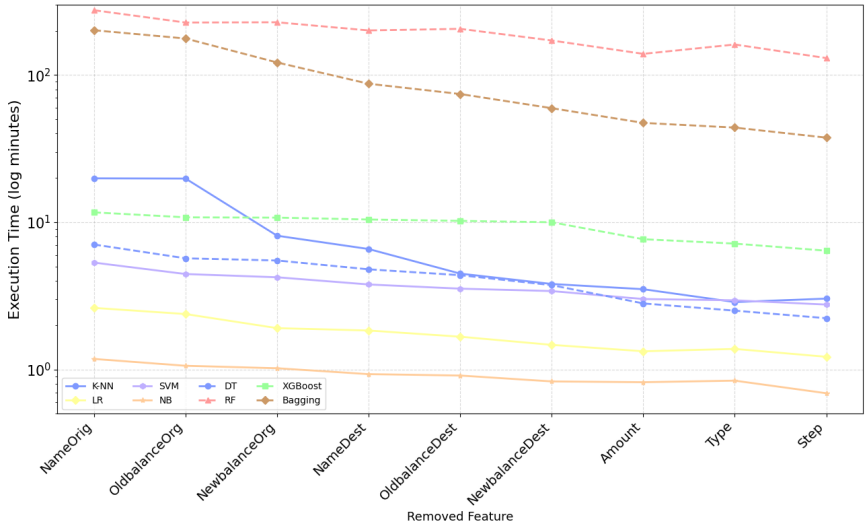


Fig. 11. Execution time (log-scaled minutes) variation as each feature is removed from the dataset

database for these computations practically alleviates the incurred overhead. Therefore, the additional costs are mainly due to the classifiers training as we explored in our previous experiments based on the dataset size, and the dimension of the features vector.

Table 11. Neo4j centrality computation times

Centrality Metric	Degree	Eigenvector	HITS	Betweenness	Closeness
Execution Time (min)	0.16	0.31	0.36	49.02	2.00

6. Conclusions and Future Work

In this study, we proposed a novel approach to address Credit Card Fraud Detection (CCFD) by utilizing a graph-based representation of the transaction network. Our methodology is focused on building a Knowledge Graph in which structural information obtained from centrality measures is added to the transaction data. Utilizing both unweighted and weighted centralities, specifically Degree, Eigenvector and HITS, we experimentally showed how the original feature vector is enhanced and demonstrates improved predictive performance across various classifiers. Our experimental results showed that, incorporating graph features, boost Fraud Detection effectiveness, with ensemble models such as XGBoost and our custom Bagging architecture consistently outperforming other models. The highest F1-Score was obtained from the combination of Degree and HITS and the use of transaction-based weights further improved the classifiers ability to distinguish between fraudulent and legitimate transactions. We also tested the scalability of our approach, demonstrating consistent performance as data volume increases, while extensive computation analysis provided valuable insights into trade-offs between feature richness and execution time.

In our future plans, we intend, in order to obtain richer end-to-end representations directly form the graph structure, to investigate the usage of graph neural networks (GNNs) in our ML pipeline. Furthermore, the exploration of temporal graph modeling, to dynamically update centrality measures for evolving fraud behavior and support real-time detection, could hold promising results, while a deeper exploration of different weighted centrality measures for specific classifiers could also be of interest.

References

1. Akoglu, L., McGlohon, M., Faloutsos, C.: oddball: Spotting anomalies in weighted graphs. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 410–421. Springer Berlin Heidelberg (2010)
2. Alarfaj, F.K., Malik, I., Khan, H.U., Almusallam, N., Ramzan, M., Ahmed, M.: Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. IEEE Access 10, 39700–39715 (2022)
3. Bonacich, P.: Technique for analyzing overlapping memberships. Sociological Methodology 4, 176–185 (1972), http://www.jstor.org/stable/270732
4. Brandes, U.: A faster algorithm for betweenness centrality*. Journal of Mathematical Sociology 25(2), 163–177 (6 2001), https://doi.org/10.1080/0022250x.2001.9990249
5. Brandes, U., Pich, C.: Centrality Estimation in Large Networks. International Journal of Bifurcation and Chaos 17(07), 2303–2318 (7 2007), https://doi.org/10.1142/s0218127407018403
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1), 107–117 (1998)

7. Charizanos, G., Demirhan, H., İçen, D.: An online fuzzy fraud detection framework for credit card transactions. Expert Systems with Applications 252, 124127 (2024), https://www.sciencedirect.com/science/article/pii/S095741742400993X
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority oversampling technique. J. Artif. Int. Res. 16(1), 321–357 (Jun 2002)
9. Cherif, A., Ammar, H., Kalkatawi, M., Alshehri, S., Imine, A.: Encoder–decoder graph neural network for credit card fraud detection. Journal of King Saud University - Computer and Information Sciences 36(3), 102003 (2024), https://www.sciencedirect.com/science/article/pii/S1319157824000922
10. Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., Imine, A.: Credit card fraud detection in the era of disruptive technologies: A systematic review. Journal of King Saud University - Computer and Information Sciences 35(1), 145–174 (2023), https://www.sciencedirect.com/science/article/pii/S1319157822004062
11. Commission, E., Centre, J.R., Hradec, J., Craglia, M., Di Leo, M., De Nigris, S., Ostlaender, N., Nicholson, N.: Multipurpose synthetic population for policy applications. Publications Office of the European Union (2022)
12. Dhankhad, S., Mohammed, E., Far, B.: Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. In: 2018 IEEE International Conference on Information Reuse and Integration (IRI). pp. 122–125 (2018)
13. Dimou, G.K., Koloniari, G.: Enhancing credit card fraud detection using knowledge graphs and centralities. In: Bădică, C., Gušev, M., Iftene, A., Ivanović, M., Manolopoulos, Y., Xinogalos, S. (eds.) Advances in ICT Research in the Balkans. pp. 14–29. Springer Nature Switzerland, Cham (2025)
14. Ding, Q., Katenka, N., Barford, P., Kolaczyk, E., Crovella, M.: Intrusion as (anti)social communication: characterization and detection. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 886–894 (2012)
15. Dornadula, V.N., Geetha, S.: Credit Card Fraud Detection using Machine Learning Algorithms. Procedia Computer Science 165, 631–641 (2019)
16. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. In: International Conference on Semantic Systems (2016), https://api.semanticscholar.org/CorpusID:8536105
17. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry 40(1), 35–41 (1977), http://www.jstor.org/stable/3033543
18. Freeman, L.C.: Centrality in social networks conceptual clarification. Social Networks 1(3), 215–239 (1978), https://www.sciencedirect.com/science/article/pii/0378873378900217
19. Gupta, P.: Leveraging machine learning and artificial intelligence for fraud prevention. International Journal of Computer Science and Engineering 10(5), 47–52 (5 2023)
20. Hooi, B., Song, H.A., Beutel, A., Shah, N., Shin, K., Faloutsos, C.: Fraudar: Bounding graph fraud in the face of camouflage. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 895–904 (2016)
21. Ikhsan, W., Ednoer, E., Kridantika, W., Firmansyah, A.: Fraud detection automation through data analytics and artificial intelligence. Riset 4, 103–119 (Sep 2022)
22. Kamuangu, P.: A Review on Financial Fraud Detection using AI and Machine Learning. Journal of Economics, Finance and Accounting Studies 6, 67–77 (Feb 2024)
23. Kendall, M.G.: Further contributions to the theory of paired comparisons. Biometrics 11(1), 43 (3 1955), https://doi.org/10.2307/3001479
24. Khanum, A., K S, C., Singh, B., Gomathi, C.: Fraud detection in financial transactions: A machine learning approach vs. rule-based systems. In: 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE). pp. 1–5 (2024)

25. Kleinberg, J.M.: Hubs, authorities, and communities. ACM Comput. Surv. 31(4es), 5–es (1999)
26. Koronaios, A., Koloniari, G.: Graph-based bitcoin fraud detection using variational graph autoencoders and supervised learning. Procedia Computer Science 257, 817–825 (2025), https://www.sciencedirect.com/science/article/pii/S1877050925008427, the 16th International Conference on Ambient Systems, Networks and Technologies Networks (ANT)/ the 8th International Conference on Emerging Data and Industry 4.0 (EDI40)
27. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 420–429. KDD '07, Association for Computing Machinery, New York, NY, USA (2007), https://doi.org/10.1145/1281192.1281239
28. Li, J., Willett, P.: Articlerank: A pagerank-based alternative to numbers of citations for analysing citation networks. Aslib Proceedings 61, 605–618 (11 2009)
29. Lopez-Rojas, E.: Synthetic financial datasets for fraud detection (4 2017), https://www.kaggle.com/datasets/ealaxi/paysim1
30. Lopez-Rojas, E.A., Elmir, A., Axelsson, S.: Paysim: A financial mobile money simulator for fraud detection (2016)
31. Prasad, P.Y., Chowdary, A.S., Bavitha, C., Mounisha, E., Reethika, C.: A comparison study of fraud detection in usage of credit cards using machine learning. In: 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI). pp. 1204–1209 (2023)
32. Raghunathan, T.E.: Synthetic data. Annual Review of Statistics and Its Application 8(Volume 8, 2021), 129–140 (2021), https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-040720-031848
33. Rahman, M.J., Zhu, H.: Predicting accounting fraud using imbalanced ensemble learning classifiers – evidence from China. Accounting & Finance 63(3), 3455–3486 (2023)
34. Rej, M.: Credit card fraud Statistics (2 2025), https://merchantcostconsulting.com/lower-credit-card-processing-fees/credit-card-fraud-statistics/
35. S, M.R., V, N., N, M., B, P., M, R.A.: The impact of machine learning algorithms on credit card fraud detection: A comparative study. In: 2025 International Conference on Visual Analytics and Data Visualization (ICVADV). pp. 1576–1580 (2025)
36. Saleh Hussein, A., Salah Khairy, R., Mohamed Najeeb, S.M., Alrikabi, H.T.: Credit card fraud detection using fuzzy rough nearest neighbor and sequential minimal optimization with logistic regression. International Journal of Interactive Mobile Technologies (iJIM) 15(05), pp. 24–42 (3 2021)
37. Sanchez-Lengeling, B., Reif, E., Pearce, A., Wiltschko, A.B.: A gentle introduction to graph neural networks. Distill 6(9), e33 (2021)
38. Schneider, E.W.: Course modularization applied: The interface system and its implications for sequence control and data analysis. ERIC (10 1973), https://eric.ed.gov/?id=ED088424
39. Singh, A., Singh, A., Aggarwal, A., Chauhan, A.: Design and implementation of different machine learning algorithms for credit card fraud detection. In: 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME). pp. 1–6 (2022)
40. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A.: Credit card fraud detection - machine learning methods. In: 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH). pp. 1–5 (2019)
41. Vejalla, I., Battula, S.P., Kalluri, K., Kalluri, H.K.: Credit card fraud detection using machine learning techniques. In: 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS). pp. 1–4 (2023)

42. Wen, S., Li, J., Zhu, X., Liu, M.: Analysis of financial fraud based on manager knowledge graph. Procedia Computer Science 199, 773–779 (2022)
43. Zaveri, A., Kontokostas, D., Hellmann, S., Umbrich, J., Färber, M., Bartscherer, F., Menne, C., Rettinger, A., Zaveri, A., Kontokostas, D., Hellmann, S., Umbrich, J.: Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. Semant. Web 9(1), 77–129 (Jan 2018), https://doi.org/10.3233/SW-170275
44. Zhang, L., Wu, T., Chen, X., Lu, B., Na, C., Qi, G.: Auto Insurance Knowledge Graph Construction and Its Application to Fraud Detection. In: Proceedings of the 10th International Joint Conference on Knowledge Graphs. pp. 64–70. IJCKG '21, Association for Computing Machinery, New York, NY, USA (2022)
45. Zhao, T., Zhang, X., Wang, S.: Graphsmote: Imbalanced node classification on graphs with graph neural networks. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. p. 833–841. WSDM '21 (2021)

George Konstantinos Dimou is an MSc student at Aristotle University of Thessaloniki and currently works as a Data and AI Consultant at Ernst & Young. He plans to continue his academic journey by pursuing a PhD in Artificial Intelligence.

Georgia Koloniari is an Associate Professor at the University of Macedonia at Thessaloniki, Greece and an ACM Member. She received her PhD in Computer Science from the University of Ioannina, Greece. Her current research interests include graph data management, entity resolution and social network analysis.

Received: April 05, 2025; Accepted: November 20, 2025.