



Contents

Editorial

Guest Editorial: Advances in Intelligent Data, Data Engineering, and Information Systems

Papers

- 893 Landslide Detection Based on Efficient Residual Channel Attention Mechanism Network and Faster R-CNN
Yabing Jin, Ou Ou, Shanwen Wang, Yijun Liu, Haoqing Niu, Xiaopeng Leng
- 911 Tourism Recommendation based on Word Embedding from Card Transaction Data
Minsung Hong, Namho Chung, Chulmo Koo
- 933 Read between the Interactions: Understanding Non-interacted Items for Accurate Multimedia Recommendation
Jiyeon Kim, Taeri Kim, Sang-Wook Kim
- 949 Class Probability Distribution Based Maximum Entropy Model for Classification of Datasets with Sparse Instances
Saravanan Arumugam, Anandhi Damotharan, Srividya Marudhachalam
- 977 Comprehensive Risk Assessment and Analysis of Blockchain Technology Implementation Using Fuzzy Cognitive Mapping
Somayeh Samsamian, Aliakbar Hasani, Saqib Hakak, Fatemeh Esmaeilnezhad Tanha, Muhammad Khurran Khan
- 996 RESNETCNN: an Abnormal Network Traffic Flows Detection Model
Yimin Li, Dezhi Han, Mingming Cui, Fan Yuan, Yachao Zhou
- 1015 Logical dependencies: extraction from the versioning system and usage in key classes detection
Adelina Diana Stana, Ioana Șora
- 1037 A Hierarchical Federated Learning Model with Adaptive Model Parameter Aggregation
Zhuo Chen, Chuan Zhou, Yang Zhou
- 1061 Point of Interest Coverage with Distributed Multi-Unmanned Aerial Vehicles on Dynamic Environment
Fatih Aydemir, Aydin Cetin
- 1085 The Effective Skyline Quantify-utility Patterns Mining Algorithm with Pruning Strategies
Jimmy Ming-Tai Wu, Ranran Li, Pi-Chung Hsu, Mu-En Wu
- 1109 Probabilistic Reasoning for Diagnosis Prediction of Coronavirus Disease based on Probabilistic Ontology
Messaouda Fareh, Ishak Riali, Hafsa Kherbache, Marwa Guemmouz
- 1133 The Proposal of New Ethereum Request for Comments for Supporting Fractional Ownership of Non-Fungible Tokens
Miroslav Stefanović, Đorđe Pržulj, Darko Stefanović, Sonja Ristić, Darko Čapko
- 1157 A Novel Multi-objective Learning-to-rank Method for Software Defect Prediction
Yiji Chen, Lianglin Cao, Li Song

Special Section: Advances in Intelligent Data, Data Engineering, and Information Systems

- 1179 Multi-perspective Approach for Curating and Exploring the History of Climate Change in Latin America within Digital Newspapers
Genoveva Vargas-Solar, José-Luis Zechinelli-Martini, Javier A. Espinosa-Oviedo, Luis M. Vilches-Blázquez
- 1207 Matching Business Process Behavior with Encoding Techniques via Meta-Learning: An anomaly detection study
Gabriel Marques Tavares, Sylvio Barbon Junior
- 1235 A Framework for Privacy-aware and Secure Decentralized Data Storage
Sidra Aslam, Michael Mrissa
- 1263 Detecting and Analyzing Fine-Grained User Roles in Social Media
Johannes Kastner, Peter M. Fischer



Computer Science and Information Systems

Published by ComSIS Consortium

Volume 20, Number 3
June 2023

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

Srdan 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-ENSM, 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*

Z. Budimac, *University of Novi Sad, Serbia*

R. Chbeir, *University Pau and Pays Adour, France*

M.-Y. Chen, *National Cheng Kung University, Tainan, Taiwan*

C. Chesnevar, *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*

V. Filipović, *University of Belgrade, Serbia*

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. Kaščelan, *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. Klačnja Miličević, *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*

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 20, Number 3, 2023
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.mpd.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 2022 two-year impact factor 1.4,
- 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 20, Number 3, June 2023

CONTENTS

Editorial

Guest Editorial: Advances in Intelligent Data, Data Engineering, and Information Systems

Papers

- 893** **Landslide Detection Based on Efficient Residual Channel Attention Mechanism Network and Faster R-CNN**
Yabing Jin, Ou Ou, Shanwen Wang, Yijun Liu, Haoqing Niu, Xiaopeng Leng
- 911** **Tourism Recommendation based on Word Embedding from Card Transaction Data**
Minsung Hong, Namho Chung, Chulmo Koo
- 933** **Read between the Interactions: Understanding Non-interacted Items for Accurate Multimedia Recommendation**
Jiyeon Kim, Taeri Kim, Sang-Wook Kim
- 949** **Class Probability Distribution Based Maximum Entropy Model for Classification of Datasets with Sparse Instances**
Saravanan Arumugam, Anandhi Damotharan, Srividya Marudhachalam
- 977** **Comprehensive Risk Assessment and Analysis of Blockchain Technology Implementation Using Fuzzy Cognitive Mapping**
Somayeh Samsamian, Aliakbar Hasani, Saqib Hakak, Fatemeh Esmailnezhad Tanha, Muhammad Khurran Khan
- 996** **RESNETCNN:an Abnormal Network Traffic Flows Detection Model**
Yimin Li, Dezhi Han, Mingming Cui, Fan Yuan, Yachao Zhou
- 1015** **Logical dependencies: extraction from the versioning system and usage in key classes detection**
Adelina Diana Stana, Ioana Şora
- 1037** **A Hierarchical Federated Learning Model with Adaptive Model Parameter Aggregation**
Zhuo Chen, Chuan Zhou, Yang Zhou
- 1061** **Point of Interest Coverage with Distributed Multi-Unmanned Aerial Vehicles on Dynamic Environment**
Fatih Aydemir, Aydin Cetin
- 1085** **The Effective Skyline Quantify-utility Patterns Mining Algorithm with Pruning Strategies**
Jimmy Ming-Tai Wu, Ranran Li, Pi-Chung Hsu, Mu-En Wu

- 1109 Probabilistic Reasoning for Diagnosis Prediction of Coronavirus Disease based on Probabilistic Ontology**
Messaouda Fareh, Ishak Riali, Hafsa Kherbache, Marwa Guemmouz
- 1133 The Proposal of New Ethereum Request for Comments for Supporting Fractional Ownership of Non-Fungible Tokens**
Miroslav Stefanović, Đorđe Pržulj, Darko Stefanović, Sonja Ristić, Darko Čapko
- 1157 A Novel Multi-objective Learning-to-rank Method for Software Defect Prediction**
Yiji Chen, Lianglin Cao, Li Song

Special Section: Advances in Intelligent Data, Data Engineering, and Information Systems

- 1179 Multi-perspective Approach for Curating and Exploring the History of Climate Change in Latin America within Digital Newspapers**
Genoveva Vargas-Solar, José-Luis Zechinelli-Martini, Javier A. Espinosa-Oviedo, Luis M. Vilches-Blázquez
- 1207 Matching Business Process Behavior with Encoding Techniques via Meta-Learning: An anomaly detection study**
Gabriel Marques Tavares, Sylvio Barbon Junior
- 1235 A Framework for Privacy-aware and Secure Decentralized Data Storage**
Sidra Aslam, Michael Mrissa
- 1263 Detecting and Analyzing Fine-Grained User Roles in Social Media**
Johannes Kastner, Peter M. Fischer

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

In the current third issue of *Computer Science and Information Systems* for 2023, we are happy to announce the impact factors of our journal, updated for 2022: the new two-year IF 1.4, and the five-year IF 1.2. We would like to thank all our authors and reviewers, whose work in their cutting-edge domains continues to increase the impact of our journal. We hope to continue this trend and that the issue in front of you, our dear reader, will offer interesting articles and ideas in both emerging and more established research areas.

This issue consists of 13 regular articles and 4 articles in the special section “Advances in Intelligent Data, Data Engineering, and Information Systems” containing selected and extended versions of papers published in *Proceedings of the 25th European Conference on Advances in Databases and Information Systems (ADBIS), 2021*. We are once again grateful for the hard work and enthusiasm of our authors and reviewers, without whom the current issue, as well as the publication of the journal itself, would not be possible.

In the first regular article, “Landslide Detection Based on Efficient Residual Channel Attention Mechanism Network and Faster R-CNN,” Yabing Jin et al. apply target detection models such as Faster R-CNN to landslide recognition and detection tasks, and propose the Efficient Residual Channel soft thresholding Attention mechanism algorithm (ERCA). ERCA aims to reduce the background noise of images in complex environments by means of adaptive soft thresholding to improve the feature learning capability of deep learning target detection algorithms.

The second regular article, “Tourism Recommendation based on Word Embedding from Card Transaction Data” by Minsung Hong et al. utilize well-known Doc2Vec techniques in the domain of tourism recommendation, using them on non-textual features, card transaction data, to recommend tourism business services to target user groups visiting a specific location, in order to tackle the challenges of missing ratings and spatial factors.

Jiyeon Kim et al., in “Read between the Interactions: Understanding Non-interacted Items for Accurate Multimedia Recommendation” address the problem of multimedia recommendation that additionally utilizes multimedia data, by challenging the common assumption that all the non-interacted items of a user have the same degree of negativity. The authors classify non-interacted items of a user into two kinds – unknown and uninteresting – and propose a novel negative sampling technique that only considers the uninteresting items as candidates for negative samples.

The article “Class Probability Distribution Based Maximum Entropy Model for Classification of Datasets with Sparse Instances” by A. Saravanan et al. proposes a maximum entropy model based on class probability distribution is for classifying sparse data with fewer attributes and instances, introducing a novel way of using Lagrange multipliers for estimating class probabilities in the process of class label prediction.

In “Comprehensive Risk Assessment and Analysis of Blockchain Technology Implementation Using Fuzzy Cognitive Mapping,” Somayeh Samsamian et al. identify and categorize a comprehensive set of risks regarding blockchain implementation. Critical risks are defined by performing a two-stage fuzzy Delphi method based on the experts’ opinions. Then, possible causal relationships between considered risks are identified and analyzed using the fuzzy cognitive mapping method. Finally, the most important risks are ranked based on the degree of prominence and the relationships between them. The methodology is applied to an enterprise resource planning system as a case study.

“RESNETCNN: An Abnormal Network Traffic Flows Detection Model,” by Yimin Li et al., proposes RESNETCCN – an intrusion detection model that fuses residual networks (RESNET) and parallel cross-convolutional neural networks. Benefits of the proposed architecture include more effective learning of data stream features and use of oversampling, which contribute to better detection of abnormal data streams in unbalanced data streams.

Adelina Diana Stana and Ioana Şora, in “Logical Dependencies: Extraction from the Versioning System and Usage in Key Classes Detection” propose a language-independent method to collect and filter dependencies from version control systems, and use it to identify key classes in three software systems. Dependencies extracted from source code are also used, independently and in combination with version-control dependencies. The combination of the two methods offers small improvements to using any single one, and version-control dependencies are shown to be comparable to source-code dependencies.

“A Hierarchical Federated Learning Model with Adaptive Model Parameter Aggregation” authored by Zhuo Chen et al. proposes a newly designed federated learning (FL) framework for the participating nodes with hierarchical associations. In the framework, an adaptive model parameter aggregation algorithm is used to dynamically decide the aggregation strategy according to the state of network connection between nodes in different layers.

In “Point of Interest Coverage with Distributed Multi-Unmanned Aerial Vehicles on Dynamic Environment,” Fatih Aydemir and Aydin Cetin aim to effectively cover points of interest (PoI) in a dynamic environment by modeling a group of unmanned aerial vehicles (UAVs) on the basis of a learning multi-agent system. Agents create an abstract rectangular plane containing the area to be covered, and then decompose the area into grids, learning to locate in a way to maximize the number of PoIs to plan their path.

Jimmy Ming-Tai Wu et al., in “The Effective Skyline Quantify-Utility Patterns Mining Algorithm with Pruning Strategies,” propose two algorithms, FSKYQUP-Miner and FSKYQUP, to efficiently mine skyline quantity-utility patterns (SQUPs). The algorithms are based on the utility-quantity list structure and include an effective pruning strategy which calculates the minimum utility of SQUPs after one scan of the database and prunes undesired items in advance.

In their article “Probabilistic Reasoning for Diagnosis Prediction of Coronavirus Disease based on Probabilistic Ontology,” Messaouda Fareh et al. address the prediction of COVID-19 diagnosis using probabilistic ontologies under the difficulties introduced by randomness and incompleteness of knowledge. The approach begins with constructing the entities, attributes, and relationships of the COVID-19 ontology, by extracting symptoms and risk factors. The probabilistic components of COVID-19 ontology are developed by creating a Multi-Entity Bayesian Network.

“The Proposal of New Ethereum Request for Comments for Supporting Fractional Ownership of Non-Fungible Tokens” by Miroslav Stefanović et al. introduces a new standard for Ethereum blockchains that would support fractional ownership of non-fungible tokens, in order to make blockchain technology applicable to an even wider number of use cases.

Finally, “A Novel Multi-objective Learning-to-rank Method for Software Defect Prediction” authored by Yiji Chen et al. proposes two multi-objective learning-to-rank methods, which are used to search for the optimal linear classifier model and reduce redundant and irrelevant features, for use in software defect prediction within the more general domain of search-based software engineering.

Guest Editorial: Advances in Intelligent Data, Data Engineering, and Information Systems

Mírian Halfeld Ferrari¹, Paolo Ceravolo², Sonja Ristić³, Yaser Jararweh⁴, and Dimitrios Katsaros⁵

¹ Université d'Orléans, INSA CVL, LIFO EA, France

² Università degli Studi di Milano, Italy

³ University of Novi Sad, Serbia

⁴ Duquesne University, USA

⁵ University of Thessaly, Greece

The Special Section on Advances in Intelligent Data, Data Engineering, and Information Systems contains papers selected from the workshops that have been held within the framework of the 25th European Conference on Advances in Databases and Information Systems ADBIS 2021, during August 24–26, 2021, at Tartu, Estonia. ADBIS 2021 conference was aimed at providing a forum where researchers and practitioners in the fields of databases and information systems can interact, exchange ideas and disseminate their accomplishments and visions. Within the scope of the Conference five workshops were held:

- DOING'21: Intelligent Data – from data to knowledge;
- SIMPDA'21: Data-Driven Process Discovery and Analysis;
- MADEISD'21: Modern Approaches in Data Engineering and Information System Design;
- MegaData'21: Advances in Data Systems Management, Engineering, and Analytics; and
- CAoNS'21: Computational Aspects of Network Science.

The authors of the best workshop papers were invited to submit extended versions of their papers in a special section of the journal *Computer Science and Information Systems*. Extended versions of submitted papers went through a rigorous reviewing procedure, the same as for regularly submitted papers. Finally, we accepted four papers presenting both theoretical and practical contributions. In the following, the accepted papers are briefly outlined.

In the first paper “Multi-perspective Approach for Curating and Exploring the History of Climate Change in Latin America within Digital Newspapers,” by the authors Geneveva Vargas-Solar, Jose-Luis Zechinelli-Martini, Javier A. Espinosa-Oviedo, and Luis M. Vilches-Blazquez, an extended description of the Latin American Climate Change Evolution platform called LACLICHEV is proposed. The objective of LACLICHEV is to provide an integrated platform to expose and study meteorological events described in historical newspapers that are possibly related to the history of climate change in Latin America. Exploring the history of climate change through digitalized newspapers published around two centuries ago introduces four challenges: (1) curating content for tracking entries describing meteorological events; (2) processing colloquial language for extracting meteorological events; (3) analyzing newspapers to discover meteorological patterns

possibly associated with climate change; and (4) designing tools for exploring the extracted content. Presented results contribute to data curation and exploration adapted for Spanish textual content within digital newspaper collections. Authors used well-known information retrieval and analytics techniques, within a data exploration environment LACLICHEV that provides tools for curating, exploring, and analyzing historical newspaper articles, their description and location, and the vocabularies used for referring to meteorological events. The platform makes it possible to understand and identify possible patterns and models that can build an empirical and social view of the history of climate change in the Latin American region.

The authors of the second paper entitled “Matching Business Process Behavior with Encoding Techniques via Meta-Learning: An anomaly detection study,” Gabriel Marques Tavares and Sylvio Barbon Jr, focus on the detection of anomalous traces in business process event logs that can diminish an event log’s quality. They combine the representational power of encoding with a Meta-learning strategy to enhance the detection of anomalous traces in event logs towards fitting the best discriminative capability between common and irregular traces. Their approach creates an event log profile and recommends the most suitable encoding technique to increase anomaly detection performance. They used eight encoding techniques from different families, 80 log descriptors, 168 event logs, and six anomaly types for experiments. The presented results indicate that event log characteristics influence the representational capability of encodings. The authors analyzed the influence of meta-features on the recommended encoding technique. This analysis leveraged the understanding of which features better capture process behavior in the context of anomaly detection.

The authors Sidra Aslam and Michael Mrissa in the third paper “A Framework for Privacy-aware and Secure Decentralized Data Storage” present a decentralized data storage and access framework that ensures data security, privacy, and mutability in the wood supply chain scenario. The proposed framework integrates blockchain technology with Distributed Hash Table (DHT), a role-based access control model, and different types of encryption techniques. Their solution allows authorized actors to write, read, delete, update their data and manage transaction history on a decentralized system. The proposed traceability algorithm enables authorized actors to trace the product data in a decentralized ledger. The main limitations of existing solutions are a single point of failure, data mutability, and public availability of the data. The presented prototype design is flexible to expand and can be easily reused for different application domains such as medicine, and agriculture. The security and privacy analysis of the proposed solution is given, as well as the results of the performance evaluation in terms of time cost and scalability. The experimental results have shown that the proposed solution is scalable, secure, and achieves an acceptable time cost.

The authors Johannes Kastner and Peter M. Fischer in the last paper “Detecting and Analyzing Fine-Grained User Roles in Social Media” have proposed a method on how to determine and label user roles in large-scale social media data sets. This largely automated and scalable detection method combines unsupervised learning (more specifically, hierarchical clustering) to discover the classes of users over a wide range of features and supervised learning – generalizing the knowledge from manually labeled smaller data sets. Presented results of the analysis on a range of large data sets from Twitter show that well-separated roles can consistently be recognized and transferred. The labeling achieves

high accuracy not only within the same data set, but also on new data sets from different event types and/or years apart. The approaches scale well with little need for human intervention and the resource requirements of such analyses are modest, bringing them in the range of commodity hardware.

We sincerely thank the workshop organizers for their support in selecting papers and especially the reviewers for their valuable comments to improve selected papers. We also thank all authors for their contribution to this special section. Special thanks are given to prof. Mirjana Ivanović, the Editor in Chief of ComSIS, for providing us the opportunity to publish this special section, valuable comments in improving the quality of selected papers, and support in the whole process.

Landslide Detection Based on Efficient Residual Channel Attention Mechanism Network and Faster R-CNN

Yabing Jin¹, Ou Ou^{2,*}, Shanwen Wang², Yijun Liu¹, Haoqing Niu², and Xiaopeng Leng²

¹ Geological Bureau of Shenzhen, Shenzhen 518028, China
jinyabing25@sina.com

² College of Computer and Network Security, Chengdu University of Technology,
Chengdu 610051, Sichuan, China
ouou@cdut.edu.cn

Abstract. Accurate landslide detection plays an important role in land planning, disaster prediction and disaster relief. At present, field investigation and exploration based on professional personnel is the most widely used landslide mapping and detection technology, but this method consumes a lot of manpower and material resources and is inefficient. With the development of artificial intelligence, landslide identification and target detection based on deep learning have attracted more and more attention due to their remarkable advantages over traditional technologies. It is a technical problem to identify landslides from satellite remote sensing images. Although there are some methods at present, there is still room for improvement in the target detection algorithm of landslides against the background of the diversity and complexity of landslides. In this paper, target detection algorithm models such as Faster R-CNN apply to landslide recognition and detection tasks, and various commonly used recognition and detection algorithm network structures are used as the basic models for landslide recognition. Efficient residual channel soft thresholding attention mechanism algorithm (ERCA) is proposed, which intends to reduce the background noise of images in complex environments by means of deep learning adaptive soft thresholding to improve the feature learning capability of deep learning target detection algorithms. ERCA is added to the backbone network of the target detection algorithm for basic feature extraction to enhance the feature extraction and expression capability of the network. During the experiment ERCA combined with ResNet50, ResNet101 and other backbone networks, the objective indicators of detection results such as AP50 (Average Precision at IOU=0.50), AP75 (Average Precision at IOU=0.75) and AP (Average Precision) were improved, and the AP values were all improved to about 4%, and the final detection results using ResNet101 combined with ERCA as the backbone network reached 76.4% AP value. ERCA and other advanced channel attention networks such as ECA (Efficient Channel Attention for Deep Convolutional Neural Networks) and SENet (Squeeze-and-Excitation Networks) are fused into the backbone network of the target detection algorithm and experimented on the landslide identification detection task, and the detection results are that the objective detection indexes AP50, AP75, AP, etc. are higher for ERCA compared with other channel attention, and the subjective detection image detection effect and feature map visualization display are also better.³

* Corresponding author

³ We released our code at: <https://github.com/fluoritess/Efficient-residual-channel-attention-mechanism-network-and-Faster-R-CNN>.

Keywords: landslide detection, deep learning, Faster R-CNN, ERCA.

1. Introduction

Landslide is a common geological natural disaster, causing serious damage to the natural environment, personal safety and property of all countries. Landslides may be caused by many factors, including earthquake [1, 2], heavy rainfall [3,4], human factors[5], etc. Field investigation of potential landslide areas by professionals is a common and reliable method, but this is time-consuming, expensive and inefficient [6], especially for large-area landslide detection. Due to the above reasons, more and more scholars have started to explore semi-automated or automated landslide detection methods based on remote sensing images in the last decade or so [38].

Remotely sensed images are images acquired from ground observations by aerial aircraft or artificial satellites. Based on the acquisition method, remote sensing images can be classified into the categories of SAR images, infrared images, multispectral images, and visible images [30]. Due to synthetic aperture radar (SAR) images based on microwave coherence imaging have a single color and lack texture detail information; multispectral images have poor resolution and image information is difficult to understand; infrared images are more suitable for identifying heat-emitting targets, visible images become the most commonly used remote sensing image category in landslide detection, and visible images have intuitive content, high resolution, and contain a large amount of information, with rich spatial information, clear geometric structure and texture information, and can truly reflect the ground geographic conditions [31-34]. Therefore, the improved model as well as the chosen dataset in this paper are for remote sensing images in the visible light category. Because most of the landslides are small in scale, large in number and the surrounding environment of the landslide is complex, detecting landslides from remote sensing images is a very challenging problem [7].

At present, there are two main methods for landslide detection in remote sensing images: one is the traditional machine learning-based landslide detection method for remote sensing images [35], which firstly uses two methods, pixel-based method or object-based method [8], to obtain the suspected landslide area in remote sensing images. In the pixel-based landslide detection method, a single pixel in the remote sensing image is the most basic processing unit [10], which determines whether a certain area in the image is a landslide. The object-based landslide detection method calculates the texture and spectral similarity between the pixels in the remote sensing image, clusters a single pixel into multiple candidate objects, and then sets a threshold to classify each candidate object for landslide classification. Then the acquired suspected landslide areas were classified, and the early rule-based classification systems were established mainly relying on the professional judgment of relevant experts on data features [39]. With the rapid development of technology, machine learning has been widely applied to landslide and other geological hazards research, and many machine learning-based landslide classification algorithms have been proposed one after another, such as Stumpf and Kerle [11] implemented object-based landslide detection with random forest (RF), Van Den Eeckhaut et al.[12], which used an object-based method and support vector machine (SVM) to identify landslides in forested areas with LiDAR and its derivatives, etc. The traditional machine learning-based landslide detection method for remote sensing images can explore large landslide

areas in complex contexts and has the advantages of lower cost and faster than field survey methods, but the accuracy of this method relies heavily on the selection of parameters for the classification of candidate landslide images, i.e., the background knowledge of the landslide domain [36].

Another one is a remote sensing image landslide detection method based on convolutional neural network [9]. With the rapid development of deep learning, convolutional neural networks (CNN) can effectively extract key features from image training samples by the two advantages of local perception and parameter sharing, which has become one of the most important feature extraction methods [13,14,15] for image processing tasks such as image classification, target recognition, etc. A convolutional neural network-based landslide detection method for remote sensing images can automatically extract important features of landslide remote sensing images through a multilayer convolution operation [37], thus avoiding the manual feature design and related parameter setting process that requires landslide expertise to perform, and making the landslide detection task more straightforward and simple. Wang [40] used an integrated geographic database to compare the recognition accuracy of five machine learning methods, namely, convolutional neural network, random forest, logistic regression, reinforcement learning, and support vector machine, in identifying landslides in natural terrain, and among the five methods, convolutional neural network had the highest recognition accuracy, while pointing out that recognition techniques based on machine learning and deep learning have excellent It is also pointed out that the recognition techniques based on machine learning and deep learning have excellent robustness and great potential for problem solving in landslide recognition research. Recently, many scholars have proposed several remote sensing image landslide detection methods based on deep learning for convolutional neural networks. Ding [16] proposed to use the traditional convolutional neural network to extract image features to find suspicious areas where landslides occurred, and then confirm these suspicious areas through change detection methods based on image texture features. Because the traditional convolutional neural network has poor characterization ability for the detection object with multiple scales [10], and landslides usually appear at different scales, with the landslide length from several meters to several kilometers [17]. Therefore, Lei [18] et al. proposed a fully convolutional neural network based on pyramid pooling, which can extract feature semantics in remote sensing images more efficiently, and performs better in multi-scale landslide detection.

This paper proposes an Efficient Residual Channel Attention Mechanism Network (ERCA). ERCA intends to improve the feature learning ability of the deep learning target detection algorithm by reducing the background noise of images in complex environments through a deep learning adaptive soft thresholding approach to improve the accuracy of landslide identification detection algorithms in scenarios with complex land cover and uncertainty of light and dark intensity of remote sensing images. ERCA is highly portable and can be easily added to mainstream networks such as ResNet, VGG. The ERCA is integrated into the Faster R-CNN model to improve the model's ability to extract landslide features in remote sensing images. Compared with other current algorithms, the improved algorithm has higher AP values. The algorithm in this paper applies to landslide images taken by remote sensing satellites of common resolution, and the landslide images used in the experiments in this paper are taken by TripleSat satellites.

2. Landslide detection and identification method

In view of the complex surrounding environment and the many types of landslides, this paper presents an Efficient Residual Channel Attention Mechanism Network (ERCA) in order to improve the effect of landslide target detection, and integrates ERCA into The Faster R-CNN model to propose a more significant landslide characterization from the background to improve the detection effect.

2.1. Faster R-CNN network structure

The detection process of the Faster R-CNN algorithm is shown in Figure 1. The process is as follows: 1) perform the feature extraction on the original image through the basic convolutional backbone network (ResNet50[21], VGG16[22], etc.); 2) use the Feature Map extracted in step 1 to generate multiple candidate regions through the RPN network; 3) output a fixed-size feature map through the ROI pooling layer based on the Feature Map extracted in step 1 and the candidate region generated in step 2; 4) classify the categories based on the feature map output in step 3, and perform the frame regression to obtain the precise position of the detection frame. The RPN network is one of the biggest

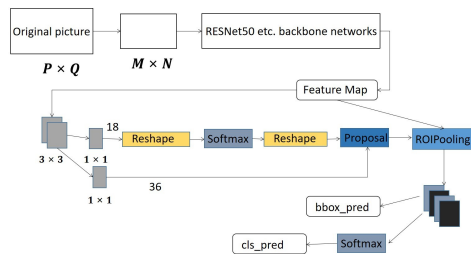


Fig. 1. Faster R-CNN Structure

innovations of the Faster R-CNN algorithm. The previous candidate region extraction methods are usually very time-consuming, such as the SS (Selective Search) algorithm adopted by R-CNN and Fast R-CNN [23] and the Sliding Window algorithm used in traditional target detection. RPN is implemented by a fully convolutional network, which is essentially a classless object detector based on a sliding window. Since RPN can share the convolutional features of the entire image with the detection network, it can output a series of candidate region suggestion frames for input images of any scale at almost no cost.

The structure of the RPN network is shown in Figure 2. First, use the sliding window to generate m anchor points for the center position of each window on the shared feature map as the initial detection frame, and then perform object classification and border regression on the generated anchor points. Since object classification is a two-classification problem, that is, to determine whether the anchor point is the detection target or the background, the object classification obtains $2m$ target scores. Similarly, because the bounding

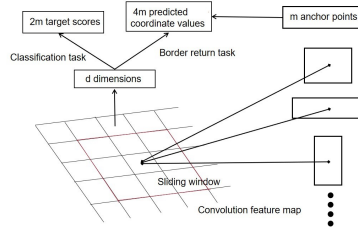


Fig. 2. RPN structure

box regression needs to modify the four coordinate values (x, y, w, h), the bounding box regression obtains 4m predicted coordinate values.

It can be seen from the above that the RPN network is a multi-task network, and its overall loss function is composed of two parts. The equation is as follows:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

The left side of the plus sign (+) in Eq. (1) is the loss value of the classification task, where p_i represents the probability that the current anchor is the target; p_i^* represents the target label value, as in Eq. (2), that is, if the current anchor is a positive sample, its value is 1, otherwise it is 0.

$$P_i^*(x) = \begin{cases} 0 & x \in \text{Negativesamples} \\ 1 & x \in \text{Positivesamples} \end{cases} \quad (2)$$

The loss value function used in the classification task is cross entropy, as shown in Eq. (3).

$$L_{cls}(p_i, p_i^*) = -\log [p_i^* p_i + (1 - p_i^*) (1 - p_i)] \quad (3)$$

The right side of the plus sign (+) in Eq. (1) is the loss value of the bounding box regression task, where $t_i = \{t_x, t_y, t_w, t_h\}$ represents the four predicted coordinate values of the rectangular bounding box; t_i^* represents the four marker coordinate values in the positive sample; the loss function of the bounding box regression task is only considered when p_i^* is 1, if p_i^* is 0, the bounding box regression loss value is also 0, as shown in Eq. (4), in which R is the Smooth L1 function as in Eq. (5).

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (4)$$

$$R(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (5)$$

2.2. Efficient residual channel attention mechanism network (ERCA)

This paper is inspired by the literature [24,25,26] to propose the efficient residual channel attention mechanism network (ERCA) which structure is shown in Figure 3. The ERCA

is implemented through the three structures of 1D convolution, soft threshold and residual network.

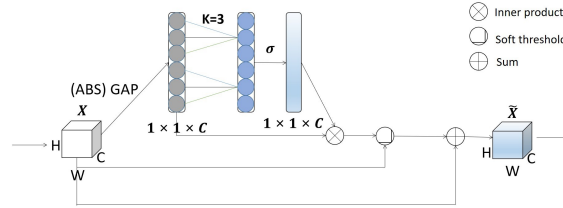


Fig. 3. ERCA Structure

1D convolution Given the normalized feature map of the data, global average pooling (GAP) is performed before 1D convolution [27]. Global average pooling adds up all the pixel values of each channel in the feature map to obtain a value, which is used to represent the feature map of this channel. The feature map for n channels is pooled by global averaging to obtain n values then $X \in R^{H \times W \times C}$ becomes C values. ERCA captures cross-channel attention interaction by considering each channel and its k neighbors. It is realised as a one-dimensional convolution with a k-size convolution kernel captures the attention of neighbors to participate in a channel, in which k represents the coverage of local cross-channel interaction. In order to avoid manual parameter adjustment, the method of reference [24] in this article realizes the automatic learning of k.

$$k = \left\lfloor \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right\rfloor_{odd} \tag{6}$$

In Eq. (6), C is the number of channels, γ, b the constants which are set $\gamma = 1, b = 2$, and $\lfloor \cdot \rfloor_{odd}$ the nearest odd. After the global average pooling and 1D convolution, the activation function σ (sigmoid) is used to activate the final output $V = [v_1, \dots, v_i, \dots, v_c]$, in which v_i is a constant.

Soft threshold A soft threshold is inserted as a non-linear transformation layer in the deep learning network to eliminate unimportant features:

$$\eta_i(x_i, \lambda_i) = \text{sgn}(x_i) (|x_i| - \lambda_i)_+ \tag{7}$$

In Eq. (7), λ_i represents a non-negative threshold. $(|x_i| - \lambda_i)_+$ equals $|x_i| - \lambda_i$ if $(|x_i| - \lambda_i) > 0$, while it equals 0 if $(|x_i| - \lambda_i) < 0$. The soft threshold λ_i in the paper adopts the similar method in reference[25], which presents $\lambda_i = v_i \cdot \text{average}_{i,j,c} |x_{i,j,c}|$. λ_i is the threshold of the feature map of c, where i, j, c are the width, height and current channel, respectively.

Residual Network The final output $\tilde{X} \in R^{H \times W \times C}$ is obtained after the re-assigned feature map is added to the original one through the residual network, as shown in Eq. (8).

$$\tilde{X} = \eta(X, \lambda) + X = [\eta_1 + x_1, \dots, \eta_i + x_i, \dots, \eta_c + x_c] \tag{8}$$

ERCA Module for Deep CNN Networks Figure 3 shows the basic structure of the efficient residual channel attention mechanism network (ERCA). Without dimensionality reduction, the convolutional features are aggregated through the global average pooling operation, and then the cross-channel attention is captured through 1D convolution. The sigmoid function is used to activate the learning channel attention, a soft threshold is inserted as a nonlinear transformation layer to eliminate unimportant features, and finally the residual structure is used for summation. The research adopts the embedding method, replacing the SENet structure with ERCA as embedding ERCA in the CNN network is similar to SENet.

Parameter Analysis The process of the SENet model can be simply described as follows: given a feature map $X \in R^{H \times W \times C}$, the first step is global average pooling (GAP), the weights in SENet are defined as W_1 and W_2 as in Eq. (9), and the model is adaptively adjusted by a fully connected neural network, and the final output of the model is $U \in R^{H \times W \times C}$. The process is as in Eqs. (10-12), where the GAP operation is defined as $F_{GAP}()$ and σ is the activation function.

$$W_1 = \begin{bmatrix} w_{1,1} & \cdots & w_{1,c} \\ \vdots & \ddots & \vdots \\ w_{c,1} & \cdots & w_{c,c} \end{bmatrix}, W_2 = \begin{bmatrix} w_{1,1} & \cdots & w_{1,c} \\ \vdots & \ddots & \vdots \\ w_{c,1} & \cdots & w_{c,c} \end{bmatrix} \tag{9}$$

$$Avg = F_{GAP}(X) \tag{10}$$

$$T = ReLU(Avg * W_1) \tag{11}$$

$$U = X * \sigma(T * W_2) \tag{12}$$

Inspired by the SENet model, the ECA model works with a similar network structure. However, unlike SENet, the ECA model uses 1D convolution to train and acquire the channel-related features, which greatly reduces the parameters of the model. We define the weights of the ECA model as W_3 , and the final output is $U \in R^{H \times W \times C}$ after adaptive adjustment by 1D convolution, as in Eq. (14).

$$W_3 = \begin{bmatrix} w_{1,1} & \cdots & w_{1,k} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & w_{2,2} & \cdots & w_{2,k+1} & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & w_{c,c-k+1} & \cdots & w_{c,c} \end{bmatrix} \tag{13}$$

$$U = X * \sigma(Avg * W_3) \tag{14}$$

Define the weight of the ERCA model in this paper as W_4 , and ERCA first performs the GAP operation consistent with SENet and ECA models, and the output after 1D convolutional adaptive adjustment is used as part of the soft threshold, and the final output is

$U \in R^{H \times W \times C}$, as in Eqs. (16-17).

$$W_4 = \begin{bmatrix} w_{1,1} & \cdots & w_{1,k} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & w_{2,2} & \cdots & w_{2,k+1} & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & w_{c,c-k+1} & \cdots & w_{c,c} \end{bmatrix} \quad (15)$$

$$\lambda = \sigma (Avg * W_4) * F_{GAP}(|X|) \quad (16)$$

$$U = X + \eta(X, \lambda) \quad (17)$$

It is not difficult to find that the model weights W_4 of our algorithm and the weights W_3 of the ECA model are sparser than the weights W_1 and W_2 of the SENet model. We define the number of channels as C , then the parameters of SENet model weights are C^2 , in contrast, our ERCA model is consistent with ECA model with only k parameters.

ERCA Utilizes Both Maximum Pooling Outputs and Average Pooling In this section, the single average pooling operation in ERCA in the previous section is replaced using the maximum pooling output and average pooling output of the shared network. The ERCA above is defined as the standard type and the ERCA using both maximum pooling output and average pooling is defined as ERCAMA. Its specific structure is shown in Figure 4. It is worth noting that although ERCAMA uses maximum pooling output and average

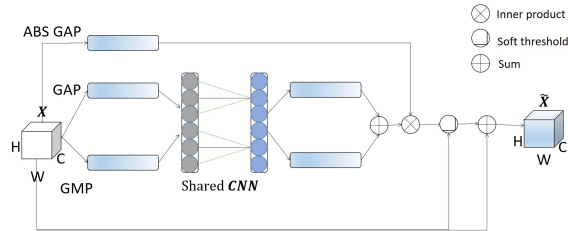


Fig. 4. ERCA Structure

pooling parallel output, the pooling results are convolved using the same set of parameters, so the parametric quantities of ERCAMA and ERCA are the same.

2.3. The Final Structure of the Model

Due to the vast and complex self-made landforms, regional differences, and diverse topography and climate in China, the method of investigating potential landslide areas through professionals is time-consuming, expensive, and inefficient. Meanwhile, with the landslides diverse and complex, it is very important to use artificial intelligence to quickly and accurately extract landslide information from satellite image data. The final detection network model of this research is shown in Figure 5, which mainly includes three parts:

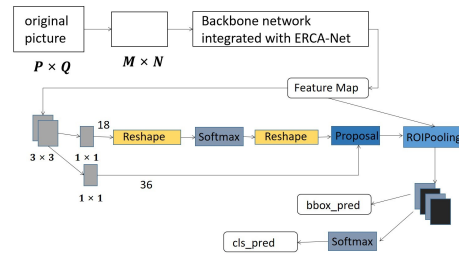


Fig. 5. final model

(1) Feature extraction: Feature Map is obtained by extracting image features through the backbone network integrated into ERCA. Among them, the commonly used backbone networks are ResNet, VGG, etc. ResNet34, ResNet50 and ResNet101 are adopted as the backbone network in the experiments of this research. (2) RPN detection: The recommended target candidate area is obtained through the RPN detection network processing. (3) Object detection and classification: The target classification result is obtained by extracting and processing the Feature Map of the candidate area. Compared with the original Faster R-CNN, the improved algorithm in this paper realizes the channel attention mechanism by adding few parameters, and improves the target detection effect in complex environments.

3. Experimental Process and Analysis

3.1. Experimental Environment and Data Enhancement

This paper adopts python3.7 as the development language, pytorch as the deep learning framework, and Pycharm as the development tool. Graphics card GeForce RTX 2080 Ti is employed with 11G video memory. In order to verify the effectiveness of the algorithm in this paper, we use landslide image in Bijie City[6]. In this experiment, more than 200 high-quality landslide images containing large, medium and small landslides were selected from the landslide images and labeled using the Labeling tool to form a training dataset in PASCAL VOC format⁴. Considering the phenomenon of model overfitting due to deeper network layers and smaller data volume in deep learning and in order to improve the accuracy of the landslide identification detection algorithm under the scenario of complexity of land cover and uncertainty of light and dark intensity of remote sensing images, we expand the dataset by code with data enhancement of the labeled images. The original data volume was expanded by 10 times, i.e. more than 2000 landslide images, by transforming the images left and right, flipping up and down, optical transformation, Gaussian blur, affine transformation and bounding box transformation. The expanded data set is divided into the training set and the test set in an 8:2 manner, and the training set and the verification set are divided in a 9:1 manner in the training set. Adam is adopted in the optimization algorithm. The first 20 epochs are normal training after freezing training,

⁴ <http://host.robots.ox.ac.uk/pascal/VOC/>

with the freezing initial learning rate of 0.0002 and the normal training initial learning rate of 0.00002. After starting training, the learning rate attenuation strategy is adopted, with the attenuation coefficient of 0.94.

3.2. Comparison of Objective Indicators of Detection Effect

The objective comparison indicators selected in this paper is AP (Average Precision) values, which is defined as follows:

$$P = TP / (TP + FP) \quad (18)$$

$$R = TP / (TP + FN) \quad (19)$$

$$AP = \int_0^1 P(R) dR \quad (20)$$

Among them, TP is the number of positive images with correct predictions, FP is the number of positive images with incorrect predictions, FN is the number of negative images with incorrect predictions, P is the precision rate, and R is the recall rate. APs for different IOU thresholds and APs of different sizes for detecting target objects are specifically defined in the Figure 6.

Average Precision(AP)	
$AP(\%)$	AP at IOU= .50 : .05 : .95 (primary challenge metric)
AP_{50}	AP at IOU= .50 (PASCAL VOC metric)
AP_{75}	AP at IOU= .75 (strict metric)
AP Across Scales:	
AP_S	AP for small objects: area < 32 ²
AP_M	AP for medium objects: 32 ² < area < 96 ²
AP_L	AP for large objects: area > 96 ²

Fig. 6. The above 6 metrics are used for charaterzing for the performance of obejct detecor

Ablation Experiment of Faster R-CNN In this chapter, the different components of the model in this article based on the improvement of the original Faster R-CNN are disassembled for ablation experiments. The Faster R-CNN with ResNet34, ResNet50, and ResNet101 as the backbone network is used as the Baseline and compared with the Faster R-CNN with the backbone network of ResNet34+ERCA, ResNet50+ERCA, and ResNet101+ERCA. The objective comparison indexes of the experimental results are shown in Table 1.

It can be drawn from Table 1 that the AP_M, AP_L, AP_{75} and final AP values of the Faster R-CNN using ResNet34+ERCA, ResNet50+ERCA and ResNet101++ERCA as the backbone network have been improved to different degrees compared to the Faster

Table 1. Ablation experiment

Model	AP(%)	AP50	AP75	APS	APM	APL	Params
ResNet34	0.656	0.976	0.827	0.614	0.641	0.678	82.47M
ResNet34+ERCA	0.684	0.988	0.847	0.619	0.667	0.709	82.48M
ResNet34+ERCAMA	0.697	0.988	0.868	0.630	0.693	0.715	82.48M
ResNet50	0.716	0.988	0.881	0.647	0.680	0.764	108.12M
ResNet50+ERCA	0.728	0.988	0.925	0.655	0.703	0.766	108.13M
ResNet50+ERCAMA	0.750	0.988	0.928	0.712	0.729	0.783	108.13M
ResNet101	0.725	0.988	0.903	0.645	0.708	0.758	180.83M
ResNet101+ERCA	0.740	0.988	0.933	0.733	0.724	0.767	180.84M
ResNet101+ERCAMA	0.764	0.985	0.940	0.763	0.749	0.788	180.84M

R-CNN using the original ResNet34, ResNet50 and ResNet101 as the backbone network, and there is only a small increase in the model parameters. The algorithm incorporated into ERCA proved to be improved compared to Baseline, and the ERCA in this paper can improve the original network.

Comparison of Different Channel Attention Models In order to verify the effectiveness of the efficient residual channel attention mechanism network (ERCA) proposed in this paper, three different channel attention mechanisms SENet, ECA, ERCA,ERCAMA are used in the backbone network combined with ResNet34, ResNet50, ResNet101 as the backbone network for comparison in this section. ECA uses the Github source code <https://github.com/BangguWu/ECANet> disclosed by the original author, and SENet uses <https://github.com/moskomule/senet.pytorch>, the warehouse with the highest number of stars in the pytorch version on Github. The objective comparison indexes of the experimental results are shown in Table 2.

Table 2. Ablation experiment

Model	AP(%)	AP50	AP75	APS	APM	APL	Params
ResNet34+ECA	0.663	0.987	0.817	0.542	0.645	0.697	82.48M
ResNet34+SENet	0.661	0.987	0.807	0.651	0.641	0.683	85.67M
ResNet34+ERCA	0.684	0.988	0.847	0.619	0.667	0.709	82.48M
ResNet34+ERCAMA	0.697	0.988	0.868	0.630	0.693	0.715	82.48M
ResNet50+ECA	0.719	0.987	0.879	0.613	0.684	0.765	108.13M
ResNet50+SENet	0.723	0.988	0.913	0.709	0.692	0.765	159.24M
ResNet50+ERCA	0.728	0.988	0.925	0.655	0.703	0.766	108.13M
ResNet50+ERCAMA	0.750	0.988	0.928	0.712	0.729	0.783	108.13M
ResNet101+ECA	0.731	0.988	0.900	0.660	0.717	0.755	180.84M
ResNet101+SENet	0.730	0.988	0.905	0.695	0.710	0.760	277.25M
ResNet101+ERCA	0.740	0.988	0.933	0.733	0.724	0.767	180.84M
ResNet101+ERCAMA	0.764	0.985	0.940	0.763	0.749	0.788	180.84M

It can be seen from Table 2 that the algorithm ERCA in this paper is basically the same as compared to the ECA model parameters, which are greatly reduced compared

to SENet. The performance results of the three different attention mechanism networks on Faster R-CNN with ResNet34 and ResNet50 as the backbone networks show that the ERCA model outperforms the ECA model with the same parameters in all metrics, and the AP_S of ERCA is slightly lower than SENet, but the rest of the metrics are higher or the same than SENet, indicating that ERCA has stronger robustness compared to SENet in detecting objects containing different sizes. The three attention mechanism networks have equal AP_{50} on the Faster R-CNN with ResNet101 as the backbone network, and the remaining metrics ERCA model outperforms the ECA model and SE model. The efficient residual channel attention mechanism in this paper is experimentally proven to have good results.

Comparison with other Target Detection Algorithms In order to quantitatively analyze the detection performance of the Faster-RCNN algorithm after adding ERCA, three classical target detection networks, Faster-RCNN, YOLOv3[28] and YOLOv4[29], are selected for experimental comparison with the algorithm in this paper. From Table 3, we can see that the Faster R-CNN with ECRA is better than YOLOV3 and YOLOV4 except for the AP50 index, which is slightly lower than YOLOV4.

Table 3. Ablation experiment

Model	AP(%)	AP50	AP75	APS	APM	APL
Faster R-CNN	0.725	0.988	0.903	0.645	0.708	0.758
YOLOV3	0.653	0.978	0.811	0.659	0.645	0.666
YOLOV4	0.667	0.990	0.856	0.620	0.660	0.678
Faster R-CNN+ERCA	0.740	0.988	0.933	0.733	0.724	0.767
FasterR-CNN +ERCAMA	0.764	0.985	0.940	0.763	0.749	0.788

3.3. Display of the Subjective Effect Detection

Table 4 shows the subjective detection results of ResNet50+SENet, ResNet50+ECA, ResNet50+ERCA and ResNet50+ERCAMA as the backbone of Faster R-CNN. For the image of a single landslide as Picture 1, the detection results of the three algorithms are basically the same. For images of multiple landslides with complex backgrounds as in Picture 2, all three algorithms show different degrees of misses and misjudgments, among which ResNet50+SENet and ResNet50+ECA show both misses and misjudgments, while ResNet50+ERCA and ResNet50+ERCAMA only shows misses. For images with multiple landslides and a single background as in Picture 3, the detection results of the three algorithms are basically the same with no misses or misjudgments.

3.4. Feature Map Visualization

Table 5 shows the results of the detection using the feature map visualization to verify the validity of the detection results, where the warmer color indicates the higher attention

Table 4. Three model results diagram.














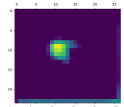
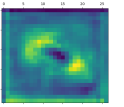
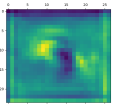
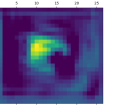
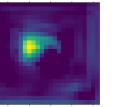

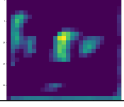
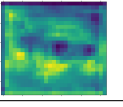
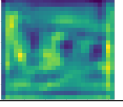
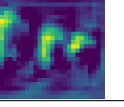
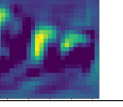
Model	Picture 1	Picture 2	Picture 3
ResNet50+SENet			
ResNet50+ECA			
ResNet50+ERCA			
ResNet50+ERCAMA			

Table 5. Feature map visualization

Original image	ResNet101	ResNet101 +ECA	ResNet101 +SENet	ResNet101 +ERCA	ResNet101 +ERCAMA
					
					

of the deep learning network. From the results in Table 5, it can be seen that the highlighted areas can cover part of the landslide area when no attention is added. And the highlighted areas increase significantly after adding the channel attention mechanisms ECA and SENet respectively, but they also cover many areas that are not landslides, which enhances the landslide features and also enhances part of the redundant features. After adding ERCA and ERCAMA attention mechanisms on the backbone network, the highlighted areas can cover the landslide areas more accurately and comprehensively.

3.5. Application Effect Display

The display site in Yingxiu Town, Wenchuan, in the Sichuan Province, Southwest of China, is selected to present its application. The 18-layer satellite image slices of Yingxiu town were downloaded by bigemap software, and then batch tested based on the program. Some of the test results are as follows.



Fig. 7. Landslide



Fig. 8. Landslide

Figures 7 and 8 show better detection results, but there Figures 9 are still some problems worthy of improvement in actual detection. For example, Figure 8 mistakenly identifies snow mountains as landslides, Figure 10 mistakenly identifies open spaces of human buildings as landslides.

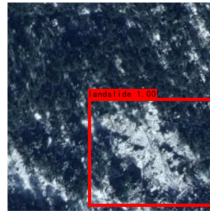


Fig. 9. Snow Mountain



Fig. 10. Open space

In the follow-up practice, more landslide data will be marked, and various terrains and buildings that are easily judged as landslides will be distinguished.

4. Conclusion

This paper uses deep learning Faster R-CNN network for landslide detection research, for the diversity and complexity of landslides this paper proposes an efficient residual channel attention mechanism network (ERCA), ERCA has high portability and can be incorporated into Resnet34, Resnet50, Resnet101 and other networks. It is incorporated with Resnet34, Resnet50, Resnet101, etc. as the backbone network of Faster R-CNN, and the objective index and subjective detection of the algorithm proposed in this paper have achieved good results in landslide image detection experiments, and the experiments prove that the algorithm proposed in this paper has some practicality. The landslide detection model in this paper mainly focuses on different landslide detection in different complex environments. In future work we will use the model in this paper as a sub-model combined with other environmental data such as rainfall, geological conditions, earthquake levels, etc. to build a more comprehensive landslide detection model.

Acknowledgments. This research was funded by the Basic Research Project of Department of Science and Technology of Sichuan Province (NO. 2021YJ0335).

References

1. Roback, K., Clark, M. K., West, A. J., Zekkos, D., Li, G., Gallen, S. F., Godt, J. W.: The size, distribution, and mobility of landslides caused by the 2015 Mw7.8 Gorkha earthquake, Nepal. *Geomorphology*, Vol. 301, 121–138. (2018)

2. Parker, R. N., Densmore, A. L., Rosser, N. J., De Michele, M., Li, Y., Huang, R., Petley, D. N.: Mass wasting triggered by the 2008 Wenchuan earthquake is greater than orogenic growth. *Nature Geoscience*, Vol. 4, No. 7, 449–452. (2011)
3. Mondini, A. C., Guzzetti, F., Reichenbach, P., Rossi, M., Cardinali, M., Ardizzone, F.: Semi-automatic recognition and mapping of rainfall induced shallow landslides using optical satellite images. *Remote Sensing of environment*, Vol. 115, No. 7, 1743–1757. (2011)
4. Hong, Y., Adler, R. F., Huffman, G.: An experimental global prediction system for rainfall-triggered landslides using satellite remote sensing and geospatial datasets. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 45, No. 6, 1671–1680. (2007)
5. Ouyang, C., Zhou, K., Xu, Q., Yin, J., Peng, D., Wang, D., Li, W.: Dynamic analysis and numerical modeling of the 2015 catastrophic landslide of the construction waste landfill at Guangming, Shenzhen, China. *Landslides*, Vol. 14, No. 2, 705–718. (2017)
6. Ji, S., Yu, D., Shen, C., Li, W., Xu, Q.: Landslide detection from an open satellite imagery and digital elevation model dataset using attention boosted convolutional neural networks. *Landslides*, Vol. 17, 1337–1352. (2020)
7. Shi, W., Zhang, M., Ke, H., Fang, X., Zhan, Z., Chen, S.: Landslide Recognition by Deep Convolutional Neural Network and Change Detection. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 59, No. 6, 2020, 4654–4672. (2020)
8. Yu, B., Chen, F., Xu, C.: Landslide detection based on contour-based deep learning framework in case of national scale of Nepal in 2015. *Computers and Geosciences*, Vol. 135, 104388–104388. (2015)
9. Zhang, L., Zhang, L., Du, B.: Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and remote Sensing magazine*, Vol. 4, No. 2, 22–40. (2016)
10. Sameen, M. I., Pradhan, B.: Landslide Detection Using Residual Networks and the Fusion of Spectral and Topographic Information. *IEEE Access*, Vol. 7, 114363–114363. (2019)
11. Stumpf, A., Kerle, N.: Object-oriented mapping of landslides using Random Forests. *Remote sensing of environment*, Vol. 115, No. 10, 2564–2577. (2011)
12. Van Den Eeckhaut, M., Kerle, N., Poesen, J., Hervás, J.: Object-oriented identification of forested landslides with derivatives of single pulse LiDAR data. *Geomorphology*, Vol. 173, 30–42. (2012)
13. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, Vol. 60, No. 6, 84–90. (2017)
14. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision pattern recognition*, 3431–3440. (2015)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision pattern recognition*, 770–778. (2016)
16. Ding, A., Zhang, Q., Zhou, X., Dai, B.: Automatic recognition of landslide based on CNN and texture change detection. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 444–448. (2016)
17. Zhang, Yunling., Fu, Yuhao., Sun, Yu., Zeng, Doudou., Xu, Zeran., Wu, Hangbin.: Combining deep neural networks for landslide detection highway with high-resolution remote sensing images, 188–194. (2021)
18. Lei, T., Zhang, Y., Lv, Z., Li, S., Liu, S., Nandi, A. K.: Landslide inventory mapping from bitemporal images using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, Vol. 16, No. 6, 1–5. (2019)
19. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE conference on computer vision pattern recognition*, 2117–2125. (2017)
20. Singh, B., Najibi, M., Davis, L. S.: SNIPER: Efficient Multi-Scale Training. *Advances in neural information processing systems*, 9310–9320. (2018)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision pattern recognition*, 770–778. (2016)

22. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*. (2014)
23. Uijlings, J. R., Van De Sande, K. E., Gevers, T., Smeulders, A. W.: Selective Search for Object Recognition. *International Journal of Computer Vision*, Voc. 104, No. 2, 154-171. (2013)
24. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11534-11542. (2020)
25. Zhao, M., Zhong, S., Fu, X., Tang, B., Pecht, M.: Deep Residual Shrinkage Networks for Fault Diagnosis. *IEEE Transactions on Industrial Informatics*, Voc. 16, No. 7, 4681-4690. (2019)
26. Hu, J., Shen, L., Sun, G.: Squeeze-and-Excitation Networks. In *Proceedings of the IEEE conference on computer vision pattern recognition*, 7132-7141. (2018)
27. Lin, M., Chen, Q., Yan, S.: Network In Network. *Computer Science*. (2013)
28. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement[J]. *arXiv e-prints*. (2018)
29. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv e-prints*. (2020)
30. Mirghasemi, S., Lotfizad, M.: A target-based color space for sea target detection. *Applied Intelligence*, Voc. 36, No. 4, 960-978. (2012)
31. Li, X., Du, Z., Huang, Y., Tan, Z.: A deep translation (GAN) based change detection network for optical and SAR remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, Voc. 179, 14-34. (2021)
32. Gao, S., Guan, H., Ma, X.: A recognition method of multispectral images of soybean canopies based on neural network, Vol. 68, 101538-101538. (2021)
33. Masouleh, M. K., Shah-Hosseini, R.: Development and evaluation of a deep learning model for real-time ground vehicle semantic segmentation from UAV-based thermal infrared imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, Voc. 155, 172-186. (2019)
34. Xu, D., Zhang, N., Zhang, Y., Li, Z., Zhao, Z., Wang, Y.: Multi-scale unsupervised network for infrared and visible image fusion based on joint attention mechanism. *Infrared Physics and Technology*, Voc. 125, 104242-104242. (2022)
35. Amatya, P., Kirschbaum, D., Stanley, T.: Use of Very High-Resolution Optical Data for Landslide Mapping and Susceptibility Analysis along the Karnali Highway, Nepal. *Remote Sensing*, Voc. 11, No. 19, 2284-2284. (2019)
36. Yu, B., Xu, C., Chen, F., Wang, N., Wang, L.: HADeenNet: A hierarchical-attention multi-scale deconvolution network for landslide detection. *International Journal of Applied Earth Observation and Geoinformation*, Voc. 111, 102853-102853. (2022)
37. Zeng, Q., Geng, J.: Task-specific contrastive learning for few-shot remote sensing image scene classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, Voc. 191, 143-154. (2022)
38. Wang, H., Zhang, L., Wang, L., Fan, R., Zhou, S., Qiang, Y., Peng, M.: Machine learning powered high-resolution co-seismic landslide detection. *Gondwana Research*. (2022)
39. Barlow, J., Martin, Y., Franklin, S. E.: Detecting translational landslide scars using segmentation of Landsat ETM+ and DEM data in the northern Cascade Mountains, British Columbia. *Canadian journal of remote sensing*, Voc. 29, No. 4, 510-517.(2003)
40. Wang, H., Zhang, L., Yin, K., Luo, H., Li, J.: Landslide identification using machine learning. *Geoscience Frontiers*, Voc. 12, No. 1, 351 - 364. (2021)

Yabing Jin is a professor who works for the Shenzhen Geological Bureau and an expert in geological hazard-related research.

Ou Ou received the Ph.D. degree from Chengdu University of Technology, Chengdu, China, in 2015. He is a professor in the School of Computer and Network Security of

Chengdu University of Technology. His current research interests are artificial intelligence and big data in geology.

Shanwen Wang received the B.S. degree in Software Engineering from the School of Information Science and Technology, Chengdu University, Chengdu, China, in 2020. He is pursuing the M.S. degree at Chengdu University of Technology. His current research interests are deep learning and computer vision.

Yijun Liu works at the Shenzhen Geological Bureau. His research interests are the area of geological hazard monitoring and prevention.

Haoqing Niu received the B.S. degree in Internet of Things Engineering from the School of Information Engineering, DaLian University, DaLian, China, in 2020. He is pursuing the M.S. degree at Chengdu University of Technology. His current research interests are small target detection.

Xiaopeng Leng received the Ph.D. degree from Chengdu University of Technology. He is an associate professor in the School of Computer and Network Security at Chengdu University of Technology. His research interests are directed towards the application of artificial intelligence in geology.

Received: August 31, 2022; Accepted: December 25, 2022.

Tourism Recommendation based on Word Embedding from Card Transaction Data

Minsung Hong¹, Namho Chung^{2*}, and Chulmo Koo²

¹ Smart Tourism Research Center, Kyung-Hee University
26-6, Kyunghedae-ro, Dongdaemun-gu, Seoul, South Korea
mshong.res@gmail.com

² Smart Tourism Education Platform, Kyung-Hee University
26-6, Kyunghedae-ro, Dongdaemun-gu, Seoul, South Korea
{nhchung, helmetgu}@khu.ac.kr

Abstract. In the tourism industry, millions of card transactions generate a massive volume of big data. The card transactions eventually reflect customers' consumption behaviors and patterns. Additionally, recommender systems that incorporate users' personal preferences and consumption is an important subject of smart tourism. However, challenges exist such as handling the absence of rating data and considering spatial factor that significantly affects recommendation performance. This paper applies well-known Doc2Vec techniques to the tourism recommendation. We use them on non-textual features, card transaction dataset, to recommend tourism business services to target user groups who visit a specific location while addressing the challenges above. For the experiments, a card transaction dataset among eight years from Shinhan, which is one of the major card companies in the Republic of Korea, is used. The results demonstrate that the use of vector space representations trained by the Doc2Vec techniques considering spatial information is promising for tourism recommendations.

Keywords: recommender system, word embedding, neural networks, smart tourism

1. Introduction

The information available on the Internet is tremendously and rapidly growing in the big data era [15]. Although it is helpful to people in general, users need a lot of energy and time to find useful information. Therefore, recommender systems have been extensively studied and developed in various domains to provide personalized information such as items, content, and services [3]. In the tourism domain, such systems automatically track tourists' preferences from their explicit or implicit feedback and match the features of tourism items with their needs [6,12]. However, the massive amount of the data available is mainly implicit, such as card payment, sensor, and mobile data, for tourism recommendations [5,17]. Accordingly, it is essential to analyze and apply implicit feedback to tourism recommendations [18]. There is also a data sparsity problem that affects negatively recommendation performance, since it is impossible that tourists generally utilize most tourism items. In addition to these, it is important to properly reflect a location factor (spatial information) in tourism recommender systems [11,16].

* Corresponding author

As stated by [2], a credit or debit card is one of the easiest payment methods in the tourism industry, as confirmed by the increasing number of operators that have adopted card payments. Therefore, many studies [29,11,10] have recently used card transaction data to recommend items for tourists. However, despite using various security techniques, using raw card payment data in the previous study might not be realistic in terms of GDPR (General Data Protection Regulation). That is, the card transaction logs contain a lot of identifiable personal data and make identifying a specific user possible [28].

In this paper, we propose novel recommendation methods based on card transaction data to recommend tourism services to user groups visiting specific tourist locations. The data was statistically processed to protect personal information by a data provider. To avoid the absence of rating scores in the dataset, we model the card transaction data to transform users and items into vector representations using neural network-based word embedding techniques (i.e., Doc2Vec). The vector representations are then used in the content, collaborative filtering, or hybrid-based recommendation algorithm to provide appropriate tourism business services to a user group when they visit a specific destination. Experimental results with around twenty-million statistical card transaction data occurred in Jeju island, one of the most famous tourist attractions in the Republic of Korea, for eight years, demonstrate that the proposed recommendation methods superior to other baseline methods. In particular, several experiments show the positive influences of spatial information on recommendation performance by comparing it with other baseline methods, which are difficult to consider the information in their data modeling. In this regard, our contributions are three-fold as follows:

- We propose competent and serviceable recommendation methods for traveler groups despite the limitation of card transaction data that are statistically processed to protect personal information. Also, they outperform other baseline methods in experiments with real-world huge card transaction data.
- We address the absence of rating scores by introducing Doc2Vec techniques without a specific method. Compared with other baseline approaches based on the RFM method of converting transaction data to rating scores, the proposed methods have better performance even on the evaluation methodology that could be favorable to the approaches.
- In our methods, it is competent to model the preferences of user groups and spatial information simultaneously. Also, it positively influences on recommendation performance, as demonstrated by comparing the methods with recommendation approaches based on Word2Vec techniques.

It is worth mentioning that the proposed methods can be directly applied to raw card transaction data for recommending items to individuals.

2. Related work

2.1. Tourism recommender systems

The tourism industry has grown on a large scale in the past decades, and numerous tourist services have been provided physically and virtually. However, the more significant number of service providers, the more difficult it is to identify and select a suitable tourist

item. To reduce the efforts, tourists need to find a tourist item appropriate to their interests. Recommender systems provide items by analyzing tourists' preferences to help them [8,14,9]. In the literature, there are four base recommendation approaches in the tourism industry: content-based, collaborative filtering, domain-specific, and hybrid approaches [22,8]. The content-based approach uses the features of items and users and calculates their similarities to make recommendations. The collaborative filtering approach uses users' past preferences who share similar interests to decide which items to recommend [22]. The domain-specific approach uses various additional information to enhance recommendations such as context, time-sensitive, location, social information, etc [8]. The hybrid method combines these approaches to overcome drawbacks and achieve high recommendation performance [22].

Al-Ghossein et al. [1] proposed a cross-domain recommender system to address the sparsity problem in hotel recommendations. Their basic idea is that users generally select a destination to visit and then look for a hotel. Therefore, their system considers location-based social networks to learn mobility patterns from hotel check-in data and uses the patterns to recommend hotels. To do this, the authors map items and users from both domains based on a number of observations and learn preferences for regions and hotels. The results are then combined to perform the final recommendation using Bayesian personalized ranking. Hong and Jung [16] developed a multi-criteria recommendation method to recommend restaurants. They consider tourists' nationality as spatial information. A tensor model, which keeps the correlation between its dimensional factors, is exploited to simultaneously model user preferences for multi-criteria, spatial and temporal information. Higher Order Singular Value Decomposition (HOSVD) predicts multiple ratings depending on the spatial and temporal information. The authors analyzed the influences of multiple ratings as well as spatial and temporal information on recommendation performance and revealed the positive efficacy of the factors. A framework, namely, filter-first, tour-second (FFTS), was proposed for addressing complex selection and producing recommendations on a multi-period personalized tour [19]. It considers mandatory Points of Interest (POIs) as well as optional points that tourists optionally visit. The optional points are filtered using an item-based collaborative filtering approach and users' online data. And then, the daily tours are built based on an iterated tabu search algorithm. Pessemier et al. [27] developed the hybrid approach that combines a content-based method handling sparse data, collaborative filtering introducing serendipity, and a knowledge-based approach for pre-filtering, in order to recommend tourist destinations to user groups by aggregating individual recommendations. The authors adopted users' rating profiles, personal interests, and specific demands to provide next destinations. Casillo et al. [7] proposed a knowledge-based approach to search for and recommend tourism services within a knowledge base, which are generated by considering user, context, and service, to individuals. The platform consists of three different points of view, such as the representation of the context, data management & organization, and inferential engines. An oriented and labeled graph model for the representation of Web resources was used.

Unlike the related work mentioned above, except for the last work [7], our approach recommends tourism business services to a user group who visits a specific location. To the best of our knowledge, there were few studies to recommend tourist services in the tourism domain. Similar to the tensor model above-mentioned, we reflect spatial information into modeling user preferences at the same time by using the Doc2Vec technique and

consider it in recommendation procedures (i.e., content-based, collaborative filtering, and hybrid approaches).

2.2. Recommendation with word embedding

In the above-mentioned four approaches, there are various methods to make a recommendation, such as a matrix factorization and neural networks. The matrix factorization-based method has been generally applied to real-world recommendation applications due to its high performance, while recently neural networks-based recommender systems have gained considerable interest by overcoming obstacles of conventional models and achieving high recommendation quality [30]. Similar to matrix factorization methods, neural networks-based word embedding techniques from the field of natural language processing field learn low-dimensional vector space representation of input elements [25]. The word embedding learns linguistic regularities and semantics from the sentences and represents the words by vectorized representations [24]. Recently, some of the recommendation methods [24,26,4] used techniques from Word2Vec to represent text-based features, and some of the recommendation algorithms [25,13] applied the techniques to represent items.

Musto et al. [24] empirically compared three word embedding techniques such as Latent Semantic Indexing, Random Indexing, and Word2Vec, on a content-based recommendation. Authors evaluate their methods on MovieLens and DBbook datasets. They map items to textual contents using Wikipedia and use the texts to make recommendations. Also, they aggregate the document representation of items a user liked for generating the user's profile. By exploiting classic similarity measures, available items are ranked according to their descending similarity with respect to the user profile, and top- k items are provided. Baek and Chung [4] developed a multimedia recommendation method using Word2Vec-based social relationship mining. They extract sentiment words from the metadata of multimedia content in TMDb (The Movie Database) and the users' social stream comments about movies. The words are classified through SVM (Support Vector Machine), and Word2Vec techniques are then applied to represent sentiment words into quantifiable vectors. The vector representations of words are used to find a social relationship. They also establish a similarity and trust relationship between users in order to the precise and reliable recommendation of content fitting a user's tendency. Ozsoy [25] also applied the well-known techniques of Word2Vec to recommendation systems. Unlike the above-mentioned work that directly apply the Word2Vec techniques on the textual contents to recommend items, the author uses the techniques to model non-textual contents, the check-ins, for venue recommendation to individual users. In order to model user preferences into a continuous vector representation, the item list in users' visit history is taken sentences into account. Three recommendation algorithms are proposed based on similarities between users and items from the vector representations for users and are evaluated with the Checkin2011 dataset. Esmeli et al. [13] proposed a session-based recommendation using Word2Vec to recommend products. They create product sequences by different session positions and apply the skip-gram method of Word2Vec techniques to calculate similarities between products. Also, they use class imbalance techniques (i.e., synthetic minority over-sampling and under-sampling) to obtain better recommendation performance. They evaluate the proposed method on the RetailRocket dataset.

Like the last two studies, we also consider the lists of items (i.e., tourism business services in card payment transaction data) as sentences to calculate similarities between user groups and the items. In this study, to recommend appropriate services to a user group that visits a specific location, we use the Doc2Vec techniques to reflect spatial information (i.e., tourist destinations) and consider the spatial information in the procedure of making a recommendation list also. The next sections explain how to model card payment transaction data using the techniques and use the trained model in the recommendation process.

3. Modeling card transaction with Doc2Vec

Our objective is to provide top- k tourism services for which a target user group prefers to expenditure, when the group visits a specific destination. In this paper, we use the Doc2Vec techniques namely PV-DM (Distributed Memory version of Paragraph Vector) and PV-DBOW (Distributed Bag of Words version of Paragraph Vector). We used them since they are the primary and initial Doc2Vec techniques. This section briefly introduces the techniques and explains how to model the card transaction data using the methods to achieve the objective.

Doc2Vec techniques were proposed by Mikilov and Le in [20] to create a numeric document representation, regardless of length. It extends the Word2Vec techniques introduced by [23] to go beyond word level to achieve phrase-level or sentence-level representations. It contains two techniques that produce distributed word representations (i.e., word embedding). The representation expresses a word in low dimensional space and carries the semantic and syntactic information of terms and documents [21]. As shown

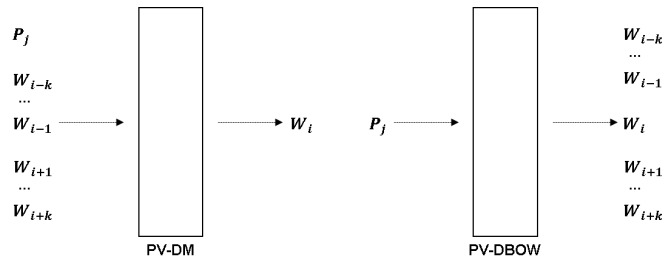


Fig. 1. Doc2Vec techniques

in Fig. 1, the PV-DM technique considers the concatenation of the paragraph vector with the word vectors to predict the next word in a text window, and it is similar to the CBOW (Continuous Bag Of Words) of Word2Vec. While, PV-DBOW predicts the words in a small window, like the skip-gram technique of Word2Vec. The latter one is faster and consumes less memory since there is no need to save the word vectors [20].

We use the Doc2Vec techniques to model card payment transaction data and propose three recommendation methods based on the models trained by the techniques. Therefore, our approach consists of the following two steps. First, the card transaction history

is modeled using the Doc2Vec techniques to represent user groups, business services, and locations as numeric vector representations. The outputs are then used to recommend tourism business services to a user group when they visit a specific destination. This section explains the first step. We use the Doc2Vec techniques implemented in the Gensim toolbox ³. It creates an internal dictionary that holds words and their frequencies, and trains a model using the input data and the dictionary. Its outputs are the vector representations of words and paragraphs. In this paper, the vector representations are considered as the features of user groups, services, and locations.

To model the service usage history of user groups from card transaction data, we generate a list of tourism business services that a user group has used in a specific destination. Destination information is added as a document tag to the list of sentences consisting of words. Therefore, the input data for Doc2Vec techniques indicate the documents, tagged by locations, containing the service usage history of user groups. In Fig. 2, the similarity of input data in the Doc2Vec techniques and recommendation systems is presented conceptually. Fig. 2a presents four sentences in two documents together with the vocabulary

<p>d0, s0: "King walked his cat" d0, s1: "King took his cat to garden" d1, s2: "Queen went to garden" d1, s3: "Queen walked her cat"</p> <p>Vocabulary: {king: w0, walked: w1, his: w2, cat: w3, took: w4, to: w5, garden: w6, queen: w7, went: w8, her: w9}</p> <p>Conceptual vector: s0: [1, 1, 1, 1, 0, 0, 0, 0, 0, 0] s1: [1, 0, 1, 1, 1, 1, 1, 0, 0, 0] s2: [0, 0, 0, 0, 0, 1, 1, 1, 1, 0] s3: [0, 1, 0, 1, 0, 0, 0, 1, 0, 1] d0: [1, 1, 0, 0] d1: [0, 0, 1, 1]</p> <p>Input data: s0: [s0, w0, w1, w2, w3, d0] s1: [s1, w0, w4, w2, w3, w5, w6, d0] s2: [s2, w7, w8, w5, w6, d1] s3: [s3, w7, w1, w9, w3, d1]</p> <p>Output data: d0: [f0, f1, ..., fn], d1: [f0, f1, ..., fn] s0: [f0, f1, ..., fn], s1: [f0, f1, ..., fn] s2: [f0, f1, ..., fn], s3: [f0, f1, ..., fn] w0: [f0, f1, ..., fn], w1: [f0, f1, ..., fn] w2: [f0, f1, ..., fn], ... w9: [f0, f1, ..., fn]</p>	<p>loc0, user0: ser0, ser1, ser2, ser3, ser4, ... loc0, user1: ser1, ser5, ser6, ser9, ... loc1, user2: ser4, ser7, ser8, ... loc1, user3: ser2, ser3, ser5, ser7, ser9, ...</p> <p>Vocabulary: {ser0, ser1, ser2, ser3, ser4, ser5, ser6, ser7, ser8, ser9, ...}</p> <p>Conceptual Vector: user0: [1, 1, 1, 1, 1, 0, 0, 0, 0, 0] user1: [0, 1, 0, 0, 0, 1, 1, 0, 0, 1] user2: [0, 0, 0, 0, 1, 0, 0, 1, 1, 0] user3: [0, 0, 1, 1, 0, 1, 0, 1, 0, 1] loc0: [1, 1, 0, 0] loc1: [0, 0, 1, 1]</p> <p>Input data: user0: [u0, s0, s1, s2, s3, s4, ..., l0] user1: [u1, s1, s5, s6, s9, ..., l0] user2: [u2, s4, s7, s8, ..., l1] user3: [u3, s2, s3, s5, s7, s9, ..., l1]</p> <p>Output data: l0: [f0, f1, ..., fn], l1: [f0, f1, ..., fn] u0: [f0, f1, ..., fn], u1: [f0, f1, ..., fn] u2: [f0, f1, ..., fn], u3: [f0, f1, ..., fn] s0: [f0, f1, ..., fn], s1: [f0, f1, ..., fn] s2: [f0, f1, ..., fn], ...</p>
(a) Document-Sentence-word data	(b) Location-user-service data

Fig. 2. Data examples for Doc2Vec and recommendation

list (dictionary). Similarly, four user groups and the lists of business services, which the groups paid at two specific destinations, are presented in Fig. 2b. Similar to the example in the left figure, it is possible to create a list of services for a user group that visited a specific location. Consequently, both examples are represented as vectors started with sentence and user group identifications followed by word or service ones. The service order is equal to the usage sequence of a user group. Each vector as an input one is added a corresponding document or location tag. Inspiring from [25], the lists of items (i.e.,

³ <https://radimrehurek.com/gensim/models/doc2vec.html>

services), user groups, and locations are used together as the input data of Doc2Vec techniques. As a result, their vectors are obtained separately and are able to be utilized to decide on which element (i.e., a service, user, or destination) is contextually closer to which elements. Accordingly, documents are abstractly separated into sentence and user group levels. Consequently, input data for the Doc2Vec techniques are constructed by sentences and user groups. Finally, elements' representation vectors trained by Doc2Vec techniques contain n real numbers as shown at the bottom of Fig. 2. Fig. 3 presents the output of PV-DM technique on the data example given in the above figure. To plot, the output vector representations with n dimension (i.e., the feature parameter above-mentioned) were converted into two dimensions using principal component analysis. In Fig. 3a, the output

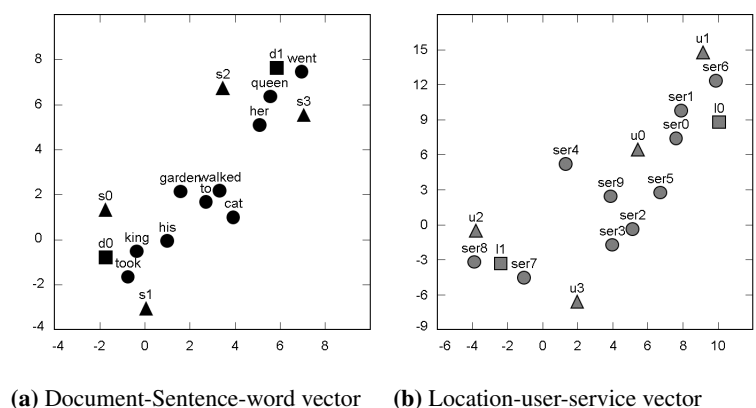


Fig. 3. Vector representation of data examples for Doc2Vec and recommendation

for document-sentence-word data is shown. According to this figure, the relations among documents, sentences, and words are captured. Note that we add the sentence IDs at the first position of input data to obtain the relations of the sentences with other elements, as shown in Fig. 2. For instance, while the “king” and “his” are closer to the document “d0” and sentences “s0” and “s1”, the word “queen”, “her”, and “went” are closer (i.e., more related) to the document “d1” and sentences “s2” and “s3”. Remind that these words are seen only in these each document and corresponding sentences. The “walked”, “to”, and “garden” are closer (more related) to each other and located in the middle of both documents since they appear in the documents. These results indicate that the PV-DM technique is able to capture the relations between these documents, sentences, and words. In Fig. 3b, the output for location-user-service data is presented. From the figure, relations among the elements are able to be observed. For example, services “ser0” and “ser6” are represented closer to user groups “u0” and “u1” respectively and located nearby location “l0”. The user groups utilize the “ser0” and “ser6” at the location “l0”. Other examples are the relations among services. The “ser2” and “ser3” are closer and are always used together in the input data, even in different locations.

4. Recommendation using vector representation

The Doc2Vec techniques provide elements in vector space where similar elements are located closer to each other, as presented in the previous section. We apply the results to three recommendation algorithms.

K-nearest business-based method (KNB) belongs to the content-based recommendation approach. In the traditional approach, item features are used to recommend other items similar to what a user likes. Therefore, we calculate the similarity between a target user group, a specific destination, and services using the vector representations resulting from Doc2Vec. Cosine similarity allowed by the Doc2Vec of the Gensim library is used. As a result, the most similar k services to the target user group and the destination are recommended to the group. Algorithm 1 presents these processes. For instance, given the

Algorithm 1: K-nearest business-based recommendation

Data: Vector representations for services S , a user group U_i , and a destination L_j

Result: List I of top- K services

- 1 Set K for # of recommendation
 - 2 Initialize an empty lists I with the length K and I_s with the length S
 - 3 Calculate a simple mean T_{ij} of the projection weight vectors of the U_i and L_j
 - 4 **for** $k = 1$ to the length of S **do**
 - 5 | Calculate cosine similarity cos_{ijk} between S_k and T_{ij} and append it into the I_s
 - 6 **end**
 - 7 Sort the list I_s in descending order with keeping indexes
 - 8 Put the K services corresponding first K elements' indexes from the I_s into the I
 - 9 Return I
-

vectors presented in Fig. 3b, assume that we want to offer two services that are not used by the user group “u0”. The most similar services to the user group are “ser5” and “ser9” except for the services already used by the group, so these services are recommended to the target user group. Note that the services used by a user group in past are provided in real, since a user group often uses a business service again.

N-nearest users method (NNU) applies the traditional user-based collaborative filtering approach to the vector representations modeled in the previous step. In the traditional approach, first, the most similar users (neighbors) to a target user are found, and the items preferred by the neighbors are provided to the target user. Similar to the approach, we first decide on N neighbors using the similarities among vector representations of a target user group and a specific location (i.e., similar neighbors visited the location). The services previously used/preferred by the neighbors are then collected. Finally, top- k services are selected to recommend by summing up the neighbor votes, as shown in Algorithm 2. For example, using the example presented in Fig. 2b, assume that we want to recommend two services to the user group “u3” visited at the location “l0”, by using two neighbor groups. According to vector representations in Fig. 3b, the most similar user groups, “u0” and “u1”, are selected as the neighbors. The service “ser1” previously visited by both of the groups and another service chosen randomly among the services utilized by “u0” or “u1”

(i.e., “ser0”, “ser2”, “ser3”, “ser4”, “ser5”, “ser6”, and “ser9”) are recommended to the target user group “u3” located at the “10”.

Algorithm 2: N-nearest users-based recommendation

Data: Vector representations of user groups U and a destination L_j , use history \mathcal{H}

Result: List I of top- n services

```

/* Define variables */
1 Set  $N$  for # of neighbor groups and  $K$  for # of recommendation
2 Initialize an empty lists  $NU$  with the length  $N$  and  $NU_s$  with the length  $U$ 
3 Initialize an empty list  $I$  with the length  $K$ 
4 Calculate a simple mean vector  $T_{ij}$  of the target user group  $U_i$  and  $L_j$ 
/* Calculate cosine similarity for neighbor groups */
5 for  $k = 1$  to the length of  $U$  do
6   | if  $k \neq i$  then
7     | Calculate cosine similarity  $cos_{ijk}$  between  $U_k$  and  $T_{ij}$ 
8     | Append the  $cos_{ijk}$  into the list  $NU_s$ 
9   | end
10 end
/* Make top- $N$  neighbor group list */
11 Sort the list  $NU_s$  in descending order with keeping indexes
12 Put the first  $N$  elements' indexes from the  $NU_s$  into the  $NU$ 
/* Collect services used by the neighbor groups */
13 for  $k = 1$  to  $N$  do
14   | Append services used by  $NU_k$  from  $\mathcal{H}$  into service pool list  $Itemp$ 
15 end
/* Get top- $K$  services by summing up the votes of the neighbor
groups */
16 Sort the list  $Itemp$  by service frequency in descending order and remove duplicates
17 Put the first  $K$  services from the  $Itemp$  into the  $I$ 
18 Return  $I$ 

```

N-nearest users and k-nearest business method (NKB) is a hybrid method of the previous two methods. In NKB, N neighbor groups are first found by using the vector representations of a target user group and a specific location. And then, we search for the top- k services that are the most similar to the combination of the user groups, which consist of the target group and the neighbor groups, and the location. The collected top- k services are recommended to the target user group visited at the location, as shown in Algorithm 3. For example, assume that we want to recommend three services to the user group “u0” visited at the location “10” using a single neighbor. The user group “u1” would be selected as the neighbor based on the vector similarity. The three most similar services to the user groups “u0” and “u1” as well as location “10” are “ser0”, “ser1”, and “ser6”. These three services are provided to the target user group “u0” visited the location “10” by the NKB method.

Our methods can handle the cold-start problem for new user groups that have never used any services in our system since the Doc2Vec techniques also result in vector representations of locations, as shown in Fig. 3b. For instance, when a new user group requests

to recommend services in a specific location, our methods can find services with the most similar vector representations to the location's vector representation or search for neighbor groups based on their vector representations.

Algorithm 3: N-nearest and k-nearest business-based recommendation

Data: Vector representations of user groups U and a destination L_j , use history \mathcal{H}
Result: List I of top- n services

```

/* Define variables */
1 Set  $N$  for # of neighbor groups
2 Set  $K$  for # of recommendation
3 Initialize an empty lists  $NU$  and  $NU_s$  with the length  $N$ 
4 Initialize an empty lists  $I$  with the length  $K$  and  $Is$ 
5 Calculate a simple mean vector  $T_{ij}$  of the target user group  $U_i$  and  $L_j$ 
/* Calculate cosine similarity for neighbor groups */
6 for  $k = 1$  to the length of  $U$  do
7   | if  $k \neq i$  then
8     | Calculate cosine similarity  $cos_{ijk}$  between  $U_k$  and  $T_{ij}$ 
9     | Append the  $cos_{ijk}$  into the list  $NU_s$ 
10  | end
11 end
/* Make top- $N+1$  neighbor group list including the target user group
*/
12 Sort the list  $NU_s$  in descending order with keeping indexes
13 Put the vectors corresponding first  $N$  elements' indexes from the  $NU_s$  into the  $NU$ 
14 Put the vectors of user group  $U_i$  into the neighbor list  $NU$ 
15 Calculate a simple mean vector  $T_j$  of the user groups in  $NU$  and location  $L_j$ 
/* Calculate cosine similarity for services */
16 for  $m = 1$  to the length of  $S$  do
17   | Calculate cosine similarity  $cos_{jm}$  between  $S_m$  and  $T_j$ 
18   | Append the  $cos_{jm}$  into the list  $Is$ 
19 end
/* Get top- $K$  services based on vector similarity */
20 Sort the list  $Is$  in descending order with keeping indexes
21 Put the  $K$  services corresponding first  $K$  elements' indexes from the  $Is$  into the  $I$ 
22 Return  $I$ 

```

5. Experimental design

5.1. Dataset

To evaluate the proposed recommendation methods, we use a transaction dataset of credit and debit cards from Shinhan card, one of the major card companies in the Republic of Korea. The dataset consists of transaction logs that happened on Jeju island, one of the most famous tourist attractions in the country. It contains 19,648,116 card transactions. As mentioned above, all identifiable personal data were statistically processed to make them

anonymous by considering GDPR. Accordingly, there are 1,260 user groups categorized by gender, age groups, habitation cities, nationalities, and time periods of card usage, as listed in Table 1. Tourism services categorized by KSIC (Korea Standard Industry Code⁴), based on the International Standard Industrial Classification (ISIC) adopted by the UN, are taken into account. Thereby, 413 services related to the tourism domain are selected, such as retail, wholesale, accommodation, restaurant, and transport businesses. Also, we have 79 destinations since all card transactions in the dataset happened in Jeju and Seogwipo tour cities. Finally, we use around fifteen million transaction data and split it by an 80-10-10 ratio for training, validation, and test sets, respectively. For the recommendation methods based on the techniques of Word2Vec and Doc2Vec, we obtained 60,410 and 47,847 sentences as training and validation sets.

Table 1. Statistic information of preprocessed dataset

Feature	Number Feature	Number
# of transaction group	14,673,210	# of user groups 1,260
# of training set	11,738,568	# of tourism business services 413
# of validation set	1,467,321	# of destination 79
# of testing sets	1,467,321	

5.2. Evaluation measure

This section introduces a segmentation technique used in the marketing field, namely RFM, to convert card payment transactions into rating scores that make the comparison of the proposed methods with other baseline approaches feasible. In other words, baseline methods are based on the rating scores to recommend items, unlike the proposed methods. Also, several measures to evaluate them on top-N recommendation are explained.

Inspiring by [29], the RFM method which, is an instrument for analysis in marketing, is used along with k-means clustering technique to determine the ratings. The RFM indicates recency, frequency, and monetary defined as follows:

- **Recency** is calculated by $R = M + (12 \times (Y - Y_b))$.
- **Frequency** presents the number of transactions per user group.
- **Monetary** means the total amount of transaction per user group.

For the recency factor, Y_b and Y indicate the initial year of transactions contained in our dataset and the year of the corresponding transaction of each user group, respectively. We set $Y_b = 2012$ since our dataset contains card payment data occurred from 2012 to 2019. To combine these three features, we use different weights according to their significance level. We set the weights of recency, frequency, and monetary as 1, 2, and 4, by following [29]. As presented in Algorithm 4, we generate ratings as labels for each transaction that is statistically processed to protect personal information. First, we remove the top 1%

⁴ KSIC: http://kssc.kostat.go.kr/ksscNew_web/ekssc/main/main.do

Algorithm 4: Calculation ratings by RFM method

Data: Transaction data T , feature weights
Result: Labeled transaction data \hat{T}

- 1 Copy data from T to \hat{T}
 /* Set the number of clusters */
- 2 Set $k = 5$
- 3 Remove top 1% records for frequency and monetary features
 /* Get labels for each feature */
- 4 **for** Each feature (i.e., recency, frequency, and monetary) **do**
- 5 Run k-means clustering with k to get initial labels
- 6 Reorder the labels based on the clusters' mean values by ascending order
- 7 Add the weighted label into \hat{T}
- 8 **end**
 /* Calculate ratings with features' labels */
- 9 Run k-means clustering with k for three feature labels to get final labels
- 10 Reorder the final labels based on the clusters' mean values by ascending order
- 11 Add ratings into \hat{T}
- 12 Remove the labels for the three features from \hat{T}
- 13 Return \hat{T}

records as outliers or genuine bulk buyers. For each feature above, we then get labels using the k-means clustering method, implemented in the Sklearn Python library⁵, with the cluster number k . We set k as 5 to generate rating scores scaled from 1 to 5 by the above RFM method. The labels resulted by the clustering are reordered by the periods of clusters' mean values with ascending order. To merge the feature labels, we multiply them with corresponding weights. Finally, k-means clustering is conducted with the weighted three features' labels (i.e., multiple feature clustering), and the final labels are reordered to obtain ratings of each transaction.

Since the rating scores are artificially made by the RFM method, we utilize the rank-based evaluation measurements instead of the RMSE and MAE that are directly based on the artificial scores. Among evaluation measures used in this paper, first of all, we introduce MRR (Mean Reciprocal Rank) defined by

$$MRR(U) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{k_u}, \quad (1)$$

where U and k_u indicate a set of users and a rank of the first relevant item for a user u . This measure is simple to compute and easy to interpret. Also, it focuses on only a single item from the list since it puts a high focus on the first relevant element of the list. Although this might not be a good evaluation metric for users who want a list of related items to browse, we consider it since a small number of services are in general used by tourists in the tourism industry, as shown in Table 2. In the table, we can see more 30% user groups used less than five services.

⁵ <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

Table 2. Percentages of travelers by the number of used services

# of used services	< 5	< 10	< 20	< 30	< 40	< 50	50 >=
Percentage	33.31	19.25	18.78	10.34	6.71	4.20	7.41

To consider multiple relevant items, we use $MAP@k$, which is the mean of $AP@k$, which calculates an average $P@k$ for a user, for all the users. And, $P@k$ measures the relevance of items on k recommended items and is defined as follows:

$$P@k = \frac{|R|}{k}, \tag{2}$$

where R refers to relevant items on top- k item list. It is a simple way to know the fraction of relevant items that are good. However, $MAP@k$ is unable to consider the recommended list as an ordered list, since $P@k$ treats all the errors in the recommended list equally.

Therefore, we use mAP (Mean Average Precision). Unlike the above $AP@k$, $AveP@k$ has the ability to reflect the order of a recommendation list. The mAP is the average of the $AveP@k$ that is defined by

$$AveP@k = \frac{1}{|R|} \sum_i^k P@i \times rel@i, \tag{3}$$

where R refers to relevant items on top- k item list, and $P@i$ indicates precision at i . The $rel@i$ is a relevant function that returns 1 if the item at rank i is relevant and 0 otherwise. This measure is able to handle the ranks of lists recommended items naturally and shines for binary (relevant/non-relevant) ratings. However, it is still not fit for fined-grained numerical ratings.

In this regard, we also consider NDCG (Normalized Discounted Cumulative Gain), which is able to use the fact that some items are more relevant than others. In other words, highly relevant items should come before medium relevant items, which should come before non-relevant items. This metric is calculated by DCG_k and $IDCG_k$ defined as follows:

$$\begin{aligned}
 DCG_k &= \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)} \\
 IDCG_k &= \sum_{i=1}^{|REL_k|} \frac{2^{rel_i} - 1}{\log_2(i + 1)} \\
 nDCG_k &= DCG_k / IDCG_k,
 \end{aligned} \tag{4}$$

where rel_i is the graded relevance of the results at position i (i.e., gain), and REL_k refers to a list of relevant items ordered by their relevance up to top- k . Also, the logarithmic reduction factor is added to penalize the relevance score proportionally to item positions.

5.3. Baseline and variant methods

This section introduces baseline approaches compared with our methods in this study. The baselines consist of two approaches [25,29]. It is difficult to directly apply traditional

rating-based methods to the card transaction data since the card payment dataset does not include rating information. Therefore, [29] exploited the RFM method to create the ratings of users' transaction groups and applied the collaborative filtering methods. We compare our methods with the approach using GSVD, SVD, and NMF. The author utilized GSVD and SVD in [29]. We call them $GSVD_{RFM}$, SVD_{RFM} , and NMF_{RFM} . Note that we only use the RFM method described in Algorithm 4 to compare our methods with the baselines, not to make recommendations. Another approach in [25] uses Word2Vec as mentioned in Section 2.2. The author proposed three techniques as content-based, collaborative filtering, and hybrid-based recommendations. These methods are named KNI_{W2V} , NN_{W2V} , and KIU_{W2V} . We use these techniques to show the effectiveness of considering location information on a top- k recommendation. Note that we only considered the skip-gram technique of Word2Vec due to its better performance on our dataset.

Furthermore, we evaluate the proposed methods' variants to reveal the effectiveness of location information in both the Doc2Vec-based data modeling and the top- k recommendations. We proposed three algorithms in Section 4. Also, two Doc2Vec techniques (i.e., PV-DM and PV-DBOW) are used to model data, and we add prefixes $_{DM}$ and $_{DB}$ for the techniques, respectively. Additionally, location information is considered in only data modeling based on Doc2Vec or in the processes of modeling data and making recommendations. These are distinguished by using prefix $_m$ and $_b$. For example, DM-based NNU with location information for both is annotated as NNU_{DM_b} .

We implemented all the above methods using Python and evaluated them in the same experimental environments with the same data sets. First, the validation and test sets were used to determine optimal parameters for each method in a grid search fashion. Using the optimal parameters, we trained the models of all the methods on the training set. Finally, the experimental results on the test set represented in the next section were obtained.

6. Evaluation and discussion

6.1. Comparison of variants

This section evaluates the variants of proposed methods to comprehend the effectiveness of considering location information in modeling and recommendation processes.

Several parameters affect data modeling and result in recommendation performance. These parameters are based on the Gensim toolbox implementation. In this paper, only four parameters are set to a different value from the default in the toolbox. The rest of the parameters are set as the same as presented on the Gensim web page⁶. The details of the parameters and how we tune them are as follows:

- *min_count* ignores the items whose frequency is less than it. Data in recommender systems is very sparse and contains many items observed only a few times in general. To prevent the loss of these items, we set this parameter as one during our experiments.
- *vector_size* represents the dimension of representation vectors, and its default is 100. We empirically set it to different values in the range of [5:50] with 5 increments.

⁶ <https://radimrehurek.com/gensim/models/doc2vec.html>

- *window* assigns the maximum distance between the current and predicted words. It should be large enough to recognize the semantic relationships between words. We test this parameter with different values in the narrow range of [2:20] with 2 increments.
- *epochs* parameter indicates the number of iterations on modeling input data, and its default is 10. In our experiments, it sets to various values in the range of [5:50] with 5 increments.

We conducted a grid search for all their combinations on validation and test sets to find an optimal set of these parameters for each variant. Table 3 lists the performance results and optimal parameter settings of the variants based on the models trained by Doc2Vec techniques. According to Table 3, it can be aware that considering the location information

Table 3. Performance results of variants with optimal parameters

Variants	<i>MAP@10</i>	<i>mAP@10</i>	<i>mNDCG₁₀</i>	<i>MRR@10</i>	Optimal setting
<i>KNB_{DM_m}</i>	0.0495	0.0159	0.0495	0.1330	V: 5, W: 10, E: 10
<i>NNU_{DM_m}</i>	0.1868	0.1034	0.2269	0.6071	V: 20, W: 10, E: 10
<i>NKB_{DM_m}</i>	0.0482	0.0160	0.0493	0.1355	V: 5, W: 10, E: 10
<i>KNB_{DB_m}</i>	0.0124	0.0023	0.0095	0.0179	V: 50, W: 2, E: 25
<i>NNU_{DB_m}</i>	0.1853	0.1086	0.2270	0.5902	V: 10, W: 8, E: 15
<i>NKB_{DB_m}</i>	0.0100	0.0016	0.0087	0.0152	V: 50, W: 6, E: 25
<i>KNB_{DM_b}</i>	0.1374	0.0588	0.1408	0.3072	V: 40, W: 8, E: 15
<i>NNU_{DM_b}</i>	0.2843	0.2009	0.3748	0.9287	V: 20, W: 4, E: 5
<i>NKB_{DM_b}</i>	0.1180	0.0504	0.1225	0.2794	V: 40, W: 8, E: 15
<i>KNB_{DB_b}</i>	0.0647	0.0199	0.0639	0.1709	V: 50, W: 10, E: 15
<i>NNU_{DB_b}</i>	0.2673	0.1816	0.3539	0.9231	V: 10, W: 8, E: 25
<i>NKB_{DB_b}</i>	0.0461	0.0132	0.0446	0.1213	V: 5, W: 10, E: 8

^a V, W, and E indicate parameters *vector_size*, *window*, and *epoch*.

in both modeling data and making recommendations has a lot of improvements from applying only in data modeling.

In terms of Doc2Vec techniques, when we consider location information in only data modeling, data modeling based on PV-DM has more positive influences than that of PV-DBOW. These results are more clear when we compare it with those of the *KN_IW_{2V}*, *NN_{W2V}*, and *K_IU_{W2V}* in Table 5. The variants considering the spatial information in only modeling recommend services based on vector representations of user groups like the baseline methods. We use Doc2Vec techniques to model users' preferences and spatial information simultaneously, but the models in [25] consider only users' preferences based on the techniques of Word2Vec. The *KNB_{DM_m}* and *NKB_{DM_m}* have better performance than the *KN_IW_{2V}* and *K_IU_{W2V}*, while the performance results of the *KNB_{DB_m}* and *NKB_{DB_m}* are worse than them (refer to Table 5). In fact, these results are related to the Doc2Vec implementation of Gensim toolbox. The PV-DBOW trains only document vectors with the default setting for *dbow_words*, and it means that

the KNB_{DB_m} , NNU_{DB_m} , and NKB_{DB_m} may be unable to appropriately consider the individual history of user groups on the recommendation process. As a result, they show lower performance than the KNB_{DM_m} , NNU_{DM_m} , and NKB_{DM_m} , respectively. Even though we had actually evaluated the PV-DBOW with $dbow_words = 1$, which set the technique works in the skip-gram fashion, we couldn't discover remarkable performance differences. Therefore, we presented the performance results based on the pure PV-DBOW technique in this paper. However, the KNB_{DB_b} , NNU_{DB_b} , and NKB_{DB_b} considering the location information in both procedures are superior to the KNI_{W2V} , NN_{W2V} , and KIU_{W2V} . It implies that considering location information in tourism service recommendations is important. Regarding recommendation algorithms, all methods, regardless of Doc2Vec techniques and the consideration of location information, show similar trends of performance results. The NNU methods are superior to the others in all evaluation metrics, and the KNB methods perform better than the NKB s. We carefully guess that the reason is caused by the construction process of input data to model card transaction data. Because a user group's identification locates as the first term in the input, the services placed at the beginning have similar vector representations with the user group due to the principle of Doc2Vec techniques. Consequently, KNB , which directly searches for similar services with a target user group, could have a severe bias between services according to their locations in the input data. It also happens to the NKB . Accordingly, we select the NNU_{DM_b} and NNU_{DB_b} to compared with baseline methods in the next section.

To comprehend the effects of three parameters for Doc2Vec techniques on top-10 recommendations, this section evaluates the proposed methods (i.e., NNU_{DM_b} and NNU_{DB_b}) by changing the Doc2Vec parameters in the corresponding ranges mentioned above. While repeating the different ranges of one parameter, the others are fixed to constant values. Figures 4a and 4b show the performance of NNU_{DM_b} and NNU_{DB_b} by parameter value. NNU methods seem to be not affected by the three

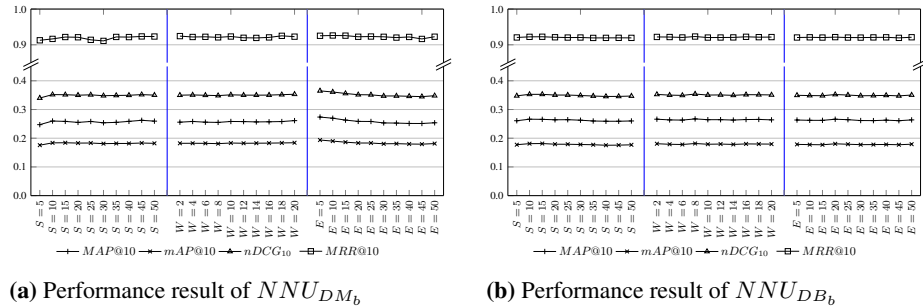


Fig. 4. Performance results of NNU

parameters unlike the other variants in our preliminary results (omitted due to limited space). To clearly reveal the effects of these parameters on these methods, we analyzed the correlation between the parameters and performance, as listed in Table 4. The correlations indicate that the increase of $vector_size$ and $window$ parameters positively affects while

Table 4. Correlation analysis between parameters and performance for NNUs

NNU_{DM_b} v_size $window$ $epoch$				NNU_{DB_b} v_size $window$ $epoch$			
$MAP@10$	0.535	0.612	-0.903	$MAP@10$	-0.695	-0.160	-0.229
$mAP@10$	0.267	0.709	-0.879	$mAP@10$	-0.700	-0.068	-0.023
$MRR@10$	0.552	-0.076	-0.752	$MRR@10$	-0.789	-0.086	-0.134
$mNDCG_{10}$	0.393	0.611	-0.897	$mNDCG_{10}$	-0.712	-0.104	-0.093

the epoch increment has a negative influence on the recommendation performance of the NNU_{DM_b} method. Whereas, in the case of NNU_{DB_b} method, the increments of all the parameters result in worse performance. Indeed, we observed that the proposed methods based on the PV-DM technique are positively affected by the increment of the $vector_size$ and $window$ parameters, while the proposed methods based on PV-DBOW have mostly negative influences by the increase of all three parameters. However, as shown in the experimental results of previous sections, we need to explore the parameter combinations rigorously to set the optimal one. According to Figure 4 and the results discussed above, the NNU methods have the best performance in general. Consequently, we select the NNU_{DM_b} and NNU_{DB_b} to compared with baseline methods in the next section.

6.2. Comparison with baselines

In this section, we compare the performance results of the two NNU methods based on the pre-trained card transaction data (i.e., the representation vectors of users and items, locations) and the baseline approaches mentioned in Section 5.3. Table 5 presents their performance on various top- k recommendations. The bold and italic font styles indicate the first and second-best performance. According to this table, the proposed methods are su-

Table 5. Performance results of the proposed and baselines methods

Methods	Top-10				Top-5				Top-2			
	MA	mA	mN	MR	MA	mA	mN	MR	MA	mA	mN	MR
$GSVD_{RFM}$	0.170	0.106	0.214	0.529	0.137	0.262	0.093	0.165	0.117	0.190	0.101	0.117
$SVDR_{RFM}$	0.131	0.082	0.176	0.492	0.125	0.239	0.087	0.150	0.097	0.164	0.083	0.106
NMF_{RFM}	0.107	0.059	0.135	0.378	0.072	0.233	0.053	0.094	0.091	0.165	0.085	0.105
$KNIW_{2V}$	0.029	0.007	0.023	0.042	0.008	0.014	0.003	0.007	0.003	0.004	0.002	0.003
NN_{W2V}	0.124	0.072	0.148	0.355	0.084	0.211	0.061	0.101	0.079	0.121	0.074	0.086
KIU_{W2V}	0.032	0.008	0.028	0.070	0.010	0.038	0.008	0.014	0.017	0.033	0.017	0.020
NNU_{DM_b}	0.284	0.201	0.375	0.929	0.366	0.265	0.426	0.862	0.294	0.277	0.341	0.535
NNU_{DB_b}	0.267	0.182	0.354	0.923	0.274	0.224	0.363	0.841	0.292	0.276	0.339	0.534

^a MA , mA , mN , MR refer to the MAP , mAP , $mNDCG$, and MRR , respectively.

perior to the other baselines in most performance measures. Interestingly, the Word2Vec-based approaches (i.e., $KNIW_{2V}$, NN_{W2V} , and KIU_{W2V}) show worse performance than the other baseline methods. These results might be because of the adopted evaluation methodology. It is based on the RFM, which makes it possible to work matrix

factorization-based collaborative filtering approaches on card payment transaction data, and is used to obtain a ground truth set. In other words, the methodology could be favorable to the RFM-based methods (i.e., $GSVD_{RFM}$, SVD_{RFM} , and NMF_{RFM}). Despite this, the proposed methods NNU_{DM_b} and NNU_{DB_b} outperform the other methods. It indicates that our methods can adequately model the card transaction data by considering the spatial information and to make appropriate recommendations to a user group that visited a specific location.

In terms of evaluation measurements, the MRR results of the proposed methods on top-10 recommendations are around 0.9, which has a large difference from the results in other measures. Considering the MRR 's evaluation purpose focusing on only a single item, we can see that the proposed methods have quite high performance to recommend the next service that can be used by a target user group at a specific location. Additionally, the second high performance of proposed methods is $mNDCG$ for top- k recommendations. It means that the NNU_{DM_b} and NNU_{DB_b} using the vector similarity trained by Doc2Vec techniques work well in a graded rating fashion.

Let's discuss the results in terms of top- k recommendations (i.e., the number of recommended items). With the more decrease of k , the performance of the baseline approaches is worse, except for in mAP . The MRR shows a larger decrement than the other measurement in the baselines, while our methods have relatively smaller decrements in the measurement. The methods' MRR performance is higher than 53% on the top-2 recommendation. These results emphasize the potential capability of our methods for a next service recommendation which can be used in many recommendation purposes such as tour planning, dynamic recommendation, and so on. Interestingly, the proposed methods show slightly higher performance in the mAP when it recommends the smaller numbers of business services. Furthermore, except for MRR , the performance decrements of the proposed methods are in general smaller than those of the other approaches. These results imply that the proposed methods provide services with more proper ordering regardless of the number of recommended services than the others.

7. Conclusion

Millions of card transactions, which eventually reflect tourist consumption behaviors and patterns, generate a massive volume of big data in tourism. However, it is difficult to directly apply the available data to recommender systems since the huge amount of data contains generally implicit preferences of the tourists. Furthermore, the row data of card payment transactions, which contain personal information, may not be available in terms of GDPR. In addition to these, it is important to properly reflect a spatial factor in tourism recommender systems.

To address these challenges, we propose tourism service recommendation methods based on Doc2Vec techniques, a set of well-known methods from the natural language processing domain, for a target user group visiting a specific location. In order to model the card transaction data statistically processed to protect personal information, the techniques train a model on the service usage history of user groups along with spatial information. The vector representations are then used in three recommendation methods to make recommendations by considering the location information.

Experiments on around fourteen million statistical card transaction data demonstrated that the proposed recommendation methods outperform other baseline methods. In particular, comparing the proposed methods with other baselines emphasized the positive influences of spatial information on recommendation performance. Furthermore, these methods showed the capability to deal with various top-k recommendations without high decrements in recommendation performance than the other compared approaches. In addition to these, the proposed methods are able to recommend business services to new user groups whose data does not exist in the dataset and are directly applied to raw transaction data to provide recommendations to individuals.

Acknowledgments. This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019S1A3A2098438).

References

1. Al-Ghossein, M., Abdessalem, T., Barré, A.: Cross-domain recommendation in the hotel sector. In proceedings of the Workshop on Recommenders in Tourism, RecTour 2018, co-located with the 12th ACM Conference on Recommender Systems (RecSys 2018). Vancouver, Canada. (2018)
2. Almeida, F., Almeida, J., Mota, M.: Perceptions and trends of booking online payments in tourism. *Journal of Tourism and Services*, Vol. 10, No. 18, 1–15. (2019)
3. Amato, F., Moscato, V., Picariello, A., Piccialli, F.: SOS: A multimedia recommender system for online social networks. *Future Generation Computer System*, Vol. 93, 914–923. (2019)
4. Baek, J.W., Chung, K.Y.: Multimedia recommendation using word2vec-based social relationship mining. *Multimedia Tools and Applications*, 1–17. (2020)
5. Bahramian, Z., Abbaspour, R.A.: An ontology-based tourism recommender system based on spreading activation model. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, Vol. 40. (2015)
6. Cai, G., Lee, K., Lee, I.: Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos. *Expert Systems with Applications*, Vol. 94, 32–40. (2018)
7. Casillo, M., Clarizia, F., Colace, F., Lombardi, M., Pascale, F., Santaniello, D.: An approach for recommending contextualized services in e-tourism. *Inf. Vol. 10, No. 5*, 180. (2019)
8. Chaudhari, K., Thakkar, A.: A comprehensive survey on travel recommender systems. *Archives of Computational Methods in Engineering*, 1–27. (2019)
9. Chen, L., Yang, W., Li, K., Li, K.: Distributed matrix factorization based on fast optimization for implicit feedback recommendation. *J. Intell. Inf. Syst.*, Vol. 56, No. 1, 49–72. (2021)
10. Dev, H., Hamooni, H.: Profiling US restaurants from billions of payment card transactions. In proceedings of the 7th IEEE International Conference on Data Science and Advanced Analytics, Sydney, Australia. (2020)
11. Du, M., Christensen, R., Zhang, W., Li, F.: Pcard: Personalized restaurants recommendation from card payment transaction records. In proceedings of the World Wide Web Conference, WWW, San Francisco, CA, United States. (2019)
12. Esmaili, L., Mardani, S., Golpayegani, S.A.H., Madar, Z.Z.: A novel tourism recommender system in the context of social commerce. *Expert Systems with Applications*, Vol. 149, 113301. (2020)
13. Esmeli, R., Bader-El-Den, M., Abdullahi, H.: Using word2vec recommendation for improved purchase prediction. In proceedings of the 2020 International Joint Conference on Neural Networks, Glasgow, United Kingdom. (2020)
14. Guo, L., Liang, J., Zhu, Y., Luo, Y., Sun, L., Zheng, X.: Collaborative filtering recommendation based on trust and emotion. *J. Intell. Inf. Syst.* Vol. 53, No. 1, 113–135. (2019)

15. Hong, M.: Decrease and conquer-based parallel tensor factorization for diversity and real-time of multi-criteria recommendation. *Information Sciences*, Vol. 562, 259–278. (2021)
16. Hong, M., Jung, J.J.: Multi-criteria tensor model consolidating spatial and temporal information for tourism recommendation. *J. Ambient Intell. Smart Environ.*, Vol. 13, No. 1, 5–19. (2021)
17. Hong, M., Jung, J.J.: Multi-criteria tensor model for tourism recommender systems. *Expert Systems with Applications*, Vol. 170, 114537. (2021)
18. Hong, M., Koo, C.M., and Chung, N.H.: DSER: Deep-Sequential Embedding for single domain Recommendation. *Expert Systems with Applications*, Vol. 208, 118156. (2022)
19. Kotiloglu, S., Lappas, T., Pelechris, K., Repoussis, P.: Personalized multi-period tour recommendations. *Tourism Management*, Vol. 62, 76–88. (2017)
20. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In proceedings of the 31th International Conference on Machine Learning, Beijing, China. (2014)
21. Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., Chen, E.: Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina. (2015)
22. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: A survey. *Decis. Support Syst.* Vol. 74, 12–32. (2015)
23. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, United States. (2013)
24. Musto, C., Semeraro, G., de Gemmis, M., Lops, P.: Word embedding techniques for content-based recommender systems: An empirical evaluation. In proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria. (2015)
25. Ozsoy, M.G.: From word embeddings to item recommendation. *CoRR abs/1601.01356* (2016)
26. Park, S.T., Liu, C.: A study on topic models using lda and word2vec in travel route recommendation: focus on convergence travel and tours reviews. *Personal and Ubiquitous Computing*, 1–17. (2020)
27. Pessemier, T.D., Dhondt, J., Martens, L.: Hybrid group recommendations for a travel service. *Multim. Tools Appl.* Vol. 76, No. 2, 2787–2811. (2017)
28. Rizvi, S., Kurtz, A., Williams, I., Gualdoni, J., Myzyri, I., Wheeler, M.: Protecting financial transactions through networks and point of sales. *Journal of Cyber Security Technology*, Vol. 4, No. 4, 211–239. (2020)
29. Sharifhosseini, A.: A case study for presenting bank recommender systems based on bon card transaction data. In proceedings of the 9th International Conference on Computer and Knowledge Engineering. (2019)
30. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* Vol. 52, No. 1, 5:1–5:38. (2019)

Minsung Hong is a researcher in Data Science Lab (DSL) of Korea Electric Power Corporation, South Korea since June 2022 after a research professor in Kyung Hee University from February 2021 to May 2022. He was a postdoctoral researcher participating in several EU Horizon2020 and Norway national projects in Western Norway Research Institute, Norway from February 2018 to January 2021. He received the Ph.D. degree in Computer Engineering from Chung-Ang University in 2018. His research topics are recommender systems, big data, artificial intelligence, data mining, machine learning, and natural language processing.

Namho Chung is a Dean of College of Hotel & Tourism Management, Professor at the Smart Tourism Education Platform and the Director of Smart Tourism Research Center at

Kyung Hee University in Seoul, South Korea. He has been a Visiting Research Fellow at School of Hospitality and Tourism Management, University of Surrey in Guildford, UK. His research interests include travel behavior, information search and decision making, destination marketing, knowledge management. Currently, he leads smart tourism city projects in the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea. His name listed in the Hall of Fame at Kyung Hee University for his outstanding achievements.

Chulmo Koo is a Professor of Smart Tourism Education Platform (STEP), College of Hotel and Tourism Management at Kyung Hee University, South Korea. He is an Editor-in-Chief of Journal of Smart Tourism and has a strong record of smart tourism research and scholarship with significant contributions to the smart tourism field.

Received: Jun 20, 2022; Accepted: December 29, 2022.

Read between the Interactions: Understanding Non-interacted Items for Accurate Multimedia Recommendation

Jiyeon Kim*, Taeri Kim*, and Sang-Wook Kim†

Department of Computer Science, Hanyang University
222 Wangsimni-ro, Seongdong-gu, Seoul, Korea
{jiyeon7, taerik, wook}@hanyang.ac.kr

Abstract. This paper addresses the problem of multimedia recommendation that additionally utilizes multimedia data, such as visual and textual modalities of items along with the user-item interaction information. Existing multimedia recommender systems assume that *all the non-interacted items of a user have the same degree of negativity*, thus regarding them as candidates for negative samples when training the model. However, this paper claims that a user’s non-interacted items do not have the same degree of negativity. We classify these non-interacted items of a user into two kinds of items with different characteristics: *unknown and uninteresting items*. Then, we propose a novel negative sampling technique that only considers the uninteresting items (*i.e.*, rather than the unknown items) as candidates for negative samples. In addition, we show that using the multiple Bayesian personalized ranking (BPR) losses with both unknown and uninteresting items (*i.e.*, all the non-interacted items) in existing multimedia recommendation methods is very effective in improving recommendation accuracy. By conducting extensive experiments with three real-world datasets, we show the superiority of our ideas. Our ideas can be easily and orthogonally applied to any multimedia recommender systems.

Keywords: recommender systems, multimedia recommendation, uninteresting items.

1. Introduction

Due to the abrupt increase in the number and variety of items around us, the problem of information overload is becoming a big issue in many applications. Recommender systems are a vital technique to solve this problem and thus are widely used in various domains, such as movie recommendations and music recommendations. *Collaborative filtering* (CF) is one of the most widely used approaches in recommender systems; intuitively, for a target user, it finds the items commonly preferred by the users with the tastes similar to hers (*i.e.*, neighbors) based on her interaction information (*e.g.*, purchase history and click logs) [6, 9, 11–16, 19, 21, 22, 25, 28, 31–34, 39]. Despite the simplicity and robustness of CF in recommendation, the sparse nature of the interaction information brings CF the limitation of not being able to accurately capture the users’ preferences on items [3].

* The first two authors have equally contributed to this work.

† Corresponding author.

To alleviate this limitation of CF, various methods have been proposed [4, 8, 11, 18, 19, 21, 23, 36, 37, 43, 44]. They can be classified into two categories: i) additional utilization of non-interacted items and ii) additional utilization of external data. The methods in category i) first divide a user's non-interacted items into her unknown items and uninteresting items [11, 18, 19, 21] where the unknown items are the items that the user did not interact with because she did not know their existence and the uninteresting items are the items that the user did not interact with even though she knew their existence but did not want to interact with the item. Then, the methods mitigate the data sparsity problem by selecting her uninteresting items amongst non-interacted items and imputing low values for the uninteresting items selected [11, 18, 19, 21]. The methods in category ii) use additional multimedia data (*e.g.*, visual data such as the item's image and textual data such as the item's specifications) along with the user-item interaction information. The recommender systems of this category are referred to as *multimedia recommender systems* [4, 8, 23, 36, 37, 43, 44].

Most multimedia recommender systems use deep learning models such as convolutional neural networks (CNNs) [1, 17, 26, 42] and recurrent neural networks (RNNs) [7, 10, 38] to extract multimodal features from the items' multimedia data. They utilize these multimodal features to represent the item embeddings; they conduct a dot product between an item embedding and a user embedding to predict the user's preference on the item. They use the *Bayesian personalized ranking* (BPR) loss [30], a representative pairwise loss to learn the ranking difference between a user's positive and negative items, to train their models. In model training, positive items are sampled from the user's *interacted items* and the negative items are randomly sampled from the user's *non-interacted items*. In other words, they simply use all the non-interacted items as the candidates for negative items based on the assumption that *all the non-interacted items for a user have the same degree of negativity*.

However, we claim that this assumption does not hold in the real-world data; *i.e.*, non-interacted items could have different degrees of negativity. Then, we propose the methods that utilize the two categories of a user's non-interacted items for accurate multimedia recommendation. Note that the proposed methods can be easily applied (*i.e.*, orthogonally applicable) to existing multimedia recommender systems. To this end, we first classify a user's non-interacted items into two categories of unknown and uninteresting items for her based on the degrees of her negativity, obtained by using the user's interacted items. Then, we propose a novel negative sampling technique that uses only the uninteresting items (rather than unknown items) as candidates for negative samples. Furthermore, we propose to use multiple BPR losses which utilize both unknown and uninteresting items (*i.e.*, all non-interacted items), in multimedia recommender systems. To demonstrate the effectiveness of our proposed methods, we employ three well-known multimedia recommender systems (spec. VBPR [8], MMGCN [37], and LATTICE [43]) and three real-world Amazon datasets.¹ To show the superiority of our negative sampling method, we compare the following three methods: i) using those randomly sampled from non-interacted items as negative samples (*i.e.*, original negative sampling); ii) using unknown items as negative samples; iii) using uninteresting items as negative samples (*i.e.*, our negative sampling). Our experimental result shows that our proposed method (*i.e.*, method iii) provides the best recommendation accuracy. The result also confirms that

¹ <http://jmcauley.ucsd.edu/data/amazon/links.html>

using multiple BPR losses is more effective in multimedia recommender systems: specifically, applying the multiple BPR losses leads to a gain of up to 20.13% and 4.12%, in terms of Recall@20, compared to the state-of-the-art multimedia recommender systems, MMGCN [37] and LATTICE [43], respectively.

The main contributions of our work are summarized as follows:

- We point out the problem of the assumption employed in existing multimedia recommender systems.
 - All the non-interacted items for a user have the same degree of negativity.
- We propose two methods that improve the recommendation accuracy by exploiting the different degrees of negativity in non-interacted items.
 - We propose a novel sampling technique that uses only the uninteresting items as negative samples.
 - We use multiple BPR losses to learn the rank differences between positive, unknown, and uninteresting items.
- We validate our proposed methods by conducting extensive experiments using three real-world datasets.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work to multimedia recommender systems. In Section 3, we describe our proposed methods in detail. In Section 4, we conduct experiments to verify the effectiveness of our methods. Finally, in Section 5, we summarize and conclude our paper.

2. Related Work

In this section, we briefly introduce the research on multimedia recommender systems. Early multimedia recommender systems utilized only one modality amongst the items' multimedia data (*e.g.*, visual, textual, and acoustic modality) along with the user-item interaction information [2, 5, 8, 12, 24, 35, 40, 41]. VBPR [8], the most popular model among them, captures the features of the visual modality of items and builds an additional embedding that reflects each user's preference for the visual modality of items. Then, it uses the well-known BPR loss in training VBPR. However, the early multimedia recommender systems have a limitation that they use only one of various modalities of items to represent the items' characteristics.

To alleviate this limitation, recent multimedia recommender systems have tried to utilize various modalities of items [4, 14, 23, 36, 37, 43, 44]. Specifically, JRL [44] and MAML [23] use deep learning models to capture the features of the various modalities of the item (*e.g.*, visual, textual, and numerical (*i.e.*, rating) modalities for JRL and visual and textual modalities for MAML). Then, they aggregate the captured features to enrich the embedding of an item and the user who interacted with the corresponding item. MMGCN [37] constructs the user-item interaction graphs for visual, textual, and acoustic modalities of items, and uses graph convolutional networks (GCNs) to capture the collaborative signals between the users and the items. Then, MMGCN aggregates the collaborative signals captured by each modality and enriches the embeddings of users and items.

Since the advent of MMGCN, various GCN-based multimedia recommender systems have emerged such as GRCN [36] and LATTICE [43]. They commonly use not only

GCNs but also the attention mechanism to distinguish the degrees of influence of users on different modalities of items. GRCN [36] is based on MMGCN and considers the degrees of influence on different modalities at an individual user level. On the other hand, LATTICE [43] captures the latent item-item structure for each modality using visual and textual modalities of items and then applies GCN to obtain enriched item embeddings. Then, it considers the degrees of influence on different modalities at all user levels (*i.e.*, globally for all users).

The aforementioned methods utilize the BPR loss, which selects positive items among the interacted items and negative items among the non-interacted items and widens the rank discrepancy between positive and negative items, in order to learn their models [8, 36, 37, 43, 44]. However, since there are many items in recommendation domain data, it is difficult for users to know the existence of all items. Therefore, a user's non-interacted items can be categorized into unknown and uninteresting items as follows:

- **Unknown item:** item that a user did not interact with because she did not know its existence.
- **Uninteresting item:** item that a user did not interact with because she did not want to interact with it, even though she knew its existence.

In other words, if the BPR loss is simply employed in a learning process as in existing multimedia recommender systems, some non-interacted items that the user may prefer can be considered as her negative items. Therefore, we argue that, in order to correctly train the model by using the BPR loss, negative items should be sampled not from her non-interacted items, but from her uninteresting items. In addition, we argue that we should train the model so that they will be able to learn all the rank discrepancies among positive, unknown, and uninteresting items.

3. Proposed Methods

In this section, we propose two methods that can be orthogonally applied to existing multimedia recommender systems, exploiting the notions of unknown and uninteresting items for accurate multimedia recommendation. Specifically, in Section 3.1, we describe our novel negative sampling method that uses only uninteresting items as negative samples. Then, in Section 3.2, we describe how to use multiple BPR losses, for interesting, unknown, and uninteresting items in model training.

3.1. Negative Sampling Method

The overall procedure of our negative sampling is shown in Figure 1. As mentioned, a user's non-interacted items are categorized into unknown items and uninteresting items. Our negative sampling method samples only the uninteresting items of a user as negative samples for BPR training.

For this, we first compute the pre-use preferences of each user on her non-interacted items by analyzing users' interaction information. A user's pre-use preference is the preference that the user has when deciding whether to interact with an item or not [11, 18, 19, 21]. Thus, we can say that, for the user's interacted items, she holds a high pre-use preference. On the other hand, for those items that she has not interacted, her pre-use

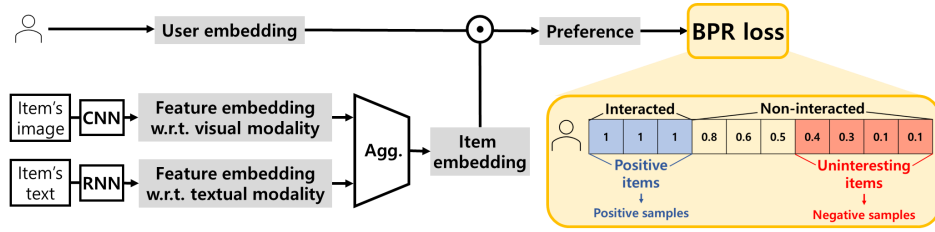


Fig. 1. Overview of our negative sampling method. The blue-colored items indicate the user’s interacted items (*i.e.*, positive items), and the red-colored items indicate the user’s uninteresting items (*i.e.*, negative items) among her non-interacted items

preferences would be lower than those of the interacted items. Among non-interacted items, the user’s pre-use preferences on unknown items are unknown, however, those on uninteresting items should be low since she was not interested in them.

In order to obtain pre-use preference scores for non-interacted items of a user, in this paper, we employ WRMF [27], a widely adopted model in one-class setting, following [11, 18, 21].² Specifically, given the pre-use preference matrix $\mathbf{P} \in R^{\# \text{ of users} \times \# \text{ of items}}$ ($p_{u,i} = 1$, if user u has interacted with item i), WRMF predicts users’ pre-use preferences for all non-interacted items. To this end, we first initialize users’ pre-use preferences for non-interacted items as 0 in \mathbf{P} and assign weights to quantify the relative contribution of each user-item interaction [27]. Then, WRMF repeats the process of decomposing the pre-use preference matrix \mathbf{P} into two low-rank matrices $\mathbf{U} \in R^{\# \text{ of users} \times d}$ and $\mathbf{V} \in R^{\# \text{ of items} \times d}$ where d indicates the dimensionality of each latent feature vector and multiplying these two decomposed matrices to recover the original pre-use preference matrix \mathbf{P} . The loss function of WRMF is as follows:

$$\mathcal{L}(\mathbf{U}, \mathbf{V}) = \sum_{u,i} w_{u,i} (p_{u,i} - \mathbf{U}_u \mathbf{V}_i^T)^2 + \lambda (\sum_u \|\mathbf{U}_u\|_F^2 + \sum_i \|\mathbf{V}_i\|_F^2), \quad (1)$$

where $p_{u,i}$ denotes user u ’s pre-use preference of item i ; $w_{u,i}$ denotes the weight for $p_{u,i}$ and \mathbf{U}_u and \mathbf{V}_i represent the latent feature vectors of user u and item i , respectively; $\|\cdot\|_F$ denotes the *Frobenius norm* and λ denotes the regularization parameter.

Lastly, we obtain the predicted pre-use preference matrix $\hat{\mathbf{P}}$ using the learned vectors \mathbf{U} and \mathbf{V} as follows:

$$\hat{\mathbf{P}} = \mathbf{U}\mathbf{V}^T. \quad (2)$$

Then, we use $(1 - \hat{p}_{u,i})$ as the final weight for non-interacted item i to be sampled as a negative sample. By doing this, we allow the negative samples to be selected only from the uninteresting items, rather than from all non-interacted items. This is because the pre-use preference $\hat{p}_{u,i}$ of an uninteresting item will be low, thus making the weight (*i.e.*, $(1 - \hat{p}_{u,i})$) high. Finally, we use the negative samples selected from the uninteresting items in the BPR loss of the existing multimedia recommender systems.

² Note that, if there is a better model available other than WRMF, it could improve more the recommendation accuracies with our proposed method.

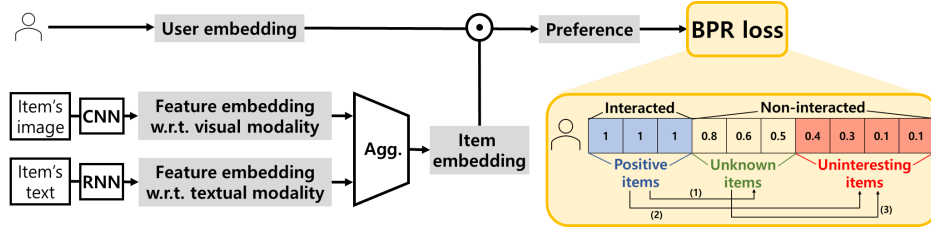


Fig. 2. Overview of the method using the multiple BPR losses. The blue-colored items indicate the user’s interacted items, the gray-colored items indicate the user’s unknown items among her non-interacted items, and the red-colored items indicate the user’s uninteresting items among her non-interacted items. (1)-(3) indicates the rank discrepancies the method using the multiple BPR losses considers

3.2. Multiple BPR losses

Note that the BPR loss employed in existing multimedia recommender systems is only used to correctly learn the rank discrepancy between the predicted preferences of positive and negative items. However, the entire items can be divided into positive, unknown, and uninteresting items; so we can better learn the rank of non-interacted items if we fully exploit the rank discrepancy among all pairs of the above three types. Toward this end, we propose to use multiple BPR losses³ in existing multimedia recommender systems, which enables to learn the rank discrepancies amongst the predicted preferences of the above three types of items (*i.e.*, not only uninteresting items but also unknown items can be used).

Our method using the multiple BPR losses is shown in Figure 2. With the items categorized into three types (*i.e.*, positive, unknown, and uninteresting items), we train the model by using the following three rank discrepancies: (1) between positive and unknown items, (2) between positive and uninteresting items, and (3) between unknown and uninteresting items. We use the three weights (*i.e.*, α for (1), β for (2), and γ for (3)) to control the importance of the three rank discrepancies. The multiple BPR losses are formulated as follows:

$$\mathcal{L} = - \sum_u \sigma(\alpha(\hat{r}_{pos} - \hat{r}_{unk})) + \sigma(\beta(\hat{r}_{pos} - \hat{r}_{unint})) + \sigma(\gamma(\hat{r}_{unk} - \hat{r}_{unint})) + R(\theta), \quad (3)$$

where \hat{r}_{pos} , \hat{r}_{unk} , and \hat{r}_{unint} denote the predicted preferences of positive, unknown, and uninteresting items, respectively; $\sigma(\cdot)$ indicates the sigmoid function and $R(\theta)$ does the regularization term for model parameters θ .

To utilize our multiple BPR losses in multimedia recommender systems, we need a user’s predicted pre-use preference scores of non-interacted items, as stated in Section 3.1. Then, with those scores, we regard the bottom $\mu\%$ of non-interacted items as uninteresting items, and the rest of them as unknown items. Lastly, we apply the multiple BPR losses in Eq. (3) to multimedia recommender systems.

The proposed methods in Sections 3.1 and 3.2, are easily and orthogonally applicable to existing multimedia recommender systems, helping to provide more accurate multimedia recommendations.

³ A similar idea proposed in non-multimedia recommendation [20].

Table 1. Statistics of datasets. The sparsity calculated by $\frac{\# \text{ of users} \cdot \# \text{ of items} - \# \text{ of interactions}}{\# \text{ of users} \cdot \# \text{ of items}} \times 100$ (%).

Dataset	# of users	# of items	# of interactions	Sparsity
Amazon Baby	19,445	7,050	160,792	99.88%
Amazon Men Clothing	4,955	5,028	32,363	99.87%
Amazon Office	4,874	2,406	52,957	99.55%

4. Evaluation

In this section, we evaluate our proposed methods via experiments; the experiments are designed aiming to answer the following key evaluation questions:

- **EQ1:** Do the notions of unknown and uninteresting items help improve the recommendation accuracy of multimedia recommender systems?
- **EQ2:** Is the idea of selecting the uninteresting items as negative samples most effective for improving the recommendation accuracy?
- **EQ3:** How sensitive is the recommendation accuracy of the multiple BPR losses to different hyperparameter values?

4.1. Experimental Settings

Datasets and competitors For evaluation, we adopt three real-world Amazon datasets widely used in multimedia recommender systems research [4, 8, 23, 43, 44]: Amazon Baby, Amazon Men Clothing, and Amazon Office⁴. As done in [37], we kept only the users and items with more than five interactions. Table 1 reports their detailed statistics. These datasets contain visual and textual modality information of items as well as the user-item interaction. Then, we extracted 4,096-dimensional visual feature embeddings using the deep CNN [17] and 1,024-dimensional textual feature embeddings using sentence transformers [29], following [43].

To evaluate the effectiveness of our proposed methods, we use the following three baselines:

- VBPR [8]: A multimedia recommender system based on matrix factorization (MF) trained with a BPR loss.
- MMGCN [37]: A multimedia recommender system based on graph convolutional networks (GCNs) using non-linear propagation trained with a BPR loss.
- LATTICE [43]: A multimedia recommender system based on graph convolutional networks (GCNs) using linear propagation trained with a BPR loss.

Evaluation protocol and metrics We repeated all our experiments five times. For each experiment, we randomly split interactions per user into 8:1:1, each for train, validation,

⁴ All the datasets are publicly available at <http://jmcauley.ucsd.edu/data/amazon/links.html>.

and test set in the same way as in [37, 43]. We assess the accuracy of top- N recommendation by using the following three widely used metrics: Precision (Prec, in short), Recall, and normalized discounted cumulative gain (NDCG). Prec and Recall are traditional accuracy metrics. They are used to validate whether the ground-truth items are in top- N recommendation list and computed as follows:

$$Prec@N = \frac{|Rel_u \cap N_u|}{N} \quad (4)$$

$$Recall@N = \frac{|Rel_u \cap N_u|}{|Rel_u|} \quad (5)$$

where Rel_u indicates the relevant observed items of user u and N_u indicates the top- N items of user u .

NDCG is a rank-sensitive metric which considers the position of the ground-truth item in the top- N recommendation list and is computed as follows:

$$NDCG@N = \frac{DCG@N}{IDCG@N}. \quad (6)$$

Additionally, $DCG@N$ in Eq. (6) is computed as follows:

$$DCG@N = \sum_{k=1}^N \frac{2^{y_k} - 1}{\log_2(k + 1)}, \quad (7)$$

where y_k indicates the binary variable for the k -th item i_k in N_u and, if $i_k \in Rel_u$, y_k is set as 1, otherwise, y_k is set as 0. And, $IDCG@N$ in Eq. (6) stands for *ideal DCG* at N where, for every item i_k in N_u , y_k is set as 1. We set N to 10 and 20 for all aforementioned metrics.

Hyperparameter Settings For a fair comparison, we fine-tuned the hyperparameters of competitors and our proposed methods via grid search using the validation set. More specifically, we set the learning rate in the range $\{0.0001, 0.0005, 0.001, 0.005, 0.01\}$ and the regularization weight in the range $\{0, 0.00001, 0.0001, 0.001, 0.01\}$. Also, for MMGCN [37] and LATTICE [43], we set the number of GCN-layers in the range $\{1, 2, 3, 4\}$; for LATTICE [43], we set the dropout ratio in the range $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$.

4.2. Experimental Results

EQ1: Do the notions of unknown and uninteresting items help improve the recommendation accuracy of multimedia recommender systems? To show the effectiveness of our proposed methods, we compared the three (original) competitors (*i.e.*, VBPR [8], MMGCN [37], LATTICE [43]) and their six variations equipped with our methods, on three datasets. Table 2 reports all the accuracy results in top-10/20 recommendation. Here, 'neg' refers to our method employing our negative sampling idea and 'mbpr' refers to our method employing multiple BPR losses. The best and the second-best recommendation accuracies on each dataset and the (original) competitor are shown in bold and underlined, respectively.

Table 2. Recommendation accuracies (%) of three state-of-the-art multimedia recommender systems and six variants, where each of our proposed methods (*i.e.*, a novel negative sampling method, neg, and multiple BPR losses, mbpr) orthogonally applied at each original method, respectively. 'gain' denotes the gains in accuracy of variants over the corresponding original method

Amazon Baby												
	Prec@10	gain	Recall@10	gain	NDCG@10	gain	Prec@20	gain	Recall@20	gain	NDCG@20	gain
LATTICE	0.537	-	5.112	-	2.846	-	0.420	-	7.975	-	3.601	-
LATTICE-neg	<u>0.543</u>	1.12	<u>5.170</u>	1.13	2.897	1.79	<u>0.426</u>	1.43	<u>8.098</u>	1.54	<u>3.669</u>	1.89
LATTICE-mbpr	0.554	3.17	5.281	3.31	<u>2.892</u>	1.62	0.433	3.10	8.231	3.21	3.670	1.92
MMGCN	<u>0.384</u>	-	3.638	-	1.906	-	<u>0.321</u>	-	<u>6.071</u>	-	2.548	-
MMGCN-neg	<u>0.384</u>	0.00	<u>3.653</u>	0.41	<u>1.931</u>	1.31	0.316	-1.56	6.007	-1.05	<u>2.551</u>	0.12
MMGCN-mbpr	0.456	18.75	4.334	19.13	2.329	22.19	0.364	13.40	6.917	13.94	3.011	18.17
VBPR	0.222	-	2.102	-	1.083	-	0.177	-	3.336	-	1.469	-
VBPR-neg	<u>0.226</u>	1.80	<u>2.139</u>	1.76	<u>1.165</u>	7.57	<u>0.180</u>	1.69	<u>3.393</u>	1.71	<u>1.501</u>	2.18
VBPR-mbpr	0.318	43.24	2.993	42.39	1.649	52.26	0.251	41.81	4.723	41.58	2.112	43.77

Amazon Men Clothing												
	Prec@10	gain	Recall@10	gain	NDCG@10	gain	Prec@20	gain	Recall@20	gain	NDCG@20	gain
LATTICE	0.415	-	4.136	-	<u>2.194</u>	-	0.309	-	6.160	-	2.705	-
LATTICE-neg	<u>0.417</u>	0.37	<u>4.157</u>	0.52	2.224	1.38	<u>0.317</u>	2.59	<u>6.332</u>	2.79	<u>2.765</u>	2.22
LATTICE-mbpr	0.418	0.67	4.168	0.76	2.224	1.38	0.321	4.04	6.414	4.12	2.794	3.29
MMGCN	0.270	-	2.694	-	1.328	-	0.223	-	4.447	-	1.769	-
MMGCN-neg	<u>0.272</u>	0.74	<u>2.712</u>	0.67	<u>1.337</u>	0.68	<u>0.234</u>	4.93	<u>4.659</u>	4.77	<u>1.826</u>	3.22
MMGCN-mbpr	0.329	21.85	3.283	21.86	1.665	25.38	0.268	20.18	5.342	20.13	2.183	23.40
VBPR	0.304	-	3.028	-	<u>1.590</u>	-	0.245	-	<u>4.894</u>	-	<u>2.061</u>	-
VBPR-neg	<u>0.307</u>	0.99	<u>3.050</u>	0.73	1.569	-1.32	<u>0.246</u>	0.41	4.885	-0.18	2.024	-1.80
VBPR-mbpr	0.380	25.00	3.791	25.20	1.917	20.57	0.300	22.45	5.980	22.19	2.469	19.80

Amazon Office												
	Prec@10	gain	Recall@10	gain	NDCG@10	gain	Prec@20	gain	Recall@20	gain	NDCG@20	gain
LATTICE	<u>1.109</u>	-	9.213	-	5.776	-	0.839	-	13.719	-	7.137	-
LATTICE-neg	1.108	-0.13	<u>9.215</u>	0.02	<u>5.783</u>	0.12	<u>0.842</u>	0.36	<u>13.739</u>	0.15	<u>7.158</u>	0.29
LATTICE-mbpr	1.134	2.25	9.451	2.58	5.854	1.35	0.859	2.38	14.061	2.49	7.253	1.63
MMGCN	0.616	-	5.077	-	2.963	-	0.532	-	8.726	-	<u>4.223</u>	-
MMGCN-neg	<u>0.637</u>	3.41	<u>5.143</u>	1.30	<u>3.007</u>	1.48	<u>0.545</u>	2.44	<u>8.814</u>	1.01	4.106	-2.77
MMGCN-mbpr	0.833	35.23	6.765	33.25	4.075	37.53	0.674	26.69	10.923	25.18	5.333	26.28
VBPR	<u>0.699</u>	-	<u>5.717</u>	-	<u>3.524</u>	-	<u>0.558</u>	-	9.072	-	<u>4.544</u>	-
VBPR-neg	0.694	-0.72	5.655	-1.08	3.430	-2.67	<u>0.558</u>	0.00	<u>9.088</u>	0.18	4.465	-1.74
VBPR-mbpr	0.792	13.30	6.366	11.35	3.932	11.58	0.623	11.65	9.874	8.84	5.029	10.67

In Table 2, we can see that the variations with the multiple BPR losses show the superiority over the original ones and those with our negative sampling, in all datasets and all models. Specifically, in the case of LATTICE [43], as it is the most recent and best performing method, its variation applied with our negative sampling outperforms the original method by up to 2.79% (see Amazon Men Clothing) and the variation applied with the multiple BPR losses outperforms the original method by up to 4.12% (see Amazon Men Clothing), both in terms of Recall@20. In the case of MMGCN [37], the variation applied

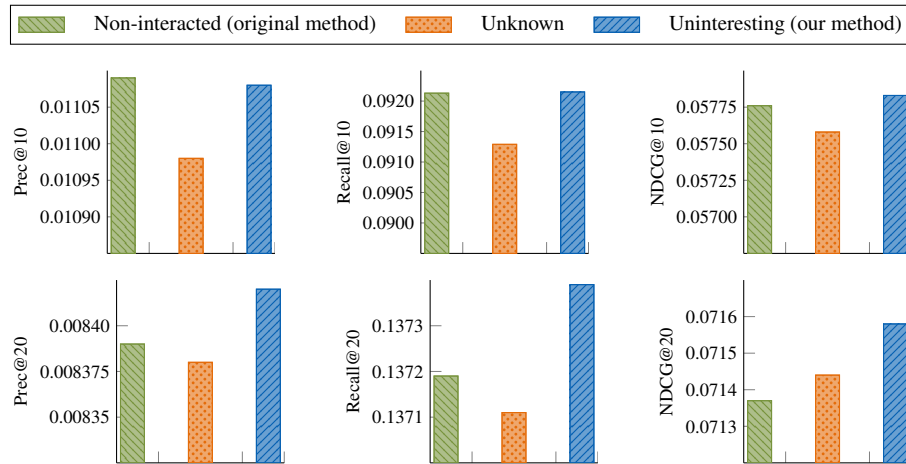


Fig. 3. Recommendation accuracies of a best state-of-the-art multimedia recommender system (*i.e.*, LATTICE [43]) and two variants equipped with two different cases of negative sampling methods, respectively. 'Random' indicates the method of using randomly chosen non-interacted items as negative samples (*i.e.*, the original negative sampling method), 'Unknown' indicates the method of using unknown items as negative samples, and 'Uninteresting' indicates the method of using uninteresting items as negative samples (*i.e.*, our proposed negative sampling method)

with our negative sampling outperforms the original method by up to 4.93% in terms of Prec@20 (see Amazon Men Clothing) and the variation applied with our multiple BPR losses outperforms the original method by up to 37.53% in terms of NDCG@10 (see Amazon Office). Lastly, in the case of VBPR [8], the variation applied with our negative sampling outperforms the original method by up to 7.53% in terms of NDCG@10 (see Amazon Baby) and the variation applied with the multiple BPR losses outperforms the original method by up to 52.26% in terms of NDCG@10 (see Amazon Baby).

Based on the results above, we have confirmed that i) employing the notions of unknown and uninteresting items (instead of the non-interacted items) in training the model is effective in terms of recommendation accuracy and ii) employing multiple BPR losses over interesting, unknown, and uninteresting items is more effective than a single BPR loss over interesting and non-interacted items in terms of recommendation accuracy in training the model.

EQ2: Is the idea of selecting the uninteresting items as negative samples most effective for improving the recommendation accuracy? To verify the effectiveness of our negative sampling method, we compare the recommendation accuracy of the following three cases: sampling negative items randomly i) from the non-interacted items (*i.e.*, original method), ii) from unknown items, and iii) from uninteresting items (*i.e.*, our proposed method). Figure 3 shows the recommendation accuracy of the three negative sam-

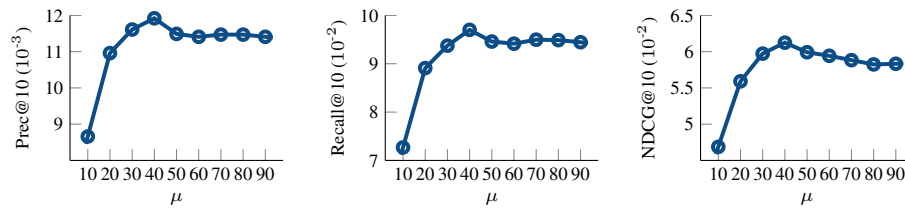


Fig. 4. The effect of μ on recommendation accuracies

pling methods on the Amazon Office dataset with LATTICE [43].⁵ Here, 'Non-interacted' refers to case i), 'Unknown' refers to case ii), and 'Uninteresting' refers to case iii).

In Figure 3, we see that our method (*i.e.*, only using uninteresting items) outperforms the original negative sampling method. The method of using unknown items as negative samples shows very poor recommendation accuracy. This indicates that, as unknown items could be the items that the users' preferences are high, they should not be used in the training process as negative samples. When randomly sampling the users' non-interacted items as negative samples, unknown items might be included as negative samples, thus likely to confuse the model in training. Therefore, this result validates that selecting negative samples from uninteresting items helps improve the accuracy in multimedia recommendation.

EQ3: How sensitive is the recommendation accuracy of the multiple BPR losses to different hyperparameter values? For our multiple BPR losses, we consider two types of hyperparameters. First, μ is to determine the ratio of uninteresting items to non-interacted items. Second, the weights α , β , and γ for different BPR losses to indicate the importance in training. Regarding the hyperparameters, we conducted experiments to answer the following two sub-questions:

- **EQ3-1:** How sensitive is the accuracy from employing the multiple BPR losses to the ratios of uninteresting and unknown items out of non-interacted items?
- **EQ3-2:** How sensitive is the accuracy from employing the multiple BPR losses to the weight for each BPR loss?

EQ3-1: Sensitiveness of hyperparameter μ . We analyze how the recommendation accuracy changes with different values of $\mu \in \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$ on the Amazon Office dataset with LATTICE [43]. Figure 4 shows the recommendation accuracy with different values of μ . As shown in Figure 4, we observe that the recommendation accuracy increases until μ increases to 40 and then decreases. The result shows that, if μ is set as too small (resp. large), some uninteresting (resp. unknown) items might be misclassified as unknown (resp. uninteresting) items, which causes the model to be confused in the training process. Therefore, the proper setting of μ allows the model to be better learned and provides a more-effective recommendation result. Based on this observation, we set μ as 40% for our proposed multiple BPR losses.

⁵ For EQ2 and EQ3, the tendencies of recommendation accuracy on other datasets with other competitors are all similar; so, we only include the results on Amazon Office with LATTICE, the latest and most powerful method.

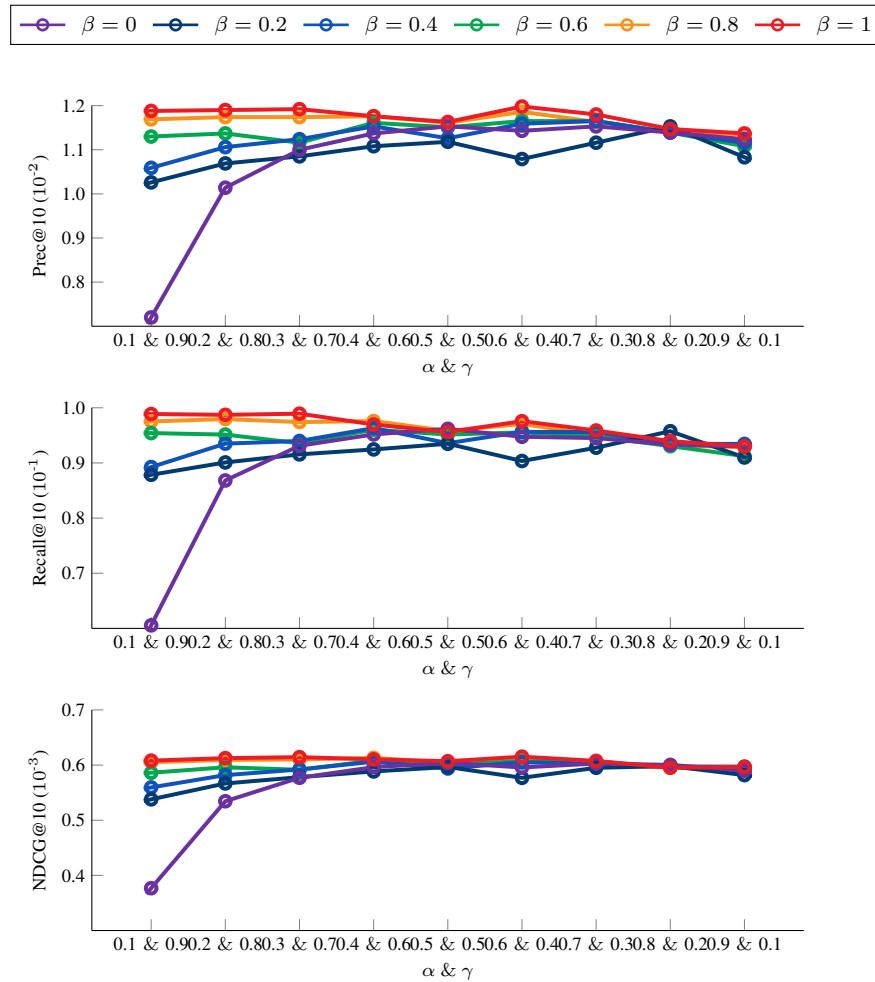


Fig. 5. The effect of α , β , and γ on recommendation accuracies

EQ3-2: Sensitiveness of hyperparameters α , β , and γ . We analyze the change of recommendation accuracy with varying the values of α , β , and $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ on the Amazon Office dataset with LATTICE [43]. Figure 5 illustrates the recommendation accuracy with different values of α , β , and γ . The recommendation accuracy becomes highest when $\alpha = 0.4$, $\beta = 1.0$, and $\gamma = 0.6$ in terms of Prec@10. The result shows that bigger the value of β , higher the recommendation accuracy regardless of the values of α and γ . Also, the result shows that the recommendation accuracy overall shows robustness regardless of the values of α and γ . Therefore, based on this result, we set $\alpha = 0.4$, $\beta = 1.0$, and $\gamma = 0.6$, in the previous experiments.

The experimental results can be summarized as follows: i) applying concept(s) of unknown and uninteresting items helps to improve the recommendation accuracy of multimedia recommender systems; ii) selecting the uninteresting items as negative samples is more effective in improving the recommendation accuracy than selecting random (original negative sampling method) or unknown items; iii) utilizing both unknown and uninteresting items (*i.e.*, all non-interacted items) in multimedia recommender systems significantly improves most of their original recommendation accuracies, also this method (*i.e.*, our method) is easily and orthogonally applicable to any multimedia recommender systems.

5. Conclusions

In this paper, we have pointed out the limitation of existing multimedia recommender systems that they do not fully exploit the characteristics of non-interacted items for users. Then, we proposed two methods to alleviate the limitation, thereby enabling existing systems to exploit the non-interacted items of users appropriately by classifying the non-interacted items into unknown and uninteresting items. Specifically, our first idea is to allow only the items highly likely not to be preferred by a user as negative items during training the recommender model. Further, our second idea is to use the multiple BPR losses, which makes possible rank discrepancies among positive, unknown, and uninteresting items learned correctly in the training process. Extensive experiments on three real-world Amazon datasets validate that our proposed methods outperform three state-of-the-art multimedia recommender systems and that our ideas are all effective in improving recommendation accuracy.

Acknowledgments. This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2022 (Project Name: Development of Intelligent Personalized Rehabilitation Service Technology, Project Number: SR202104001, Contribution Rate: 100%).

References

1. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: International Conference on Engineering and Technology (ICET). pp. 1–6 (2017)
2. Bao, Y., Fang, H., Zhang, J.: Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In: AAAI. pp. 2–8 (2014)

3. Chae, D., Kim, J., Chau, D.H., Kim, S.: AR-CF: Augmenting virtual users and items in collaborative filtering for addressing cold-start problems. In: ACM SIGIR. pp. 1251–1260 (2020)
4. Chen, X., Chen, H., Xu, H., Zhang, Y., Cao, Y., Qin, Z., Zha, H.: Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In: ACM SIGIR. pp. 765–774 (2019)
5. Cohen, R., Shalom, O.S., Jannach, D., Amir, A.: A black-box attack model for visually-aware recommender systems. In: ACM WSDM. pp. 94–102 (2021)
6. Goldberg, D., Nichols, D.A., Oki, B.M., Terry, D.B.: Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35(12), 61–70 (1992)
7. Graves, A., Mohamed, A., Hinton, G.E.: Speech recognition with deep recurrent neural networks. In: IEEE ICASSP. pp. 6645–6649 (2013)
8. He, R., McAuley, J.: VBPR: visual bayesian personalized ranking from implicit feedback. In: AAAI. pp. 144–150 (2016)
9. He, X., Du, X., Wang, X., Tian, F., Tang, J., Chua, T.: Outer product-based neural collaborative filtering. In: IJCAI. pp. 2227–2233 (2018)
10. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. *CoRR* abs/1511.06939 (2015)
11. Hwang, W., Parc, J., Kim, S., Lee, J., Lee, D.: “Told you I didn’t like it”: Exploiting uninteresting items for effective collaborative filtering. In: IEEE ICDE. pp. 349–360 (2016)
12. Kim, D.H., Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation. In: ACM RecSys. pp. 233–240 (2016)
13. Kim, T., Kim, Y., Lee, Y.C., Shin, W., Kim, S.: Is it enough just looking at the title?: Leveraging body text to enrich title words towards accurate news recommendation. In: ACM CIKM. pp. 4138–4142 (2022)
14. Kim, T., Lee, Y.C., Shin, K., Kim, S.: Mario:Modality-Aware attention and modality-preserving decoders for multimedia recommendation. In: ACM CIKM. pp. 993–1002 (2022)
15. Ko, Y., Yu, J.S., Bae, H.K., Park, Y., Lee, D., Kim, S.W.: Mascot: A quantization framework for efficient matrix factorization in recommender systems. In: IEEE ICDM. pp. 290–299 (2021)
16. Kong, T., Kim, T., Jeon, J., Choi, J., Lee, Y.C., Park, N., Kim, S.W.: Linear, or non-linear, that is the question! In: ACM WSDM. pp. 517–525 (2022)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60(6), 84–90 (2017)
18. Lee, J., Hwang, W., Parc, J., Lee, Y., Kim, S., Lee, D.: l-injection: Toward effective collaborative filtering using uninteresting items. *IEEE TKDE* 31(1), 3–16 (2019)
19. Lee, Y., Kim, S., Lee, D.: gOCCF: Graph-theoretic one-class collaborative filtering based on uninteresting items. In: AAAI. pp. 3448–3456 (2018)
20. Lee, Y.C., Kim, T., Choi, J., He, X., Kim, S.: M-bpr: A novel approach to improving bpr for recommendation with multi-type pair-wise preferences. *Information Sciences* 547, 255–270 (2021)
21. Lee, Y., Kim, S., Park, S., Xie, X.: How to impute missing ratings?: Claims, solution, and its application to collaborative filtering. In: WWW. pp. 783–792 (2018)
22. Lim, H., Lee, Y.C., Lee, J.S., Han, S., Kim, S., Jeong, Y., Kim, C., Kim, J., Han, S., Choi, S., Ko, H., Lee, D., Choi, J., Kim, Y., Bae, H.K., Kim, T., Ahn, J., You, H.S., Kim, S.W.: Airs: A large-scale recommender system at naver news. In: IEEE ICDE. pp. 3386–3398 (2022)
23. Liu, F., Cheng, Z., Sun, C., Wang, Y., Nie, L., Kankanhalli, M.: User diverse preference modeling by multimodal attentive metric learning. In: ACM MM. pp. 1526–1534 (2019)
24. Liu, Q., Wu, S., Wang, L.: Deepstyle: Learning user preferences for visual recommendation. In: ACM SIGIR. pp. 841–844 (2017)
25. Niu, W., Caverlee, J., Lu, H.: Neural personalized ranking for image recommendation. In: ACM WSDM. pp. 423–431 (2018)
26. O’Shea, K., Nash, R.: An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458* (2015)

27. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R.M., Scholz, M., Yang, Q.: One-class collaborative filtering. In: IEEE ICDM. pp. 502–511 (2008)
28. Park, S.J., Chae, D.K., Bae, H.K., Park, S., Kim, S.W.: Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In: ACM WSDM. pp. 784–793 (2022)
29. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
30. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI. pp. 452–461 (2009)
31. Ricci, F., Rokach, L., Shapira, B. (eds.): Recommender Systems Handbook. Springer (2015)
32. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. pp. 285–295 (2001)
33. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: Autoencoders meet collaborative filtering. In: WWW. pp. 111–112 (2015)
34. Tang, J., Wang, K.: Personalized top-N sequential recommendation via convolutional sequence embedding. In: ACM WSDM. pp. 565–573 (2018)
35. Wang, S., Wang, Y., Tang, J., Shu, K., Ranganath, S., Liu, H.: What your images reveal: Exploiting visual contents for point-of-interest recommendation. In: WWW. pp. 391–400 (2017)
36. Wei, Y., Wang, X., Nie, L., He, X., Chua, T.: Graph-refined convolutional network for multimedia recommendation with implicit feedback. In: ACM MM. pp. 3541–3549 (2020)
37. Wei, Y., Wang, X., Nie, L., He, X., Hong, R., Chua, T.: MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In: ACM MM. pp. 1437–1445 (2019)
38. Wu, C., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. In: ACM WSDM. pp. 495–503 (2017)
39. Xue, H., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: IJCAI. pp. 3203–3209 (2017)
40. Yao, W., He, J., Wang, H., Zhang, Y., Cao, J.: Collaborative topic ranking: Leveraging item meta-data for sparsity reduction. In: AAAI. pp. 374–380 (2015)
41. Ying, H., Chen, L., Xiong, Y., Wu, J.: Collaborative deep ranking: A hybrid pair-wise recommendation algorithm with implicit feedback. In: PAKDD. pp. 555–567 (2016)
42. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
43. Zhang, J., Zhu, Y., Liu, O., Wu, S., Wang, S., Wang, L.: Mining latent structures for multimedia recommendation. In: ACM MM. pp. 3872–3880 (2021)
44. Zhang, Y., Ai, Q., Chen, X., Croft, W.B.: Joint representation learning for top-n recommendation with heterogeneous information sources. In: ACM CIKM. pp. 1449–1458 (2017)

Jiyeon Kim received her bachelor’s degree in International studies (B.A.) and Computer Science (B.S.) from Hanyang university, Seoul, Korea in 2020. She received her master’s degree in computer science from Hanyang University, Seoul, Korea, in 2022. Her research interests include data mining, recommender systems and machine learning.

Taeri Kim received a B.S. degree in Multimedia Engineering from Hansung University, Seoul, Korea, in 2015. Since 2017, she has been pursuing a Ph.D. degree in Computer Science at Hanyang University. Her current research interests include data mining, graph neural network analysis, and recommendation.

Sang-Wook Kim received the B.S. degree in computer engineering from Seoul National University, in 1989, and the M.S. and Ph.D. degrees in computer science from the Korea

Advanced Institute of Science and Technology (KAIST), in 1991 and 1994, respectively. From 1995 to 2003, he served as an associate professor with Kangwon National University. In 2003, he joined Hanyang University, Seoul, Korea, where he currently is a professor in the Department of Computer Science and the director of the BrainKorea-21-FOUR research program. He is also leading a National Research Lab (NRL) Project funded by the National Research Foundation since 2015. From 2009 to 2010, he visited the Computer Science Department, Carnegie Mellon University, as a visiting professor. From 1999 to 2000, he worked with the IBM T. J. Watson Research Center, USA, as a postdoc. He also visited the Computer Science Department of Stanford University as a visiting researcher in 1991. He is an author of more than 200 papers in refereed international journals and international conference proceedings. His research interests include databases, data mining, multimedia information retrieval, social network analysis, recommendation, and web data analysis. He is a member of the ACM and the IEEE.

Received: October 31, 2022; Accepted: December 29, 2022.

Class Probability Distribution Based Maximum Entropy Model for Classification of Datasets with Sparse Instances

Saravanan Arumugam*, Anandhi Damotharan, and Srividya Marudhachalam

Department of Computing, Coimbatore Institute of Technology
Coimbatore, Tamil Nadu, India – 641014
a.saravanan21@gmail.com
anandhi.cit@gmail.com
msrividya2013@gmail.com

Abstract. Due to the digital revolution, the amount of data to be processed is growing every day. One of the more common functions used to process these data is classification. However, the results obtained by most existing classifiers are not satisfactory, as they often depend on the number and type of attributes within the datasets. In this paper, a maximum entropy model based on class probability distribution is proposed for classifying data in sparse datasets with fewer attributes and instances. Moreover, a new idea of using Lagrange multipliers is suggested for estimating class probabilities in the process of class label prediction. Experimental analysis indicates that the proposed model has an average accuracy of 89.9% and 86.93% with 17 and 36 datasets. Besides, statistical analysis of the results indicates that the proposed model offers greater classification accuracy for over 50% of datasets with fewer attributes and instances than other competitors.

Keywords: classification, fewer attributes and instances, Lagrange multipliers, class probability distribution, relative gain, maximum entropy.

1. Introduction

In this digital era, data mining has become an inevitable technique and a milestone in technological development. It is applied to a wide range of historical data to extract useful information that helps to make decisions effectively [1]. It covers other important areas like machine learning, statistics and the database management system. It is extremely influential and even changed the perspective of handling business. Although it was originally used to develop the business, later it seems to be an inseparable technique in almost every area [2]. It focuses on extracting various pieces of knowledge from the vast amount of data. This can be achieved by several data mining functions such as classification, association rule mining, prediction, outlier and cluster analysis and pattern recognition. Nevertheless, classification and prediction have become the two major pillars of data mining [3].

* Corresponding author

Classification and prediction are the most common methods that researchers use in all areas to find solutions to various problems. A few of the domain applications where classification and prediction are used frequently include the educational field (students' performance classification, result prediction) [4], bank and financial sectors (customers classification based on their credit risk, fraud detection) [5], health care industries (diagnosing the disease based on the past data containing symptoms) [6], agricultural field (analysing soil nutrients and crop prediction) [7], retail industries (customer churn and sales prediction) [8], classifying spam or junk emails [9], weather forecasting and rainfall prediction [10], predicting current behaviour by analyzing the human activities [11], classifying customer segment [12], classifying attack traffic from normal network traffic [13], software defect prediction [14] and even more.

Generally, classification outcomes are often influenced by the quality of input data. Pre-processing of input data is carried out before applying the classification model to improve its prediction accuracy. The data can be preprocessed by removing missing data and normalizing the attribute values along with the feature selection process [15]. Feature selection aims at selecting the relevant study-related attributes for the target class. In general, classification models can be based on machine learning or statistical models [16]. The machine learning based models include decision trees (DT), random forest (RF), artificial neural networks (ANN), k nearest neighbour (KNN), case based reasoning (CBR), support vector machines (SVM), AdaBoost, Stochastic gradient descent (SGD), other ensemble and boosting classifiers. The statistical model includes linear and logistical regression and naïve Bayesian classifiers. These models are currently available and many more new models were also suggested by the various researchers. However, most of these existing models are an extension of specific conventional models designed for specific applications.

The type of data to be classified, such as categorical data, real or integer-valued data, typically affects the performance of the classification model. Some algorithms are only suitable for certain types of data like logistic regression cannot manage huge categorical data. In addition to data types, the size of attributes and instances plays a crucial role in the accuracy of classification. If models are not chosen based on the analysis of the adequacy and applicability of the specific characteristics of the datasets, there is a greater possibility of classification error. Moreover, some classifiers classify the data with appropriate results, but with greater computational complexity. Consequently, classification models should be constructed by considering various other characteristics of the underlying datasets.

This paper provides a simple statistical classification model that is appropriate for datasets with fewer attributes and instances. It utilizes the novel idea of using Lagrange multipliers on the class probabilities that is suitable for the classification of samples in small datasets. The proposed class probability distribution based maximum entropy classifier works as follows. To begin with, the dataset is subjected to feature selection and data pre-processing to improve the dataset's data quality and classification accuracy. During the training phase, the datasets are categorized according to the labels assigned to the target classes. Then, for each chosen attribute, the average class relative distance is estimated for the training samples, from which the attribute relative gain is calculated for the given test sample. The Lagrange multipliers are applied and evaluated to assess the class probabilities of the attribute by maximizing the entropy. Finally, the class probabilities of each attribute are aggregated to predict the class label for the given test

instance. An extensive experimental analysis is also made to examine the performance and effectiveness of the proposed model.

The organization of the paper is as follows. Section 2 presents the works from the literature that are related to the proposed study. Section 3 discusses the study background. Section 4 describes the proposed class probability distribution based maximum entropy model for classification. The overall framework, algorithm pseudocode and working procedure with an illustration are presented in sub-sections. Section 5 presents the various experimental analysis, results obtained for the proposed model and research findings from the statistical analysis. Finally, the paper is concluded by listing out the scope for future enhancements.

2. Related Works

Owing to the widespread use of data mining and other machine learning techniques, classification models are evolving day by day. Several classification models and their variations were proposed in the literature by the researchers. For easy understanding, the existing classifiers that are related to the study are clustered under two groups. The first category is the standard classifiers that are significant and widely used in classification problems and the second category is the new existing state-of-the-art classifiers that are developed recently yet to be researched further. These categories are presented in this section.

2.1. Standard Classifiers

To properly categorize unlabeled data, the majority of supervised learning algorithms use statistical analysis of the training set in one way or another. Among these classifiers, KNN, Naïve Bayes (NB), Logistic regression (LR) and Decision trees use statistical inference to classify the data. A univariate location estimator, termed proximity based KNN classifier was a simple classic classification model proposed for estimating regression curve. In this model, the classification results of the given test data point are the closest point among a given set of data points [17]. In general, KNN classifier is computationally inefficient and challenging to determine the right k value, even though it is most frequently employed in numerous applications with numerous variants [18].

Naïve Bayes classifiers are probabilistic model that applies the Bayes theorem to predict the class probability of the given instance [19]. The main drawback of this model is that the model treats each attribute independently and so cannot identify the relationship between the attributes. Nonetheless, the model is still frequently utilized in various applications because of its efficient performance [20]. Logistic regression is another statistical model that applies a logistic function for modelling the dependent variable using independent variables [21]. The model is sensitive to overfitting and cannot be used for non-linear problems or when the number of instances is less than the number of attributes. Decision trees are another type of classification model that makes use of the gain of an attribute at each precedent node [22]. Numerous types of trees exist

such as ID3, CART and C4.5 among which C4.5 [23] offers better results. However, the decision trees anticipate poor results with small datasets and cause overfitting.

For the datasets with high dimensions or when the number of attributes is greater than the number of instances, SVM offers improved results and so it is widely popular among various fields [24], [25]. However, the SVM is not suitable for non-linear problems. Alternatively, Sparse Representation based Classification (SRC) [26] is another classifier model that offers better performance. However, SRC is more suitable for multimedia datasets involving image, audio and video data.

Ensemble classifiers are another milestone in the classification model. It utilizes two or more classifiers to classify the data and the results are combined using schemes such as majority voting or weighting technique [27]. The ensemble learners can use various techniques such as boosting, bagging or stacking to convert weak learners to strong learners. Algorithms such as AdaBoost (AB) and Gradient Boosting use boosting to reduce the bias between various models used [28]. Random forest algorithm employs bootstrap aggregation (bagging) to reduce the variance [29] or stacking [30] to increase the prediction. As gradient boosting interprets the boosting as an optimization, Stochastic Gradient Boosting Decision Trees (GBDT) apply regression to the gradient boosting algorithm [31]. Though ensemble learners offer better accuracy it is less widely used due to their increased time complexity.

Not only machine learning techniques but artificial intelligence models were also incorporated for the classification of test instances. Artificial neural network (ANN) is widely adapted in classification inspired by the neural networks in the animal brain. These models are specifically designed to recognize patterns [32]. Similarly, Deep Learning (DL), a model that mimics the working of the human brain in recognizing patterns was proposed specifically for making decisions [33], [34]. Extreme Learning Machine (ELM), a feedforward neural network utilizes single-layer feed-forward neural networks [35]. These models offer better classification accuracy in minimum time than other traditional neural networks such as backpropagation. Still, the models are the least widely used since SVM outperforms them in various cases.

Several analyses were made in the literature to examine the performance of the conventional classifiers. An analysis was made using several machine learning classifiers such as NB, Bayesian networks, J48, RF, multilayer perceptron (MLP), and LR to identify the better classifiers [36]. This study with the credit risk dataset indicates that the RF produce improved performance than others. Similar analysis was carried out for SVM, KNN, Gradient boosting, decision tree, RF and LR on diabetes datasets [37]. The results indicate that RF outperforms the other 6 classification algorithms with many of the evaluation metrics. An analysis of the performance of the classification algorithms such as ELM, SRC, DL, GBDT, SVM, RF, C4.5, KNN, LR, AB, and NB on various datasets was evaluated. The result outcomes are surprising that GBDT offers better results across various datasets than SVM and RF [38]. Most of the classification algorithms or the comparative studies found in the literature are specific to a particular application. Though the models were proved to be effective, the results may not be same for all the applications. Thus, lead to performance degradation for other applications or different datasets having different attribute types for the same applications [39].

2.2. State-of-the-Art Classifiers

Several researchers had contributed more on classification problems with various new probabilistic models for different data types. The use of various probabilistic models such as multinomial Bernoulli assuming naive Bayes [40], the combination of Expectation-Maximization (EM) and NB classifier [41], and generative/discriminative model [42] were found in the literature. The detailed study of these probabilistic models shows the performance improvement over text datasets than other data types. The use of conditional random fields based on a probabilistic model was proposed to segment and label the data. However, it was only evaluated sequence data and evidenced to have improved classification accuracy [43].

Many instance based classifiers attained a notable position in the literature. Data Gravitation based Classification (DGC) makes the comparison between the data gravitation and distinct classes for classifying the given input record [44]. This work was extended by adding weights to the data gravitation (DGC+) [45]. Despite the improved accuracy, the models undergo high computational complexity. Another classification model that computes the average weighted pattern score (AWPS) to classify the given data using attribute rank based feature selection was proposed [46]. The comprehensive analysis of the study indicates that the model is suitable for imbalanced datasets and yet the results are not accurate for low dimensional space.

An instant based classifier termed attribute value frequency based instance weighted naive Bayes (AVFWNB) was proposed [47]. In this model, the weights are assigned for the training sets that offer good results than traditional NB. Similarly, a simple model that is a variation of NB called correlation based attribute weighted naive Bayes (CAWNB) was proposed. The model aims at assigning weights for the attributes based on the dependency between the attribute and the class [48]. Moreover, the weights are verified using sigmoid transformation. The results of CAWNB proved to be effective with improved classification accuracy than NB. Inspired by AVFWNB and CAWNB models, a unique model that assigns weight for instances and attributes was proposed recently. This model utilizes two approaches eager learners (AIWNB^E) aa lazy learners (AIWNB^L) for implementing instance weights [49]. The performance of these classifiers highly depends on the how accurately the weights are assigned to the instances and attributes.

Discriminatively weighted naive Bayes (DWNB) and eager learning approach was opposed that iteratively re-assigns the weights by computing the conditional probability loss. Though the model seems have effective performance in terms of accuracy, the model needs more iterations to improve efficiency in assigning weights [50]. A model that computes the weights for the instances and attributes collaboratively was proposed. The model utilizes posterior probability loss to compute the weights and is termed as collaboratively weighted naive Bayes (CWNB) [51]. An instance weighted hidden naive Bayes (IWHNB) was proposed that integrates the instance weight with a hidden naive Bayes model for computing probabilities [52]. For all these weight assignment based classifiers, the optimization in assigning weights to the instances and attributes is to be incorporated for ensuring effective performance. Moreover, in all these methods, the authors show improved performance than existing models, yet the accuracy still needs improvement. The summary of the significant existing classifiers is presented in Table 1.

Table 1. Summary of Existing State-of-the-Art Classifiers

Model	Authors	Approach	Merits	Drawback
Data gravitation classification (DGC)	Peng et al., (2009) [44]	Classifies the instances by comparing the data gravitation between the different data classes	Simple and effective to implement	Reduction in accuracy when the points are away from centroid and class borders
Extended data gravitation classification (DCG+)	Cano et al., (2013) [45]	Assigns matrix of weights for attributes based on its significance in each class	Improved accuracy	High computational complexity
Discriminatively weighted naive Bayes (DWNB)	Jiang et al., (2012) [50]	Iteratively the weights are re-assigned based on conditional probability loss	Eager learning approach	Needs more iterations
Average weighted pattern score based classification (AWPS)	Sathya Bama and Saravanan., (2019) [46]	Feature selection with and classification using average weighted pattern score	Simple and outperforms many existing classifiers	Not accurate for low dimensional datasets
Correlation-based attribute weighted naive Bayes (CAWNB)	Jiang et al., (2018) [48]	Attributes weight are assigned by computing the difference between attribute-class correlation and attribute-attribute redundancy	Better than NB and simple to implement	Need more time to compute similarity between the attributes in high dimensional space
Attribute value frequency-based instance weighted naive Bayes (AVFWNB)	Xu et al., (2019) [47]	Instance weights are assigned based on attribute value frequency and attribute value number	Better than NB and simple to implement	Low performance on datasets with high dimensions
Attribute and instance weighted naive Bayes (AIWNB)	Zhang et al., (2021) [49]	Weights for instance is assigned based on the distribution of the instance	Applies both lazy and eager approach for assigning weights	Accuracy depends on weight assignment
Collaboratively weighted naive Bayes (CWNB)	Zhang et al., (2021) [51]	Optimal weights for the instance are computed by maximizing conditional log-likelihood with prior and conditional probabilities	More accurate than Naïve Bayes and other similar models	High computational complexity in assigning weights for the instances
Instance weighted hidden naive Bayes (IWHNB)	Yu et al., (2021) [52]	Integrates the instance weighting with improved Hidden naïve Bayes model for computing probability estimates	Better than NB and Low time complexity	No optimization in assigning weight for the instances

3. Study Background

3.1. Attribute Rank based Feature Selection

The attribute rank based feature selection algorithm is a simple probabilistic method that makes use of probability based attribute scores for their contribution toward better classification. The relevant features that are significant for the classification are selected by computing the attribute rank based on the distinct attribute values present in the training set.

Initially, the model computes the overall database score based on the class labels as in Eq. (1) where p_i is the probability that an arbitrary instance in D belongs to class C_i .

$$Score(D) = Avg \left(\sum_{i=1}^m p_i^{p_i} \right) \tag{1}$$

$$p_i = \frac{\text{Number of instances belonging to } C_i}{\text{Total number of instances in } D} \tag{2}$$

The attribute score for each attribute having n distinct values can be computed by grouping the tuples based on n distinct values as $\{G_1, G_2, \dots, G_j\}$. The count of tuples in each group is represented as $\{n_1, n_2, n_3, \dots, n_j\}$. The calculation of the attribute score A_{score} is given in Eq. (3).

$$A_{score} = Score(D) - Avg \left(\sum_{j=1}^n p(n_j) \times Score(G_j) \right) \tag{3}$$

where $p(n_j)$ is the probability that an arbitrary instance in G_j belongs to class C_i . Finally, the score is calculated and the rank is allocated for each attribute. The ranks are then converted to rank scores using the rank sum method. The attributes having a rank score higher than the specified threshold value are then selected for the further classification process. A detailed illustration of attribute selection is discussed in [46].

3.2. Lagrange Multipliers

Shannon entropy computes the entropy of a random variable and it specifies the amount of information or the uncertainty in the variable [53]. Consider the random variable A with n possible outcomes as $\{A_1, A_2, \dots, A_n\}$ that occur with the probability $\{P(A_1), P(A_2), \dots, P(A_n)\}$. Then the entropy of the variable A can be identified as in Eq. (4).

$$S = - \sum_{i=1}^n p(A_i) \log_2 p(A_i) \tag{4}$$

However, in a more uncertain situation, the entropy value will be higher and it leads to chaos. Thus, to solve this problem effectively, Lagrange multipliers with maximum entropy can be applied. In simple words, maximum entropy allows choosing the best value from the number of the probability distribution that specifies the knowledge at the

current state [54]. Maximum entropy is a powerful probabilistic model that has wide usage in the classification of data in different datasets such as text [55], image [56], audio [57] and video [58]. To solve using Lagrange multipliers, several constraints are to be taken into account.

For a random variable A , each possible outcome A_i has some probability of occurrence $p(A_i)$ where i represents the index representing possible outcomes. Generally, the probability distribution of a variable $p(A)$ has specific constraints such as (a) the probability of occurrence of each outcome $p(A_i)$ always lies between 0 and 1 and (b) the sum of the probability of occurrence of all outcomes is 1 and is represented in Eq. (5).

$$1 = \sum_i p(A_i) \quad (5)$$

For framing the next constraint, the expected value of the variable is computed by averaging the values corresponding to each outcome and its probabilities. Therefore, for the quantity G with the value $g(A_i)$ for each outcome, the probability distributions having the expected value G will be considered. However, the value of G always lies between the smallest $g(A_i)$ and the largest $g(A_i)$ and the constraint is given in Eq. (6).

$$G = \sum_i g(A_i)p(A_i) \quad (6)$$

4. Proposed Class Probability Distribution based Maximum Entropy (CPDME)

The proposed class probability distribution based maximum entropy model anticipates to classify the instances of sparse datasets having a minimum number of attributes and instances. The overall framework of the proposed class probability distribution based maximum entropy classification model (CPDME) is depicted in Fig. 1. The model is subdivided into four phases: 1) data pre-processing and feature selection, 2) relative distance computation, 3) attribute probability computation and 4) class probability based classification. Data pre-processing is an inevitable step in data mining that transforms incomplete raw data into a complete format that is suitable for mining [59]. In the proposed model, the missing and incomplete records are processed using predictive mean imputation [60]. Further, the data is transformed using data discretization [61] and min-max normalization [62]. To achieve feature selection, the model employs an attribute rank based feature selection (ARFS) which has been discussed in section 3.1.

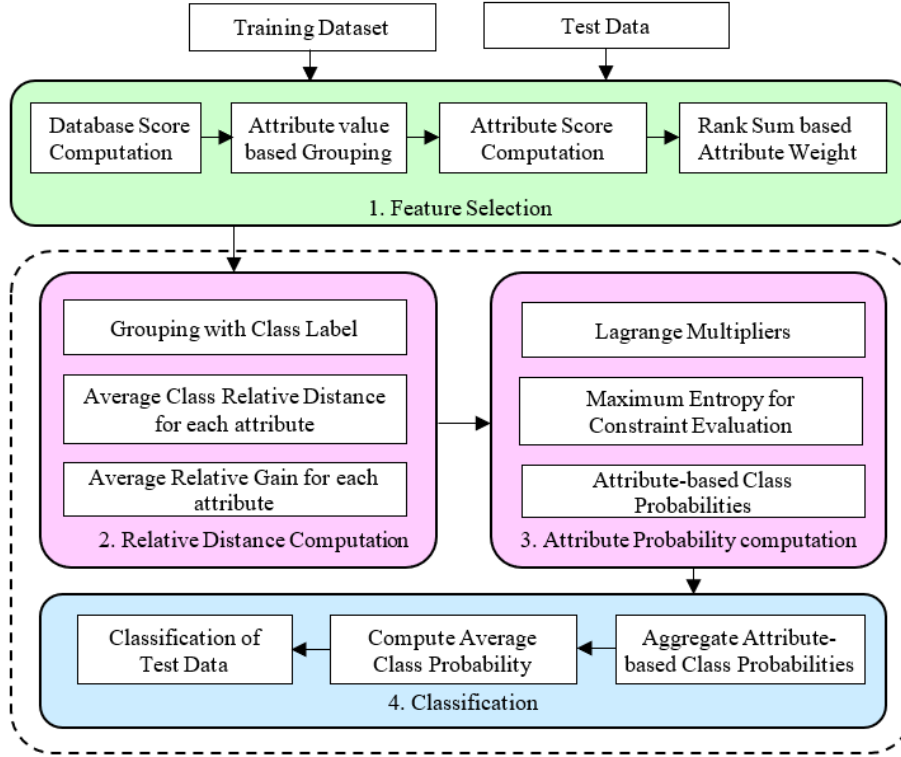


Fig. 1. Overall Framework of the Proposed Class Probability Distribution based Maximum Entropy Model

4.1. Relative Distance Computation

To compute the relative distance, the class relative distance and the relative gain are evaluated. Primarily, the set of training records is grouped based on the class label i where i vary from 1 to n . The average class relative distance $g(A_i)$ is computed for each attribute A concerning each class i as in Eq. (7).

$$g(A_i) = |C(A_i) - A_t| \tag{7}$$

Thus, the value of $g(A_i)$ is computed by finding the relative distance between the value of an attribute A of the test sample represented as A_t with the centroid of the attribute value of all the training records belonging to each class i represented as $C(A_i)$. Here the centroid of the attribute value of all training records belonging to the class label i is the mean value of attribute A of i^{th} class. The centroid is computed as in Eq. (8) in which m represents the number of records in each class.

$$C(A_i) = \frac{\sum_{j=1}^m A_i^j}{m} \tag{8}$$

Upon computing the value for $g(A_i)$ for each class i , the value of average relative gain G is computed by averaging the distance between each class $g(A_i)$ with all the other

classes. The formula to compute the relative gain G of each attribute is presented in Eq. (9).

$$G = \frac{\sum_{j=1}^n \sum_{k=j+1, j < k}^n |g(A_i^j) - g(A_i^k)|}{n(n-1)/2} \tag{9}$$

4.2. Attribute Probability Computation

To compute the attribute probability, the maximum entropy principle has been extended to the larger system using Lagrange multipliers. Lagrange multipliers are named after the French mathematician, Joseph-Louis Lagrange [63]. Instead of processing the constraint equation to reduce the variables, Lagrange augments two more unknown variables α and β termed Lagrange multipliers. The Lagrange method assumes Maximum Entropy. Thus, the Lagrange function L can be defined by using the constraints given in Eq. (5) and Eq. (6) as in Eq. (10).

$$L = S - (\alpha - \log_2 e) \left(\sum_i p(A_i) - 1 \right) - \beta \left(\sum_i g(A_i) p(A_i) - G \right) \tag{10}$$

Here, L can be maximized for each $p(A_i)$ and is made by differentiating L concerning one of $p(A_i)$ with α , β , and other $p(A_i)$ as constant. The resulting functions are given in Eq. (11) and Eq. (12).

$$\log_2 \frac{1}{p(A_i)} = \alpha + \beta g(A_i) \tag{11}$$

$$p(A_i) = 2^{-\alpha} 2^{-\beta g(A_i)} \tag{12}$$

The values of α and β can be computed from the above equation specified for $p(A_i)$ and the results are shown in Eq. (13) and Eq. (14).

$$\alpha = \log_2 \left(\sum_i 2^{-\beta g(A_i)} \right) \tag{13}$$

$$f(\beta) = \sum_i (g(A_i) - G) 2^{-\beta(g(A_i) - G)} \tag{14}$$

The value of $f(\beta) = 0$, since it maximizes the L . On determining the value of α and β , the value of Entropy S can be computed by using the shortcut formula as shown in Eq. (15).

$$S = \alpha + \beta G \tag{15}$$

Thus, by solving Eq. (14), the value of the variable β can be obtained. And then by substituting the value β in Eq. (13), the value of α can be obtained. Once the value of α and β are known, they can be substituted in the expanded constraint given in Eq. (12) for various cluster groups i . Accordingly, the probability of an attribute for each class label $p(A_i)$ is identified. The process is repeated for all the significant attributes selected through the feature selection phase.

4.3. Class Probability based Classification

Consecutively, to find the class probability, the probabilities $p(A_i)$ of all attributes for each class label i for the given test sample is averaged. Finally, the test sample t can be labelled with the class having maximum average class probability as in Eq. (16).

$$t_i = \max_i \left(\frac{\sum_{j=1}^m p(A_i^j)}{m} \right) \quad (16)$$

Here i represents the class label that varies from 1 to n and j represents the attribute index that varies from 1 to m .

The algorithm steps for the proposed class probability distribution based maximum entropy model for the classification of instances having fewer attributes are presented below in Algorithm 1.

Algorithm1: CPDME_Model

Input: A training set with m attributes, n instance, k classes, and test instances

Output: Class label prediction for test instances

Procedure CPDME(training_set, test_data)

Begin

//Preprocessing of Data

1. Preprocess the given input training set by performing data cleaning by processing missing records, and data transformation using discretization and normalization.

//Phase 1: Feature selection using ARFS

2. Calculate the probability of the instances in each class c and the database score having k distinct classes.
3. Compute the relevance score of the features having q discrete values
4. Sort the attributes based on the computed score and rank them accordingly.
5. Normalize the scores by evaluating rank weights using the rank sum method.
6. Select the attributes having scores greater than the given threshold.
7. For each attribute in the test instances

//Phase 2: Relative Distance Computation

- a. Group the training instances based on the class variable
- b. Compute average class relative distance as in Eq. (7)
- c. Evaluate the value of relative gain G as in Eq. (9)

//Phase 3: Class Probability Computation using the Lagrange model

- a. Evaluate the Lagrange multipliers α and β as in Eq. (13) and (14).
- b. Evaluate the Entropy constraints and compute class probabilities as in Eq. (12) for all classes.

//Classification of the test instance

8. For each class
 - a. Aggregate the class probabilities of all the attributes obtained in the previous phase and average the class probability as in Eq. (16)
 - b. Classify the instance with the class label having maximum probability

End Procedure

Here for each attribute, the sum of the probabilities of all the classes will always be 1. Similarly, the sum of class probabilities for each test instance will be 1. The overall workflow of the proposed CPDME model is presented in Fig. 2. This proposed classification model provides better results for the datasets having fewer attributes and instances.

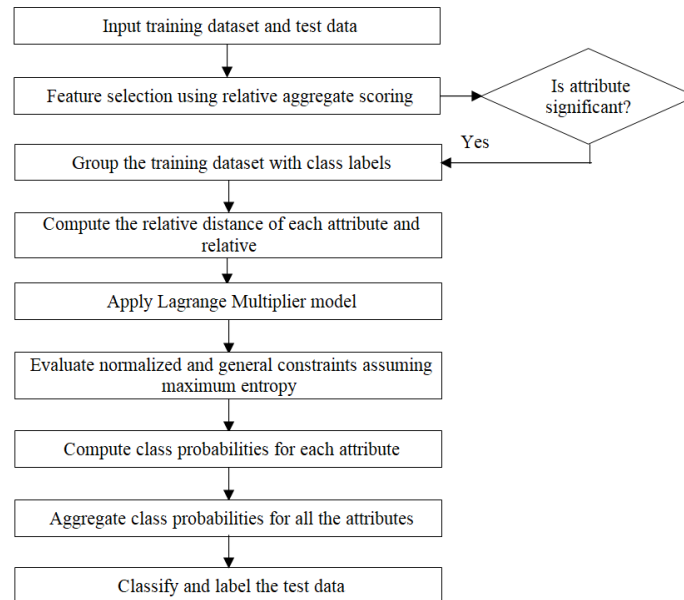


Fig. 2. Detailed Workflow of the Proposed Class Probability Distribution based Maximum Entropy Model

4.4. Case Study

The case study for the proposed probability distribution based maximum entropy model is discussed in this section. To explain the proposed model, the Iris dataset, donated by R. A. Fisher is employed. The dataset contains three classes of Iris in which each class contains 50 instances with 4 attributes. Theoretical experimentation has been performed by selecting 3 instances in each class with a total of 9 instances at random. As the dataset does not contain any missing values and as the number of attributes in the dataset is minimum, the selected instances do not undergo pre-processing step. The selected random samples from the Iris dataset that serves as training instances are presented in Table 2. Two random samples are picked from the Iris dataset, to serve as test instances and are shown in Table 3.

Initially, the training samples are grouped based on the class value. Then the probability of an attribute value of a test sample to be in each class is estimated. It is then combined with all the attribute values of the test sample to predict the classification. To proceed with an illustration of classifying the test sample $T1$, the sepal length attribute denoted as A_1 is evaluated. The centroid of an attribute in class 1 denoted as $C(A_1)$ is 5.27 by computing the mean of values of A_1 in class 1. Similarly, the centroid of an attribute for other classes such as class 2 and class 3 are 5.77 and 6.77 respectively.

Table 2. Random Training Samples from Iris Dataset

Test Sample ID	Sepal Length in cm	Sepal Width in cm	Petal Length in cm	Petal Width in cm	Class
S1	4.7	3.2	1.3	0.2	Iris-Setosa
S2	5.4	3.9	1.7	0.4	Iris-Setosa
S3	5.7	3.8	1.7	0.3	Iris-Setosa
S4	6.9	3.1	4.9	1.5	Iris-Versicolor
S5	4.9	2.4	3.3	1	Iris-Versicolor
S6	5.5	2.4	3.8	1.1	Iris-Versicolor
S7	5.8	2.7	5.1	1.9	Iris-Virginica
S8	7.7	2.8	6.7	2	Iris-Virginica
S9	6.8	3.2	5.9	2.3	Iris-Virginica

Table 3. Test Samples to be Classified

Test Sample ID	Sepal Length in cm	Sepal Width in cm	Petal Length in cm	Petal Width in cm
T1	4.7	3.2	1.3	0.2
T2	7.7	3.0	6.1	2.3

Then the value of $g(A_1)$, $g(A_2)$ and $g(A_3)$ are computed as in Eq. (7) which is the difference between the centroid of the attribute value of a class and a test sample T1 and it result in $g(A_1) = 0.23$, $g(A_2) = 0.27$, $g(A_3) = 1.27$. Based on the obtained values $g(A_1)$, $g(A_2)$ and $g(A_3)$, the expected value G is evaluated as in Eq. (9) and results in $G = 0.6889$.

Eventually, to find the value of β , Eq. (14) can be expressed as below.

$$-0.46 \times 2^{0.46\beta} - 0.42 \times 2^{0.42\beta} + 0.58 \times 2^{-0.58\beta} = 0$$

By applying Logarithm, the value of β is computed as 1.08962.

On substituting the value of β in Eq. (13) results in

$$\alpha = \log_2 (2^{-0.23\beta} + 2^{-0.27\beta} + 2^{-1.27\beta})$$

After evaluating the above equation, the value of α is evaluated as 1.0281

The obtained value of α and β can be substituted in the expanded version of Eq. (12).

$$p(A_1) = 2^{-1.0281} \times 2^{(-1.08962 \times 0.23)}$$

$$p(A_2) = 2^{-1.0281} \times 2^{(-1.08962 \times 0.27)}$$

$$p(A_3) = 2^{-1.0281} \times 2^{(-1.08962 \times 1.27)}$$

Upon solving the equations, we obtain $p(A_1) = 0.412177$, $p(A_2) = 0.399911$, $p(A_3) = 0.187912$.

Table 4. Class Probability of the Test Sample T₁

Class/ Attributes	Sepal Length	Sepal Width	Petal Length	Petal Width	Average Class Probability
Iris-Setosa	0.4122	0.0955	0.3316	0.3232	0.2906
Iris-Versicolor	0.3999	0.5581	0.3365	0.3625	0.4143
Iris-Virginica	0.1879	0.3465	0.3319	0.3143	0.2951
Attribute Probability	1.0000	1.0000	1.0000	1.0000	1.000

Table 5. Predicted Class for the Test Samples

Test Sample	Sepal Length in cm	Sepal Width in cm	Petal Length in cm	Petal Width in cm	Predicted Class
T1	4.7	3.2	1.3	0.2	2(Iris-Versicolor)
T2	7.7	3.0	6.1	2.3	3(Iris-Virginica)

The above steps can be continued for all the attributes in the given training dataset. The probability of all the attributes in each class is evaluated and the obtained results are presented in Table 4. It also specifies the overall probability of the test sample in each class. Here, the average class probability of the test sample of class 2 (Iris-Versicolor) is higher than the other classes. Hence, the test sample can be classified as Iris-Versicolor. It is also noted that the sum of attribute probability for all the classes will always 1. The predicted class labels for both test instances are presented in Table 5.

5. Experimental Analysis

The experimental and result analysis carried out for the proposed study is presented in this section. The experiments are performed on a system with intel core, i3-4005U CPU at 1.70Hz, 8 GB RAM, running 64bit OS of Windows 8.1 Pro windows edition. The experimental analysis is made for the proposed model with various datasets and the results are analysed in two sections 1) performance and statistical analysis with standard classifiers and 2) performance analysis with existing classification models.

5.1. Performance Analysis with Standard Classifiers

To evaluate the performance of the proposed model with standard classifiers, 17 datasets are employed. These datasets are available publically and are extracted from the UCI repository [64,] and KEEL [65] for classification. The number of attributes in the datasets varies widely from a minimum of 4 to a maximum of 60. Among the datasets used in the study, the datasets *Balance*, *Hayes Roth* and *Iris* have a minimum of 4 attributes and the dataset *Sonar* has a maximum number of attributes of 60. The number of classes in each dataset varies from 2 to 11. The datasets *German_credit*, *Ionosphere*, *Mushroom*, *Phoneme*, *Pima* and *Sonar* have the minimum number of class attribute values as 2 whereas *Vowel_context* has the maximum number of class attribute values as 11. Also, the number of instances in the datasets varies from 150 to 8124 with *Iris* as the smallest dataset with fewer instances and *Mushroom* as the largest dataset with a maximum number of instances. The number of attributes (bars graph) and classes (line graph) in each dataset used for the study is presented in Fig. 3 and the number of instances in each dataset is presented in Fig. 4.

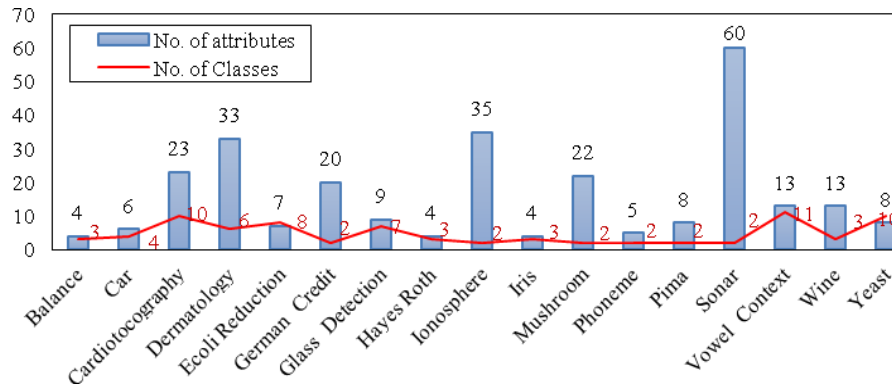


Fig. 3. Number of attributes and classes in different datasets

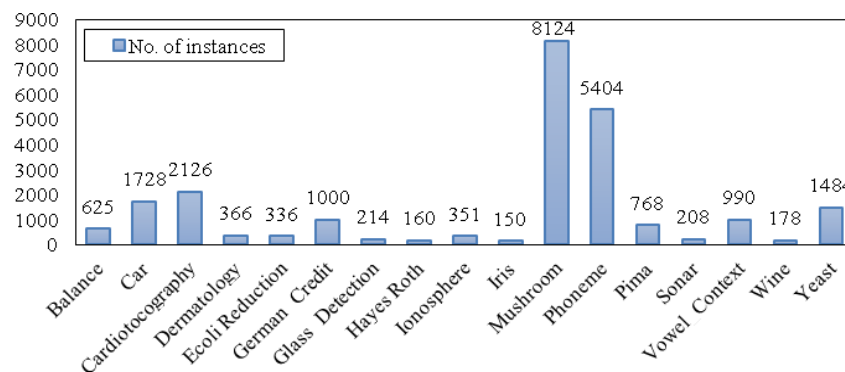


Fig. 4. Number of instances in different datasets

Diverse classifiers such as DL, NB, AB, LR, KNN, SRC, C4.5, SVM, ELM, RF, GBDT, DGC+ and AWPS are used for comparing the results of the proposed CPDME model. In general, 10-fold cross-validation is used for the evaluation of the proposed and existing models. For the classifiers that require parameter tuning, 80% of the instances in the datasets are used for training with 10% of the instances in the datasets being used as testing instances and the remaining 10% of the instances for tuning the parameters. While in the case of classifiers that do not require parameter tuning, 80% of the instances in the datasets are used for training and the remaining 20% of the instances in the datasets are used as a testing set [38]. Also, before applying classification, the significant attributes are selected utilizing the attribute rank based feature selection.

Accuracy Comparison: Table 6 shows the accuracy obtained with different classifiers for various datasets used for the analysis. The underlined values indicate the highest accuracy obtained for each dataset. From the results obtained, it is evident that the proposed model offers a better accuracy rate for 7 datasets such as *Car*, *Ecoli*

Reduction, Glass_Detection, Hayes Roth, Iris, Phoneme and Vowel_Context out of 17 datasets used for the evaluation. The average accuracy of CPDME with all 17 datasets is 89.9%. Out of 13 classifiers compared, the classifiers such as AWPS, RF and GBDT have the next higher accuracies at 88.2%, 87.9% and 87.3% respectively. Though the proposed model seems to be effective only with 7 datasets, it acquires the top position in average classification accuracy with an average rank of 3.47 and the classifiers GBDT, AWPS and RF, acquire the next three positions with ranks of 4.18, 4.71 and 4.88 respectively.

The statistical analysis for the obtained accuracy for the classification process is made using ANOVA with the null hypothesis that there is no significant difference in the accuracy of the classification algorithms. The statistical model is generated using F-distribution for which the obtained F value is 8.46 and the critical value is 1.76. The computed critical difference is 6.69 and the results are significant at a 5% significance level. Since the F value is greater than F critical value, the null hypothesis can be rejected and thus the alternate hypothesis is accepted indicating that there is a difference in the accuracy of the classification algorithms used for comparison.

Table 6. Accuracy results for different datasets

Dataset	CPDME	AWPS	DGC+	GBDT	RF	ELM	SVM	C4.5	SRC	KNN	LR	AB	NB	DL
Balance	0.987	0.904	0.899	0.968	0.952	0.952	0.921	0.857	<u>1.000</u>	0.952	0.937	0.809	0.968	0.460
Car	<u>1.000</u>	0.995	0.952	<u>1.000</u>	0.971	0.948	0.919	0.954	0.861	0.856	0.676	0.671	0.786	0.671
Cardiotocography	0.892	0.995	<u>0.999</u>	0.911	0.897	0.747	0.855	0.864	0.737	0.718	0.869	0.390	0.714	0.019
Dermatology	0.963	0.979	0.975	0.973	0.946	0.946	<u>1.000</u>	0.946	0.973	0.919	0.973	0.541	0.946	0.324
Ecoli Reduction	<u>0.892</u>	0.829	0.823	0.879	0.818	0.879	0.849	0.849	0.758	0.818	0.788	0.667	0.727	0.364
German_Credit	0.735	0.752	0.732	<u>0.760</u>	0.740	0.710	0.720	0.740	0.690	0.720	0.720	0.710	<u>0.760</u>	0.740
Glass_Detection	<u>0.857</u>	0.758	0.704	0.762	0.810	<u>0.905</u>	0.810	0.429	0.667	0.762	0.714	0.429	0.381	0.429
Hayes Roth	<u>0.872</u>	0.854	0.840	0.786	0.786	0.786	0.786	0.786	0.643	0.500	0.643	0.214	0.786	0.571
Ionosphere	0.912	<u>0.945</u>	0.931	0.917	0.917	0.889	0.806	0.944	0.944	0.889	0.889	0.917	0.806	0.722
Iris	<u>0.975</u>	0.972	0.953	0.947	0.953	0.922	0.960	0.867	0.967	0.967	0.953	0.947	0.867	0.867
Mushroom	0.987	0.999	0.995	<u>1.000</u>	0.995	0.978	<u>1.000</u>	<u>1.000</u>	<u>1.000</u>	0.998	0.987	0.967	0.957	0.967
Phoneme	<u>0.904</u>	0.878	0.871	0.867	0.895	0.880	0.775	0.847	0.899	0.893	0.745	0.771	0.734	0.285
Pima	0.827	0.737	0.745	0.701	0.805	0.662	0.650	0.766	0.597	0.610	0.805	<u>0.831</u>	0.753	0.597
Sonar	0.852	0.835	0.848	0.905	<u>0.952</u>	0.619	0.905	0.762	0.857	0.714	0.667	0.857	0.762	0.667
Vowel_Context	<u>0.999</u>	0.985	0.982	0.849	0.939	0.990	0.970	0.788	0.980	0.950	0.697	0.162	0.636	0.111
Wine	0.982	0.972	0.973	<u>1.000</u>	0.944	0.722	0.944	1.000	0.944	0.833	0.889	0.889	0.944	0.278
Yeast	0.645	0.598	0.593	0.622	0.622	<u>0.649</u>	0.628	0.514	0.574	0.547	0.622	0.412	0.595	0.331
Avg. Accuracy	<i>0.899</i>	<i>0.882</i>	<i>0.871</i>	<i>0.873</i>	<i>0.879</i>	<i>0.834</i>	<i>0.853</i>	<i>0.818</i>	<i>0.829</i>	<i>0.803</i>	<i>0.798</i>	<i>0.658</i>	<i>0.772</i>	<i>0.494</i>
Avg. Rank	<i>3.47</i>	<i>4.71</i>	<i>6.06</i>	<i>4.18</i>	<i>4.88</i>	<i>7.29</i>	<i>6.00</i>	<i>6.82</i>	<i>6.76</i>	<i>8.71</i>	<i>8.41</i>	<i>10.59</i>	<i>9.24</i>	<i>12.65</i>

AUC Comparison: Table 7 shows the AUC values obtained for the 13 classifiers with 17 datasets in which the underlined values represent the maximum AUC obtained for each dataset. From the reported result, it is evident that the proposed model offers a better AUC value for 7 datasets such as *Car*, *Ecoli Reduction*, *Glass_Detection*, *Hayes Roth*, *Iris*, *Phoneme* and *Yeast* with an average AUC value of 0.946. Among 13 standard classifiers, the next highest average AUC values are acquired by DGC+, AWPS, and RF as 0.932, 0.930 and 0.883 respectively. Despite acquiring minimum AUC values with 10 datasets, the proposed model holds the first position with an average rank of 2.82 which is better than other classifiers such as DGC+ and AWPS, with average ranks of 3.47 and 3.59.

The statistical analysis for the obtained AUC is carried out using the ANOVA test with the null hypothesis stating that there is a difference in AUC values of the classification algorithms. The statistical model is constructed using F-distribution in which the obtained F value and critical value are 10.326 and 1.76 (10.326 > 1.76). The computed critical difference is 8.56 and the results are significant at a 5% significance level. Thus, the null hypothesis is rejected and the alternate hypothesis is accepted which indicates that there is a difference in AUC values of the classification algorithms under comparison.

Table 7. AUC Comparison among different classifiers

Dataset	CPDME	AWPS	DGC+	GBDT	RF	ELM	SVM	C4.5	SRC	KNN	LR	AB	NB	DL
Balance	0.992	0.862	0.875	0.833	0.833	0.833	0.867	0.781	<u>1.000</u>	0.984	0.956	0.724	0.833	0.500
Car	<u>1.000</u>	0.994	0.998	<u>1.000</u>	0.951	0.930	0.897	0.929	0.878	0.836	0.575	0.500	0.867	0.509
Cardiotocography	0.895	<u>0.999</u>	0.996	0.882	0.874	0.823	0.862	0.844	0.834	0.824	0.922	0.736	0.822	0.500
Dermatology	0.983	0.989	0.991	0.980	0.980	0.960	<u>1.000</u>	0.921	0.980	0.980	0.980	0.752	0.960	0.684
Ecoli	<u>0.982</u>	0.978	0.957	0.875	0.903	0.906	0.895	0.908	0.892	0.892	0.906	0.763	0.888	0.500
German_Credit	0.832	0.751	0.743	0.699	<u>0.939</u>	0.654	0.645	0.685	0.623	0.694	0.654	0.613	0.741	0.500
Glass_Detection	<u>0.992</u>	0.865	0.854	0.719	0.748	0.986	0.790	0.817	0.875	0.963	0.727	0.806	0.815	0.500
Hayes Roth	<u>0.965</u>	0.936	0.948	0.955	0.927	0.952	0.949	0.902	0.936	0.834	0.895	0.914	0.952	0.904
Ionosphere	0.902	<u>0.950</u>	0.931	0.889	0.644	0.844	0.835	0.909	0.909	0.864	0.869	0.889	0.809	0.622
Iris	<u>0.999</u>	<u>0.999</u>	0.994	0.874	0.986	0.987	0.989	0.887	0.878	0.897	0.957	0.960	0.979	0.878
Mushroom	0.992	0.999	0.995	<u>1.000</u>	0.994	0.992	0.992	0.935	0.994	0.927	0.994	0.972	0.991	0.921
Phoneme	<u>0.898</u>	0.875	0.866	0.844	0.859	0.848	0.651	0.799	0.863	0.856	0.639	0.672	0.707	0.500
Pima	0.857	0.788	<u>0.866</u>	0.677	0.747	0.628	0.617	0.731	0.563	0.579	0.774	0.806	0.725	0.500
Sonar	0.799	0.886	0.891	<u>0.896</u>	0.885	0.882	0.799	0.721	0.879	0.882	0.789	0.633	0.856	0.692
Vowel	0.998	0.985	0.982	0.914	0.935	0.999	0.998	0.914	0.997	<u>1.000</u>	0.853	0.713	0.791	0.576
Wine	0.998	0.965	0.973	<u>1.000</u>	0.967	0.719	0.900	<u>1.000</u>	0.967	0.790	0.873	0.917	0.967	0.500
Yeast	<u>0.999</u>	0.996	0.991	0.847	0.837	0.829	0.825	0.825	0.799	0.823	0.836	0.675	0.838	0.500
Avg. AUC	<i>0.946</i>	<i>0.930</i>	<i>0.932</i>	<i>0.876</i>	<i>0.883</i>	<i>0.869</i>	<i>0.854</i>	<i>0.853</i>	<i>0.875</i>	<i>0.860</i>	<i>0.835</i>	<i>0.767</i>	<i>0.855</i>	<i>0.605</i>
Avg. Rank	2.82	3.59	3.47	6.18	6.41	7.12	7.76	8.41	7.12	8.29	8.35	10.94	8.47	13.59

Execution Time Comparison: The running time to train and test the proposed CPDME model are analysed and compared with 13 different classifiers with 17 different datasets. The results of the execution time for the proposed and the existing models are presented in Table 8. The underlined values in the table represent the minimum execution time for the dataset. From the results obtained, it is clear that the proposed model has a minimum running time for the datasets such as *Balance*, *Hayes Roth*, *Iris*, *Phoneme* and *Yeast* which are less than 2 ms. In general, the average running time of the proposed model is 4.64 ms and acquires 7th rank whereas the execution times of the top 6 classifiers are 28 ms (NB), 0.30 ms (KNN), 0.34 ms (C4.5), 0.34 ms (AB), 1.03 ms (LR), 3.77 ms (SVM), 4.14 ms (AWPS). Optimistically, still, the proposed model has a minimum execution time than the other 8 classifiers used in the comparison.

Table 8. Execution Time (in ms) Comparison among different classifiers

Dataset	CPDME	AWPS	DGC+	GBDT	RF	ELM	SVM	C4.5	SRC	KNN	LR	AB	NB	DL
Balance	0.121	4.45	5.71	15.47	7.481	4.11	0.723	0.022	4.88	0.022	0.04	0.02	<u>0.016</u>	3.09
Car	0.236	5.06	11.89	34.71	12.73	38.93	0.589	0.044	83.32	<u>0.028</u>	0.15	0.06	0.034	2.964
Cardiotocography	10.1	6.13	14.64	190.8	109.61	64.77	<u>0.285</u>	0.871	307.3	0.631	11.5	0.419	0.621	1.14
Dermatology	9.98	2.12	9.89	20.7	48.25	1.54	<u>0.125</u>	0.879	1.25	0.325	0.879	0.623	0.741	2.85
Ecoli	0.396	2.09	5.72	16.49	10.92	1.21	0.425	0.014	1.09	0.022	0.04	<u>0.01</u>	0.012	1.31
German_Credit	2.36	4.11	13.18	13.29	89.69	11.92	<u>0.171</u>	0.952	20.66	0.369	0.234	0.412	0.357	4.265
Glass_Detection	0.936	2.02	5.98	15.61	15.39	0.52	0.323	0.014	0.35	0.034	0.04	<u>0.01</u>	0.013	1.875
Hayes_Roth	0.109	1.96	5.07	11.58	10.36	9.55	7.51	0.187	2.35	0.245	<u>0.09</u>	<u>0.09</u>	0.131	5.87
Ionosphere	2.91	3.94	10.56	7.22	60.14	1.21	0.721	0.532	0.99	0.567	<u>0.236</u>	0.413	0.561	3.89
Iris	1.22	1.86	5.11	8.32	10.48	5.32	7.99	0.057	1.89	0.057	0.08	0.234	<u>0.015</u>	2.457
Mushroom	3.978	13.84	26.01	16.32	31.26	19.49	28.78	<u>0.723</u>	45.5	0.811	1.18	0.987	0.725	30.49
Phoneme	0.203	9.43	23.01	26.64	34.5	388.45	0.331	0.187	3530.1	<u>0.074</u>	0.19	0.22	0.091	2.753
Pima	0.121	2.44	10.24	7.19	16.57	7.58	0.133	0.028	11.88	<u>0.022</u>	0.03	0.241	0.038	2.45
Sonar	12.31	2.35	12.79	15.36	18.18	11.23	15.54	0.977	17.28	0.932	1.23	0.912	<u>0.812</u>	12.36
Vowel	0.102	1.93	13.08	112.3	43.69	11.13	0.245	0.083	20.31	<u>0.024</u>	0.6	0.321	0.125	2.68
Wine	0.222	1.85	6.37	8.57	19.79	0.57	0.345	<u>0.012</u>	0.59	0.023	0.03	0.369	0.235	1.235
Yeast	1.832	4.87	14.24	93.85	25.96	29.36	<u>0.184</u>	0.259	112.8	0.949	0.89	0.412	0.196	1.857
Avg. Exec. Time.	2.77	4.14	11.38	36.14	33.24	35.70	3.79	0.34	244.86	0.30	1.03	0.34	0.28	4.91
Avg. Rank	6.47	8.71	10.88	12.41	13.06	9.94	6.29	3.18	10.94	3.06	4.35	3.65	2.71	9.12

From the result analysis, it is clear that the proposed CPDME model outperforms other existing models for the various datasets such as *Car*, *Ecoli Reduction*, *Glass_Detection*, *Hayes Roth*, *Iris*, *Phoneme* and *Vowel_Context* in which most of the datasets have fewer attributes. This shows that the proposed model is effective with the datasets having the minimum number of attributes and offers better performance. The obtained ranks for accuracy, AuC and execution time of the proposed CPDME and the other standard models under comparison are presented as a graph and shown in Fig. 5.

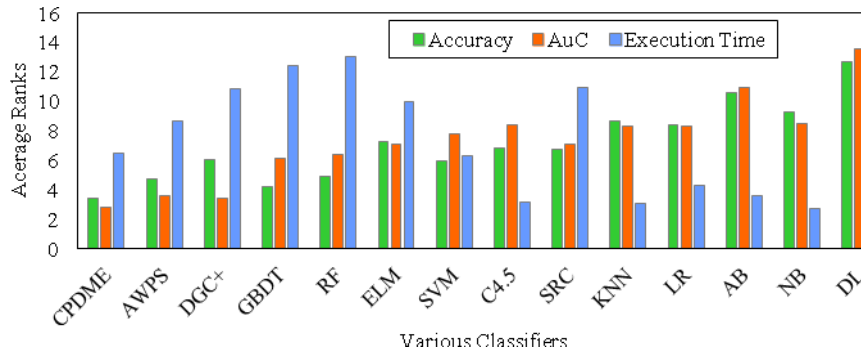


Fig. 5. Ranks obtained for various metrics

5.2. Performance Comparison with Existing Models

An analysis has been carried out for the proposed model by evaluating the classification accuracy of various models using 36 datasets. The datasets used for the study are publically available and are downloaded from the UCI repository [64] and KEEL [65]. The results of the proposed model are compared with the various existing model such as AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB. The values in Table 9 are the result of performing the average on the accuracies obtained from 10 individual runs under stratified ten-fold cross-validation as in [49]. The various algorithms are applied to the same training and test set. The values represented in the boldface at each row indicate the highest accuracy value corresponding to the dataset. Moreover, the underlined values indicate that the proposed model outperforms other models under comparison with paired two-tailed t-tests at the p=0.05 significance level. The symbol * denotes the significant performance degradation over its competitors. The last two rows indicate the average accuracy and Win/Tie/Lose (W/T/L) of each classifier. Each W/T/L specifies that the proposed model wins W datasets, ties on T datasets and loses on L datasets to the respective competitor model [49].

From the analysis, it is clear that the proposed CPDME classifier has the highest accuracy for 15 datasets which is better than the competitors. The models IWHNB, AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB have the highest classification accuracy for 8, 3, 7, 3, 2 and 2 datasets respectively. In terms of the average classification accuracy, the proposed CPDME classifier is 86.93%. On the other hand, the models such as IWHNB, AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB have the average accuracy of 86.37%, 84.94%, 85.52%, 84.41%, 84.21% and 83.86% respectively. The performance of the proposed model is as similar as the IWHNB classifier. Also, the CPDME model has 6 wins, 26 ties and 4 losses which is better than the other models IWHNB (6 wins, 26 ties, 4 losses) AIWNB^E (8 wins, 25 ties, 3 losses), AIWNB^L (6 wins, 27 ties, 3 losses), CAWNB (13 wins, 21 ties, 2 loss), AVFWNB (14 wins, 20 ties, 2 loss) and NB (17 wins, 17 ties, 2 loss).

Table 9. Comparison of Classification Accuracy

Dataset	CPDME	IWHNB	AIWNB ^E	AIWNB ^L	CAWNB	AVFWNB	NB
Anneal	98.94±1.12	<u>98.31±1.29</u>	98.94±1.05	98.90±1.10	98.5±1.29	98.62±1.15	<u>96.36±1.97</u>
Anneal.ORIG	95.12±2.30	<u>94.65±2.24</u>	95.06±2.23	95.06±2.23	94.6±2.48	<u>93.32±2.65</u>	<u>92.71±2.7</u>
Audiology	85.32±6.91	<u>78.17±7.15</u>	83.93±7.00	84.81±6.83	<u>74.22±6.36</u>	<u>78.58±8.44</u>	<u>75.74±6.58</u>
Autos	88.93±9.01	85.56±7.93	<u>78.04±9.02</u>	<u>79.80±8.63</u>	<u>77.95±8.95</u>	<u>77.27±9.43</u>	<u>77.02±9.69</u>
Balance-scale	74.21±4.51	<u>69.05±3.74</u>	73.75±4.22	73.52±4.39	73.76±4.15	71.1±4.3	71.08±4.29
Breast-cancer	70.01±7.21	70.47±6.29	71.9±7.55	71.52±7.23	72.46±7.25*	71.41±7.98	72.32±7.91
Breast-w	97.11±1.63	96.30±1.94	97.17±1.68	97.15±1.77	97.14±1.81	97.48±1.68*	97.25±1.79
Colic	82.36±5.43	81.20±6.00	83.45±5.45*	83.4±5.44	83.34±5.62	81.47±5.86	81.2±5.8
colic.ORIG	73.36±6.3	74.23±6.52	73.87±6.4	74.38±6.7*	73.7±6.46	72.91±6.34	73.43±6.27
credit-a	86.12±3.84	85.23±3.82	87.03±3.83*	86.93±3.85	86.99±3.81	86.23±3.85	86.17±3.94
credit-g	75.83±3.7	75.85±3.69	75.81±3.6	75.86±3.67	75.7±3.53	75.38±3.9	75.4±4.01
Diabetes	79.36±4.7	<u>76.75±4.20</u>	<u>77.87±4.86</u>	78.32±4.67	78.01±4.89	<u>77.89±4.66</u>	<u>77.88±4.65</u>
Glass	78.21±8.3	77.70±8.98	<u>74.02±8.41</u>	<u>74.9±8.25</u>	<u>73.37±8.38</u>	76.25±8.07	<u>74.2±8.11</u>
heart-c	83.26±6.41	81.52±7.12	82.71±6.61	82.81±6.61	82.94±6.57	83.04±6.68	83.73±6.46*
heart-h	84.87±5.91	84.56±6.05	84.29±5.85	84.26±5.89	83.82±6.16	84.9±5.68*	84.43±5.88
heart-statlog	83.31±6.28	82.33±6.59	83.22±6.61	83.19±6.71	83.44±6.69	83.78±6.29	83.74±6.25
Hepatitis	85.91±9.24	87.38±8.43*	85.75±8.97	86±9.07	85.95±9.25	85.38±9	85.05±9.45
Hypothyroid	98.23±0.59	99.32±0.40*	99.07±0.48	99.05±0.5	98.56±0.56	98.98±0.48	98.74±0.57
Ionosphere	92.25±3.92	93.96±3.65	92.4±4.13	92.68±3.76	<u>91.82±4.34</u>	<u>91.94±4.09</u>	<u>91.37±4.55</u>
Iris	96.23±5.8	93.27±5.72	94.4±5.5	94.4±5.5	94.4±5.5	94.4±5.5	94.33±5.56
kr-vs-kp	93.85±1.41	92.70±1.37	93.73±1.28	94.06±1.27*	93.58±1.32	<u>88.18±1.86</u>	<u>87.81±1.9</u>
Labor	96.33±10.13	95.90±9.21	94.33±9.3	93.8±10.17	92.1±10.94	94.33±10.13	93.83±10.41
Letter	85.6±0.85	90.17±0.62*	<u>75.56±0.89</u>	<u>79.6±0.85</u>	<u>75.22±0.83</u>	<u>75.07±0.84</u>	<u>74.67±0.86</u>
Lymphography	86.12±7.83	85.89±8.02	84.68±7.99	85.08±7.72	84.81±8.13	85.49±7.83	85.7±7.95
Mushroom	99.9±0.31	99.96±0.06	99.53±0.23	99.71±0.2	<u>99.19±0.32</u>	<u>99.12±0.31</u>	<u>98.03±0.49</u>
primary-tumor	48.21±5.37	46.14±6.17	47.76±5.25	47.76±5.21	47.2±5.27	45.85±6.53	47.11±5.65
Segment	95.18±1.41	96.87±1.07	94.16±1.38	95.32±1.32	<u>93.47±1.46</u>	<u>93.69±1.41</u>	<u>92.91±1.56</u>
Sick	96.23±0.89	97.52±0.76	97.33±0.85*	97.36±0.83*	97.36±0.84*	97.02±0.86	97.07±0.84
Sonar	83.89±8.57	84.63±7.72	<u>82.23±8.65</u>	<u>82.28±8.57</u>	<u>82.56±8.25</u>	84.49±7.79	84.96±7.57*
Soybean	94.62±2.23	94.61±2.18	94.74±2.19	94.82±2.24	<u>93.66±2.73</u>	94.52±2.36	<u>93.53±2.79</u>
Splice	96.32±1.11	96.24±1.00	96.21±0.99	96.55±1.01	96.19±0.99	<u>95.61±1.11</u>	<u>95.58±1.12</u>
Vehicle	72.58±3.58	73.70±3.41*	<u>63.59±3.92</u>	<u>67.57±3.27</u>	<u>62.91±3.88</u>	<u>63.36±3.87</u>	<u>62.64±3.84</u>
Vote	94.74±3.21	94.39±3.21	92.18±3.76	93.68±3.52	<u>92.11±3.74</u>	<u>90.25±3.95</u>	<u>90.3±3.89</u>
Vowel	89.95±4.12	90.32±2.71	<u>69.98±4.11</u>	74.48±3.93	<u>68.84±4.3</u>	<u>67.46±4.62</u>	<u>66±4.58</u>
waveform-5000	88.61±1.52	<u>86.24±1.45</u>	<u>82.98±1.37</u>	<u>83.51±1.38</u>	<u>83.11±1.38</u>	<u>80.65±1.46</u>	<u>80.76±1.49</u>
Zoo	98.71±5.2	98.33±3.72	96.05±5.6	96.05±5.6	95.96±5.61	96.05±5.6	95.75±5.68
Average	86.93	86.37	84.94	85.52	84.41	84.21	83.86
W/T/L	-	6/26/4	8/25/3	6/27/3	13/21/2	14/20/2	17/17/2

6. Result Analysis of the Proposed Model

6.1. Complexity Analysis

The computational complexity of the proposed CPDME model using Big-O notation is $O(nmk)$ where m is the attribute count in the dataset, k is the count of target class values and n is the count of the instances at each class in the dataset. The computation complexity of the other existing algorithms AWPS and DGC+ having higher ranks in average classification accuracy or AUC values are $O(nm)$ and $O(mn^2)$, where n is the count of instances and m is the count of attributes in the dataset. Similarly, the computational complexity of RF is $O(tmn(\log n))$ in which the complexity depends on the number of trees (t) to be constructed. Though the computational complexity of the proposed classifier is slightly higher than the other models, the accuracy of CPDME is better than many of the existing algorithms.

6.2. Statistical Analysis of Results

A statistical analysis has been carried out to assess the performance of the proposed model for which the results presented in Table 6 are used. The highest accuracy obtained by the proposed model among the 17 datasets is statistically distributed based on various characteristics such as attribute count, instance count and the number of classes. The statistical distribution is provided in Table 10. The column *count* indicates the number of datasets won by the proposed model with the highest accuracy and the percentage indicates the values in percentage. Thus, out of 12 datasets having an attribute count of less than 20, the proposed model acquires the highest accuracy for 7 datasets indicating a success rate of 58.33%.

Table 10. Statistical Analysis of Data Characteristics for Proposed CPDME

Characteristics	No. of datasets	CPDME		Other Models
		Count	Percentage	
Attributes \leq 20	12	7	58.33	41.67
Attributes $>$ 20	5	0	0	100
Instances \leq 500	8	4	50	50
Instances $>$ 500	9	3	33.33	66.67
Classes \leq 5	11	4	36.36	63.64
Classes $>$ 5	6	2	33.33	66.67

Similarly, out of 5 datasets having an attribute count greater than 20, the proposed model has 0 success signifying that the standard models under comparison achieve the highest accuracy (100%). While considering the instances less than or equal to 500, the proposed CPDME model has a success rate of about 50% with the highest accuracy for 4 out of 8 datasets. On the other hand, for the datasets having an instance count greater than 500, the proposed model has less success rate of about 33.33% by achieving the

highest accuracy for 3 out of 9 datasets. With a distribution based on the distinct class count, the results are even for the proposed and existing models. Thus, the model offers better results than many other competitors with the datasets having fewer attributes or instances.

Furthermore, the highest accuracy obtained by the proposed model and existing models among the 36 datasets presented in Table 9 is statistically distributed based on various characteristics such as attribute count, instance count. Table 11 provides the statistical distribution of highest accuracies on number of datasets for the models CPDME, IWHNB, AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB.

Table 11. Comparison of the percentage of datasets having higher accuracy

Characteristics	CPDME	IWHNB	AIWNB ^E	AIWNB ^L	CAWNB	AVFWNB	NB
Attributes ≤ 25	50.00	20.83	8.33	8.33	8.33	8.33	4.17
Attributes > 25	25.00	25.00	8.33	41.67	8.33	0.00	8.33
Instances ≤ 500	50.00	11.11	5.56	5.56	11.11	5.56	11.11
Instances > 500 and ≤ 1000	36.36	27.27	18.18	18.18	0.00	9.09	0.00
Instances > 1000	28.57	42.87	0.00	57.14	14.29	0.00	0.00

From the analysis, the proposed CPDME has a higher success rate of about 50% with the highest accuracy for the datasets having attributes less than or equal to 25, whereas the combined success rate of other 6 models is 50%. For the datasets having a number of attributes greater than 25, the proposed CPDME model has the lowest success rate of about 25% than other models (75%). Correspondingly, the proposed CPDME model has a higher lowest success rate of 50% and 36% with the highest accuracy for the datasets having a number of instances less than or equal to 500 and between 500 and 1000 respectively. With the datasets having instance count greater than 1000, the proposed model has a less lowest success rate of about 28.57% of the highest accuracy. The increase in the number of instances or attributes gradually decreases the performance of the proposed model. Thus, from the results of the statistical analysis, it clear that the proposed model offers better results with the minimum number of attributes and instances.

7. Conclusion

This paper suggests the class probability distribution based on maximum entropy classification to classify the instances of the datasets having fewer attributes and instances. In the first phase, the important features are identified using attribute rank based feature selection. For each selected attribute, the average class relative distance is evaluated for the training samples. Then the relative gain of the attributes is computed from the test sample and the relative distance of each class. The Lagrange multipliers are applied and evaluated and the class probabilities concerning the attributes are computed by maximizing the entropy. Finally, the class label is predicted by aggregating the class probabilities of all the attributes. Experimental analysis has been performed with two sets of experiments using 17 and 36 datasets. The proposed model offers better average

accuracy of about 89.9% and 86.93% for the two experiments respectively which is better than many of the other existing and standard models. The statistical result analysis shows that the proposed model offers better results with improved accuracy for more than 50% of the datasets having fewer attributes and instances than other competitors. The future work focuses on the imbalanced class distribution along with the sparse distribution of attributes and instances. Though the proposed model has better accuracy, it suffers from time overhead which is below the top 5 positions in comparison with other models. Thus, future work concentrates on increasing the classification speed of the proposed model.

References

1. Olson, D.L., Shi, Y.: Introduction to Business Data Mining. McGraw-Hill Education, New York. (2007)
2. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques, A Volume in the Morgan Kaufmann Series in Data Management Systems, Third Edition. Elsevier. (2011)
3. Urso, A., Fiannaca, A., La Rosa, M., Ravì, V., Rizzo, R.: Data Mining: Classification and Prediction. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, Vol. 1, No. 3, 384-402. (2018)
4. Francis, B.K., Babu, S.S.: Predicting Academic Performance of Students Using a Hybrid Data Mining Approach. Journal of Medical Systems, Vol. 43, No. 6, 1-15. (2019)
5. Shen, F., Zhao, X., Li, Z., Li, K., Meng, Z.: A Novel Ensemble Classification Model based on Neural Networks and a Classifier Optimisation Technique for Imbalanced Credit Risk Evaluation. Physica A: Statistical Mechanics and its Applications, Vol. 526, 121073. (2019)
6. Fatima, M., Pasha, M.: Survey of Machine Learning Algorithms for Disease Diagnostic. Journal of Intelligent Learning Systems and Applications, Vol. 9, No. 1, 1-16. (2017)
7. Van Klompenburg, T., Kassahun, A., Catal, C.: Crop Yield Prediction using Machine Learning: A Systematic Literature Review. Computers and Electronics in Agriculture, Vol. 177, 105709. (2020)
8. Dingli, A., Marmara, V., Fournier, N.S.: Comparison of Deep Learning Algorithms to Predict Customer Churn within a Local Retail Industry. International Journal of Machine Learning and Computing, Vol. 7, No. 5, 128-132. (2017)
9. Sharaff, A., Gupta, H.: Extra-tree classifier with metaheuristics approach for email classification. In Advances in Computer Communication and Computational Sciences, Springer, Singapore. pp. 189-197. (2019)
10. Choubin, B., Zehtabian, G., Azareh, A., Rafiei-Sardooi, E., Sajedi-Hosseini, F., Kişi, Ö.: Precipitation Forecasting Using Classification and Regression Trees

- (CART) Model: A Comparative Study of Different Approaches. *Environmental Earth Sciences*, Vol. 77, No. 8, 1-13. (2018)
11. Nayak, S., Panigrahi, C.R., Pati, B., Nanda, S., Hsieh, M.Y.: Comparative analysis of HAR datasets using classification algorithms. *Computer Science and Information Systems*, Vol. 19, No. 1, 47-63 (2022).
 12. Rogić, S., Kaščelan, L.: Class balancing in customer segments classification using support vector machine rule extraction and ensemble learning. *Computer Science and Information Systems*, Vol. 18, No. 3, 893-925, (2021).
 13. Sathya Bama, S., Irfan Ahmed, M.S., Saravanan, A.: Network Intrusion Detection using Clustering: A Data Mining Approach. *International Journal of Computer Applications*, Vol. 30, No. 4, 14-17. (2011)
 14. Sahu, K., Srivastava, R.K.: Predicting software bugs of newly and large datasets through a unified neuro-fuzzy approach: Reliability perspective. *Advances in Mathematics: Scientific Journal*, Vol. 10, No. 1, 543-555 (2021).
 15. Liu, H., Motoda, H., Setiono, R., Zhao, Z.: Feature Selection: An Ever Evolving Frontier in Data Mining. In *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, Proceedings of Machine Learning Research*, Vol. 10, 4-13. (2010)
 16. Kumar, P., Ambekar, S., Kumar, M., Roy, S.: Analytical Statistics Techniques of Classification and Regression in Machine Learning. In *Data Mining-Methods, Applications and Systems*. IntechOpen. (2020)
 17. Altman, N.S.: An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, Vol. 46, No. 3, 175-185. (1992)
 18. Shaban, W.M., Rabie, A.H., Saleh, A.I., Abo-Elsoud, M.A.: A New COVID-19 Patients Detection Strategy (CPDS) Based on Hybrid Feature Selection and Enhanced KNN Classifier. *Knowledge-Based Systems*, Vol. 205, 106270. (2020)
 19. Leung, K.M.: Naive Bayesian Classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 123-156. (2007)
 20. Berrar, D.: Bayes' Theorem and Naive Bayes Classifier. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*; Elsevier Science Publisher: Amsterdam, The Netherlands, 403-412. (2018)
 21. Cox, D.R.: The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. 20, No. 2, 215-232. (1958)
 22. Boonchuay, K., Sinapiromsaran, K., Lursinsap, C.: Decision Tree Induction based on Minority Entropy for the Class Imbalance Problem. *Pattern Analysis and Applications*, Vol. 20, No. 3, 769-782. (2017)
 23. Cherfi, A., Nouira, K., Ferchichi, A.: Very Fast C4. 5 Decision Tree Algorithm. *Applied Artificial Intelligence*, Vol. 32, No. 2, 119-137. (2018)
 24. Cortes, C., Vapnik, V.: Support Vector Networks. *Machine Learning*, Vol. 20, No. 3, 273-297. (1995)
 25. Suthaharan, S.: Support Vector Machine, Machine Learning Models and Algorithms for Big Data Classification, *Integrated Series in Information Systems*, Springer, Boston, MA., 207-235. (2016)

26. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 2, 210-227. (2008)
27. Smolyakov, V.: Ensemble Learning to Improve Machine Learning Results. *Stats and Bots*, Available: <https://blog.statsbot.co/ensemble-learning-d1dcd548e936>. (2017)
28. Freund, Y., Schapire, R. E.: Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, Vol. 96, 148–156. (1996)
29. Breiman, L.: Random Forests. *Machine Learning*, Vol. 45, No. 1, 5–32. (2011)
30. Wolpert, D.H.: Stacked Generalization. *Neural Networks*. Vol. 5, No. 2, 241–259. (1992)
31. Friedman, J. H.: Stochastic Gradient Boosting. *Computational Statistics & Data Analysis*, Vol. 38, No. 4, 367–378. (2002)
32. Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H.: State-of-the-Art in Artificial Neural Network Applications: A Survey. *Heliyon*, Vol. 4, No. 11. (2018)
33. Bengio, Y.: Learning Deep Architectures for AI. *Found. Trends Mach. Learn.*, Vol. 2, 1–127. (2009)
34. Sejnowski, T.J.: *The Deep Learning Revolution*. MIT Press. (2018)
35. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 42, No. 2, 513-529. (2011)
36. Çığsar, B., Ünal, D.: Comparison of Data Mining Classification Algorithms Determining the Default Risk. *Scientific Programming*, 1-8. (2019)
37. Katarya, R., Jain, S.: Comparison of different machine learning models for diabetes detection. In *International Conference on Advances and Developments in Electrical and Electronics Engineering (ICADEE)*, IEEE, 1-5. (2020)
38. Zhang, C., Liu, C., Zhang, X., Almpandis, G.: An Up-to-Date Comparison of State-of-the-Art Classification Algorithms. *Expert Systems with Applications*, Vol. 82, 128-150. (2017)
39. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L. C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, Vol. 247, 124–136. (2015)
40. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, Vol. 752, No. 1, 41-48. (1998)
41. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. *Machine learning*, Vol. 39, No. 2, 103-134. (2000)
42. Raina, R., Shen, Y., Mccallum, A., Ng, A.: Classification with hybrid generative/discriminative models. *Advances in neural information processing systems*, Vol. 16. (2003)

43. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In International Conference on Machine Learning 2001, 282-289. (2001)
44. Peng, L., Yang, B., Chen, Y., Abraham, A.: Data Gravitation based Classification. Information Sciences, Vol. 179, No. 6, 809-819. (2009)
45. Cano, A., Zafra, A., Ventura, S.: Weighted Data Gravitation Classification for Standard and Imbalanced Data. IEEE Transactions on Cybernetics, Vol. 43, No. 6, 1672-1687. (2013)
46. Sathya Bama, S.S., Saravanan, A.: Efficient Classification using Average Weighted Pattern Score with Attribute Rank based Feature Selection. International Journal of Intelligent Systems and Applications, Vol. 10, No. 7, 29. (2019)
47. Xu, W., Jiang, L., Yu, L.: An Attribute Value Frequency-based Instance Weighting Filter for Naive Bayes. Journal of Experimental & Theoretical Artificial Intelligence, Vol. 31, No. 2, 225-236. (2019)
48. Jiang, L., Zhang, L., Li, C., Wu, J.: A Correlation-based Feature Weighting Filter for Naive Bayes. IEEE Transactions on Knowledge and Data Engineering, Vol. 31, No. 2, 201-213. (2018)
49. Zhang, H., Jiang, L., Yu, L.: Attribute and Instance Weighted Naive Bayes. Pattern Recognition, Vol. 111, 107674. (2021)
50. Jiang, L., Wang, D., Cai, Z.: Discriminatively weighted naive Bayes and its application in text classification. International Journal on Artificial Intelligence Tools, Vol. 21, No. 01, p.1250007 (2012).
51. Zhang, H., Jiang, L., Li, C.: Collaboratively weighted naive Bayes. Knowledge and Information Systems, Vol. 63, No. 12, 3159-3182. (2021)
52. Yu, L., Gan, S., Chen, Y., Luo, D.: A Novel Hybrid Approach: Instance Weighted Hidden Naive Bayes. Mathematics, Vol. 9, No. 22, 2982. (2021)
53. Vajapeyam, S.: Understanding Shannon's Entropy Metric for Information. arXiv Preprint arXiv:1405.2061. (2014)
54. Glen, S., Maximum Entropy Principle: Definition, From StatisticsHowTo.com: Elementary Statistics for the rest of us! <https://www.statisticshowto.com/maximum-entropy-principle/>
55. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In workshop on machine learning for information filtering, Vol. 1, No. 1, 61-67. (1999)
56. Lazebnik, S., Schmid, C., Ponce, J.: A maximum entropy framework for part-based texture and object recognition. In IEEE International Conference on Computer Vision, Vol. 1, 832-838. (2005)
57. Feng, Z., Zhou, Y., Wu, L., Li, Z.: Audio classification based on maximum entropy model. International Conference on Multimedia and Expo, IEEE. Vol. 1, pp. I-745. (2003)
58. Alrashdi, I., Siddiqi, M.H., Alhwaiti, Y., Alruwaili, M., Azad, M.: Maximum entropy Markov model for human activity recognition using depth camera. IEEE Access, Vol. 9, 160635-160645 (2021).

59. Arunraj, G, Radha, B.: Feature Selection using Multiple Ranks with Majority Vote Based Relative Aggregate Scoring Model for Parkinson Dataset. International Conference on Data Science and Applications. (2021)
60. Sim, J., Lee, J.S., Kwon, O.: Missing Values and Optimal Selection of an Imputation Method and Classification Algorithm to Improve the Accuracy of Ubiquitous Computing Applications. *Mathematical Problems in Engineering*, pp. 1-14. (2015)
61. Yang, Y., Webb G.I., Wu X.: Discretization Methods. In: Editor, Maimon O., Rokach L. (eds) *Data Mining and Knowledge Discovery Handbook*. Springer, Boston, MA, 101-116. (2010)
62. Jo, J.M.: Effectiveness of normalization pre-processing of big data to the machine learning performance. *The Journal of the Korea Institute of Electronic Communication Sciences*, Vol. 14, No. 3, 547-552. (2019)
63. Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, Second Edition, John Wiley. (2006)
64. Frank, A. Asuncion, UCI Machine Learning Repository, Univ. California, School Inf. Comput. Sci., Irvine, CA. [Online]. Available: http://archive.ics.uci.edu/ml/citation_policy.html
65. Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL Data Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework, *J. Multiple-Valued Logic Soft Comput.*, Vol. 17, 255–287. (2011).

Arumugam Saravanan completed his Ph.D. in Computer Applications from Anna University, Chennai. He is currently Associate Professor, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India. He has 23 years of experience teaching undergraduate and graduate students, as well as 12 years of research experience. He has published more than 30 papers in national and international journals as well as at conferences. His areas of interest and research include web security, network security, machine learning, web mining, and database management.

Damotharan Anandhi received her Doctoral Degree in computer science at Bharathiar University, Coimbatore, Tamil Nadu, India. Presently she works as an assistant professor at the Coimbatore Institute of Technology in Coimbatore, Tamil Nadu, India. She has 18 years of job experience in the field of education. She has published a number of research articles in prestigious national and international journals.

Marudhachalam Srividya received her doctoral degree in computer science at Bharathiar University, Coimbatore. She is currently an Assistant Professor at the Coimbatore Institute of Technology, Coimbatore, India, and has 18 years of work experience in the area of teaching. She has several research publications in well-known

international journals and conferences. She has conducted many programmes at CIT, which include workshops and a short-term course under the QIP.

Received: October 30, 2021; Accepted: December 30, 2022.

Comprehensive Risk Assessment and Analysis of Blockchain Technology Implementation Using Fuzzy Cognitive Mapping

Somayeh Samsamian¹, Aliakbar Hasani^{2,*}, Saqib Hakak³, Fatemeh Esmailnezhad Tanha⁴, and Muhammad Khurran Khan⁵

¹ Master of Business Administration, Department of Industrial Engineering and Management, Shahrood University of Technology

² Associate professor, Department of Industrial Engineering and Management, Shahrood University of Technology
aa.hasani@shahroodut.ac.ir

³ Canadian Institute for Cybersecurity, Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

⁴ Ph.D. student, Department of Economics, Marche Polytechnic University, Ancona, Italy

⁵ Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

Abstract. Identifying and assessing potential risks of implementing new technologies is critical for organizations to respond to them efficiently during the technology life cycle. Blockchain has been introduced as one of the emerging and disruptive technology in the field of information technology in recent years, which system developers have noted. In this study, a comprehensive set of risks have been identified and categorized based on the literature findings to identify the risks of blockchain implementation. Critical risks are defined by performing a two-stage fuzzy Delphi method based on the experts' opinions. Then, possible causal relationships between considered risks are identified and analyzed using the fuzzy cognitive mapping method. Finally, the most important risks are ranked based on the degree of prominence and the relationships between them. Industry enterprise resource planning system based on blockchain technology has been studied as a case study. The obtained results indicate that the technology's immaturity has the most impact, the high investment cost is the most impressive risk, and privacy has a critical role in risks relationships. In addition, the high investment cost has the highest priority among other risks and the privacy and issues with contract law are ranked second and third, respectively.

Keywords: Risk Assessment, Blockchain, Fuzzy Cognitive Mapping, Fuzzy Delphi, Enterprise Resource planning.

1. Introduction

Implementing new technology is one of the organization's tactical decisions; it will have significant and long-term effects on the organization's processes and overall

* Corresponding author

performance. Hence, the successful implementation of new technology is important [1]. In recent years, after the Internet emersion, blockchain technology has been introduced as one of the most important information technologies [2]. Bitcoin and Ethereum are the most popular cryptocurrencies developed based on blockchain technology. The early adopters of this technology were traders and merchants [3]. Blockchain usage has increased continuously because of its extensive advantages such as decentralization, immutability, transparency, security, and anonymity [4]. Nevertheless, these significant features will not be effective without recognizing and analyzing the risks of blockchain implementation [5]. According to the conducted studies in the literature, various risks of blockchain implementation have been identified in an enterprise. One of the identified risks is a lack of technology maturity, which has significant effects on how issues such as governance and authority are conceptualized and performed [6]. Scalability is another risk of blockchain implementation that arises when better and qualified technological infrastructures are needed to more efficiently launch blockchain technology [7]. The other risk is facing new concepts without sufficient awareness during blockchain implementation, which makes using this technology difficult for users. Some of these complicated concepts are public key, private key, and cryptography. In addition, the lack of skilled human resources is another risk of blockchain implementation [8]. In addition, emerging technology implementation needs high investment without necessarily a short return period [9]. All potential risks should be identified and evaluated for successful blockchain implementation (see Figure 1).

In recent years, enterprise resource planning system (ERP) has attracted more attention as an efficient solution for integrating all business processes in the organization. Data exchange security is a key point of security issues in ERP systems. Blockchain technology offers an opportunity to build highly integrated, smarter, secure, and flexible ERP systems [10]. Therefore, it is expected to see more development of blockchain-based ERP systems in future years.

By increasing in number as well as types of risks and then potential causal relationships between them with high uncertainty, an integrated and systematic risk analysis model is required to consider experts' opinions. Despite the importance of this issue, only a few studies have been conducted on the risk identification and analysis of blockchain implementation. Therefore, in this study, the comprehensive set of risks of blockchain implementation has been identified based on the literature and experts' opinions. Then the potential causal relationships between these risks have been analyzed. Industry ERP systems based on blockchain technology have been studied as a case study in Iran. Experts' opinions are extracted via the two-phase fuzzy Delphi method. Next, the final identified risks are weighted and prioritized using the fuzzy cognitive mapping (FCM) technique.



Figure 1. Steps of successful blockchain implementation in the enterprise

2. Literature Review: Blockchain Implementation Risks

This section provides a comprehensive review of blockchain implementation risks which are categorized into eight general groups as follows: Technical (T), Security (S), Organizational (O), Legal (L), Financial (F), Environmental (E), Cultural (C), and Social risks (I).

Staples and Chen [11] studied the risks and opportunities of using blockchain systems and smart contracts. They found that providing a neutral ground between organizations would reduce the technical risks compared to centralized databases and blockchain computing platforms. Kim and Kang [12] investigated the potential risks and challenges of blockchain technology as a means of eliminating illicit activities in various domain areas such as supply chain and logistics, government and public sectors, and international trade. They realized that blockchain technology may not always bring socio-economic benefits without a strategic planned policy. In another study, Zamani et al. [13] analyzed blockchain security risks at the operational level. For this purpose, required standards and rules related to blockchain implementation are investigated and analyzed in numerous blockchain incidents to determine the root cause of the most vulnerable aspects of this technology. Harris [14] also discussed the risks of blockchain relying, such as manipulation of the majority consensus, limiting the access of minors, privacy, anonymity, and pseudo-anonymity, speed and accuracy of transaction, scalability and storage issues, taxation, regulation, and issues with contract law in underdeveloped countries for transparent transaction among parties, reducing corruption and facilitating trust. Lu and Huang [4] examined blockchain implementation risk in four aspects of the trade, management and decision making, monitoring, and cyber security in the oil and gas industry. The results of this study showed that there is not

enough understanding of blockchain in the oil and gas industry. The current blockchain implementation status in the oil and gas industry is still experimental and investment is not enough compared with available capacities. Blockchain can provide many opportunities for this industry, such as reducing transaction costs and improving transparency and efficiency. In another study, Bürer et al. [15] focused on applied use cases for implemented blockchain architectures in the aspect of energy usage, risks and opportunities while guaranteeing a reliable distribution network and supply security are achieved. Norta and Matulevičius [16] examined the protection of an official formal blockchain authentication protocol by using security risk patterns. Based on the results, they have identified some major risks that threaten the protocol. Sayeed and Marco-Gisbert [17] evaluated the agreement of blockchain and security mechanisms against 51% of attacks in aspect of several main security risks. Their analysis presented that all of the applied security techniques are failed to protect against mentioned attack, lack power of the implemented security policies, and need to stronger policy to overcome this failure. In another study, Prewett et al. [18] mentioned that blockchain adoption as a transformative technology is unavoidable for future business enterprises. Therefore, appropriate attention to risks and challenges before, during, and after blockchain implementation will guarantee long-term success. Furthermore, Feng et al. [19] examined the cyber risk management of blockchain networks with a theoretical game approach. They have proposed a new approach to cyber risk management for blockchain services. In particular, they used cyber insurance as an economic tool to counteract the cyber risks posed by attacks on blockchain networks. They considered a blockchain services market which is consisted of infrastructure, a blockchain provider, an internet insurer, and users. Furthermore, White et al. [5] surveyed fundamental technologies of private blockchain and how the auditor can evaluate and respond to the risks of blockchain applications. Biswas and Gupta [20] analyzed the risks of blockchain implementation in industries and services. This study presented a framework for investigating blockchain risks to acceptance and its successful implementation in various industries using the DEMATEL technique. They identified and categorized a group of risks by using the existing literature and experts' opinions. Afterwards, they evaluated the causal relationships among these risks and ranked them based on their degree of prominence and relationships. This study's results showed that scalability and market-based risks are the most significant risks.

At the same time, high sustainability costs and inappropriate economic behavior have the greatest impact on the successful blockchain implementation. In another study, Öztürk and Yıldızbaşı [9] examined the risks of blockchain implementation in supply chain management using the multi-criteria decision-making method. This study determined the existing risks in supply chain processes with blockchain technology and evaluated these risks emerging during technological transformation. This study discussed security, financial, organizational, and environmental risks. They used Fuzzy hierarchical analysis and fuzzy TOPSIS methods. The obtained results are as follows: (a) high investment costs, data, and facilities security are important, (b) less complex supply chain integrations can coordinate faster than blockchain technology development, and (c) integration is harder for health and logistic sectors. Moreover, Özkan et al. [21] evaluated the risks of blockchain technology using the multi-criteria decision-making method based on Fuzzy Pythagorean sets. Their goal was to find the most vital risks for real-life case studies. This process considered organizational, environmental/cultural, security, technical and financial risks prioritization. As a result,

security related risks are identified as the most important. In another study, Drljevic et al. [22] examined perspectives on risks and standards affecting blockchain technology requirements' engineering. This study's results indicate a gap in the normative frameworks that affect the sustainable adoption and use of blockchain technology.

Despite the importance of analyzing implementation risks of blockchain as a disruptive technology, only a few studies have focused on this issue. Therefore, the comprehensiveness risks assessment model is developed in this study to cover an extensive set of potential risks and ultimately analyze their causal relationships. Tables 1 and 2 summarize the conducted studies on identification and evaluation of blockchain implementation risks and comparison of current study with other studies.

Table 1. Classification of potential risks of blockchain (BC) implementation

Study	Type of risks							Analysis method	Case study	Results
	T	S	O	L	F	E	C			
KPMG [23]	*	*	*	*	*			Systematic Literature Review	Bank industry	Importance of risk assessment for increasing efficiency and effectiveness of BC implementation
Zetsche et al [24]				*				Descriptive research method	Legal Risks of Blockchain	Importance of considering legal related risks for successful BC implementation
Staples et al [11]	*							Description and analysis	Smart contracts	Importance of no limitation for BC type selection in accordance to organization requirements
Lindman et al [25]	*			*				Review of previous literature	Research agenda	Indicating on importance of risks of organizational issues, competitive environment, and technology design issues
Caron [26]	*	*	*	*	*			Descriptive research method	Identifying risk on the road to distributed ledgers	Increased in BC risks because of immaturity of regulatory framework for BC technology
Kim and Kang [12]	*	*	*	*				Descriptive research method	Anti-corruption	a holistic and coordinated effort is required because of a black box nature of a BC
Tarr [27]	*	*	*	*				Review of previous literature	Insurance industry	Via BC and linked smart contracts, there is considerable potential for frictional delays and the risks of human error to be controlled.
Harris [14]	*	*	*	*	*			Fundamental research method	Blockchain technology in underdeveloped countries	BC technologies deal significant hurdles for implementation in underdeveloped countries
Santhana and Biswas [28]	*	*	*	*	*	*		Fundamental research method	Risk performance in China's strategy	To respond to BC risks, organizations should consider establishing a robust risk management strategy, governance, and controls framework.
White et al [5]	*	*	*	*	*			Descriptive research method	Blockchain security risk assessment and the auditor	Indicating on risks include technological risks, data security risks, interoperability risks, and third-party vendor risks.
Özkan et al[21]	*	*	*	*	*	*	*	MCDM (PF-AHP)	British telecommunication company	Security and its related risks have more critical during BC implementation.

Table 2. Classification of potential risks of blockchain implementation

Study	Type of risks										Analysis method	Case study	Results
	T	S	O	L	F	E	C	I					
Biswas and Gupta [20]	*	*	*	*	*	*	*	*	*	*	DEMATEL	Industry and Services	Top importance of scalability, transaction-level risks, market-based risks, and regulatory risks.
Sayeed and Marco-Gisbert [17]	*	*	*	*	*	*	*	*	*	*	Analytical descriptive	Mechanisms against the 51% attack	A security policy accepting a restricted number of blocks by totally disregarding the longest chain rule must be discovered to diminish the risks of 51% attack successfully.
Norta, et al [16]	*	*	*	*	*	*	*	*	*	*	Project management life cycle	General	Security requirements and -controls necessary process are proposed to mitigate the BC risks.
Lu, et al [4]	*	*	*	*	*	*	*	*	*	*	Systematic Literature Review	Oil and gas industry	Indicating on primarily technological, regulatory and system transformation risks.
Bürer, et al [15]	*	*	*	*	*	*	*	*	*	*	Fundamental research method	Energy industry of emerging business models and related risks	Indicating the importance of reliability/stability risks of power resources management.
Prewett, et al [18]	*	*	*	*	*	*	*	*	*	*	Systematic Literature Review	General	Indicating on reputational and business continuity risks.
Wang [29]	*	*	*	*	*	*	*	*	*	*	Fuzzy Networks	Supply chain Financial Risk	Indicating on supply chain financial credit, financing enterprises, core enterprises, and the overall operation of supply chain finance risks.
Drijevic, et al [22]	*	*	*	*	*	*	*	*	*	*	Neural Networks	General	Indicating on an identified gap in normative frameworks that affect the adoption and sustainable use of BC technology.
Öztürk and Yildizbaşı [9]	*	*	*	*	*	*	*	*	*	*	Systematic Literature Review	Supply chain management	Indicating on investment risk, data security and utility risks and coordination and integration risks.
Esmailnezhad Tanha et al. [33]	*	*	*	*	*	*	*	*	*	*	Fuzzy Delphi and FBWM	Cyber physical systems	Top importance of technical challenges for BC implementation
Zhang and Song [34]	*	*	*	*	*	*	*	*	*	*	BWM	Supply chain	Top priority of cooperation complexity and increased costs
Nguyen et al., [35]	*	*	*	*	*	*	*	*	*	*	Mixed-methods risk analysis	Maritime container shipping	Gap of encouraging legal and technological environments for BC implementation
Sadeghi et al., [36]	*	*	*	*	*	*	*	*	*	*	Fuzzy MCDM	Construction organizations	Critical blockchain risks facing construction organizations are communication and information, supply chain management, financial, and corporate social responsibility
Gorbunova et al., [37]	*	*	*	*	*	*	*	*	*	*	Systematic Literature Review	BC applicability in various sectors	A comparative analysis of the challenges that could become a baseline for potential future research activities in the BC and DLT fields
Current Study	*	*	*	*	*	*	*	*	*	*	Fuzzy Delphi and FCM	Blockchain-based software development	Prioritizing all of the considered risk based on the holistic model

3. Research Method: Integrated Fuzzy Delphi and FCM

In this study, an integrated analysis technique incorporates the fuzzy Delphi technique, and the fuzzy cognitive map is applied. Uncertainty of risks assessment process is handled by fuzzy approach.

3.1. Fuzzy Delphi Technique: Risk Identification and Assessment

In this study, the Fuzzy two-phase Delphi method is used to identify and evaluate the potential risks based on the experts' opinions. For this purpose, verbal expressions are used to measure the extracted viewpoints. In the first phase, a semi-open questionnaire has been developed for potential risks identification and assessment based on the results of the conducted studies in the literature. The proposed risk name, definition, and classification are validated based on the experts' opinions. New risks and their related information could be proposed by experts. The fuzzy technique is applied to handle the uncertainty of risk assessment by representing verbal expressions with the triangular fuzzy number (TFN) (see Table 3) [30]. Based on the final results of the first phase, the second phase questionnaire is developed. In the second phase, the relative importance of concluded risks is evaluated by experts. Achievement of consensus (i.e., results from convergence status) is calculated at the end of the second phase. The Delphi evaluation process will be finalized if this consensus measure (i.e., the difference between the averages of two successive obtained defuzzy results of Delphi phases) is less than 0.1. The applied Delphi method is stopped at the second phase.

Table 3. Triangular fuzzy numbers for a five-point scale [31]

Verbal expressions		TFN
FCM	Fuzzy Delphi	
Strong positive	Strongly agree	(0.6,0.8,1)
Positive	Agree	(0.4,0.6,0.8)
Ineffective	Neutral	(0.2,0.4,0.6)
Negative	Disagree	(0,0.2,0.4)
Strong negative	Strongly disagree	(0,0,0.2)

3.2. Fuzzy Cognitive Map: Cause-and-Effect Analysis

In this study, the FCM method has been used as an analysis tool which uses a graph-based system to present the causal relationships of influencing risk factors in decision-making. The analysis graph of the FCM consists of two key elements: node and edge.

Nodes represent the main factors of the concepts that define a system of blockchain implementation risks analysis. Edges represent the potential causal relationships between the considered nodes. Fuzzy binary numerical descriptions are used to introduce causal effects into the cognitive map instead of positive or negative symbols. The edge of each fuzzy cognitive map between concepts (i.e., risks) of C_i and C_j is

related to a relative weight variable from -1 to 1. This variable indicates the strength of the related relationship. There are three different types of possible causes between each pair of risks, C_i and C_j [32]:

- $W_{ij} > 0$ which determines a positive cause. If the value of C_i increases, this will lead to an increase in C_j value.
- $W_{ij} < 0$ which determines negative causation. If the value of C_i increases, this will decrease C_j value.
- $W_{ij} = 0$ which indicates no causal relationship between considered concepts.

The FCM of considered concepts is represented mathematically by an adjacency weight matrix with $n \times n$ size. The three types of variables in the cognitive map are as follows: transmitter variables, receiver variables, and ordinary variables. These variables are calculated via their out-degree (*OUT*) and in-degree (*IN*). *OUT* and *IN* are the row and column sums of absolute values of a variable in the adjacency matrix that present the cumulative strengths of exiting relations variables, respectively (see equations 1-2). Transmitter, receiver, and ordinary variables have a positive *OUT* and zero *IN*, a positive *IN* and zero *OUT*, and both a non-zero *IN* and *OUT*, respectively. The centrality of a variable (*IMP*) is the summation of the related *IN* and *OUT* indexes (see equation 3).

$$IN = \sum_{i=1}^n W_{ik} \tag{1}$$

$$OUT = \sum_{k=1}^n W_{ik} \tag{2}$$

$$IMP = IN + OUT \tag{3}$$

It should be mentioned that a three-point Likert scale has been used to present the experts' opinions about the effect of each risk on others (see Table 2). Fuzzy triangular evaluation (FTE) is presented by equations (4). Rank of each risk is determined based on the related calculated FTE. The mean of fuzzy number (MFN) is calculated by equation (5) for conducted assessment. All final fuzzy evaluations are defuzzified by equation (6).

$$\text{Fuzzy evaluation (FTE)} = (m_l^i, m_m^i, m_u^i) \tag{4}$$

$$\text{Mean of fuzzy number (MFN)} = \frac{\sum_{i=1}^n (m_l^i, m_m^i, m_u^i)}{n} \tag{5}$$

$$W^{df} = \frac{m_l + 2m_m + m_u}{4} \tag{6}$$

4. Research Findings

In this section, based on the applied two-phase fuzzy Delphi and fuzzy cognitive mapping, the main results of identifying and evaluating blockchain implementation risks are presented for the blockchain-based ERP software as a case study.

Table4. Identified risks based on the literature review as an input of the applied Delphi method include risk type (TR) and name (R)

TR	Risk	TR	Risk			
Technical (T)	Architecture and design risk	T1	Data security risks	S1		
	Oracle risk	T2	Cryptography, key management, and tokenization	S2		
	Speed and accuracy of transactions	T3	Cyberattacks	S3		
	Consensus mechanism and network management	T4	Vulnerability	S4		
	Lack of technological maturity	T5	Transaction leakage	S5		
	Lack of customer awareness	T6	Privacy	S6		
	Level of Access to technology	T7	Criminal activity	S7		
	Sustainable infrastructure lackness	T8	Double spending	S8		
Financial (F)	Limiting the Access of Miners	F1	Security (S)	The complexity of the blockchain system compared to existing systems	S9	
	High investment cost	F2		Compatibility of different blockchain platforms	S10	
	Usage cost	F3	Social (I)	Information sharing obstacles	I1	
	Taxation	F4		Environmental (E)	Wasted resources	E1
	Scalability and performance	F5	High energy Consumption		E2	
	Technology Implementation and Acquisition	F6	Cultural (C)		Smart contract risk	C1
	Training cost	F7			Blockchain myths	C2
	Storage Limitations	F8			Lack of implement transparent structure	C3
	Lack of research and development Units	F9			Participatory persuasion	C4
	Resistance from the incumbents	O1		Tracking transactions	C5	
	Vendor risk	O2	Legal (L)	Compatibility risk	L1	
	Distinctive opportunities	O3		Issues with Contract Law	L2	
	Applicability to use blockchain as a solution	O4		Regulation	L3	
	Chain defense	O5		Working within limitations of blockchain	L4	
Business continuity and disaster recovery	O6	Jurisdiction		L5		
Lack of skilled human resources	O7	Data management and segregation		L6		
Lack of management support	O8	Compliance risk		L7		
Lack of equipment and tool	O9	Data control		L8		
Resistance to change technology	O10	User identity		L9		
Strong hierarchical structure and bureaucracy	O11	Decentralization		L10		
Strict administrative control	O12	Regulatory Hurdles		L11		
Mind set of people needs to be changed	O13	Lack of control over malicious operations and information		L12		
Stable network connection	O14					
Organizational (O)						

4.1. Case study profile: Blockchain-based ERP Systems

In this section, based on the applied two-phase fuzzy Delphi and fuzzy cognitive mapping, the main results of identifying and evaluating blockchain implementation risks are presented for the blockchain-based ERP software as a case study.

The market size of global ERP software was achieved at USD 50.57 billion in 2021 and is expected to keep growing in future years. The ERP system would provide a centralized and integrated system for enterprises. The capabilities of ERP systems could be boosted by integrating with blockchain technology. Once blockchain is integrated into the ERP system, it optimizes system operations, internal data control, and business processes such as intercompany transactions. A client-server technology is at the core of an enterprise's integrated management systems, which are now information-centric systems that use common standards for communication infrastructure, applications, databases, data exchange, and security [10]. The integration of ERP and blockchain systems improves the transparency and reliability of financial transactions in financial and accounting systems. In addition, potential contradictions in terms of invoices, shipments, returns and purchases are also reduced.

Integrating ERP with blockchain has twofold benefits: creating more transparency and reducing costs. Blockchain uses its capabilities to monitor business processes and facilitate their entry into the blockchain network. Finally, this study investigates the risks of implementing ERP systems based on blockchain in one of the biggest information technology and services companies in Iran. The company name could not be mentioned because of a confidential issue. Research experts are selected in the field of blockchain technology for developing ERP systems.

4.2. Results of Identifying and Evaluating Risks

The final summary of the identified risks of blockchain implementation based on the literature review is presented in Table 4. These findings are considered input data for the applied two-phase fuzzy Delphi method. The final results of the second phase of fuzzy Delphi are presented in Table 5.

According to the evaluation of identified risks of blockchain implementation, the value of 0.7 has been determined as the priority threshold based on the experts' viewpoints to prepare the FCM questionnaire for risk assessment. For this purpose, a set of 24 risks has been selected with priority values greater than the considered threshold. These risks have been analyzed and evaluated using the FCM approach to investigate the potential causal relationships. Then, influential risks are identified by analyzing each risk's impact on other risks. By preparing a questionnaire, experts were asked to examine the risks carefully and use verbal expressions via the Likert scale to determine the type and intensity of the impact of each risk on others.

Finally, weak casual relations between risks are removed from the constructed cognitive map because of the realized low importance weight. For instance, two relations, including O8-L4 and L1-T5, are removed from the map. For developing a group-based FCM, a simple average of obtained fuzzy evaluation for each casual relation is calculated and defuzzified. The final calculated weights of risks are presented in Table 6.

Table 5. Ranking of the final identified risks by the Delphi method

Risk	Score value		Rank	Risk	Score value		Rank
	FTE	MFN			FTE	MFN	
T1	2.666	0.444	29	O5	3.000	0.500	12
T2	2.867	0.478	19	O6	2.600	0.433	31
T3	4.000	0.667	1	O7	2.400	0.400	43
T4	3.667	0.611	6	O8	3.200	0.533	7
T5	2.800	0.467	22	O9	2.600	0.433	35
T6	3.200	0.533	7	O10	2.600	0.433	31
T7	2.067	0.344	52	O11	2.267	0.378	47
T8	3.000	0.500	12	O12	2.067	0.344	52
S1	2.467	0.411	36	O13	2.067	0.344	50
S2	2.733	0.456	27	O14	2.867	0.478	20
S3	2.267	0.378	47	E1	2.467	0.411	36
S4	3.000	0.500	12	E2	3.000	0.500	12
S5	1.667	0.278	61	L1	3.800	0.633	3
S6	3.000	0.500	12	L2	3.800	0.633	4
S7	2.467	0.411	36	L3	4.000	0.667	1
S8	3.200	0.533	54	L4	2.800	0.467	22
S9	2.600	0.433	21	L5	3.200	0.533	7
S10	2.600	0.433	43	L6	2.400	0.400	43
F1	1.933	0.322	56	L7	1.867	0.311	58
F2	3.200	0.533	7	L8	1.933	0.322	56
F3	2.400	0.400	40	L9	2.733	0.456	27
F4	2.267	0.378	46	L10	1.800	0.300	60
F5	3.000	0.500	12	L11	2.600	0.433	30
F6	3.200	0.533	7	L12	2.200	0.367	49
F7	2.600	0.433	31	I1	2.467	0.411	36
F8	2.000	0.333	54	C1	1.867	0.311	58
F9	2.600	0.433	31	C2	2.800	0.467	22
O1	2.067	0.344	50	C3	3.800	0.633	4
O2	2.800	0.467	22	C4	3.000	0.500	12
O3	2.400	0.400	40	C5	2.400	0.400	40
O4	2.800	0.467	22				

In Table 7, weights of unrelated risks are zero and weights of related risks are non-zero, which are presented in blue-colored cells. Then, the graph-based structure of the proposed fuzzy cognitive map is analyzed using FCMAPPER software. The output of this mathematical analysis obtained based on the Graph theory is investigated. The final ranking of each risk is done based on the centrality index (see Table 7).

The in-degree and out-degree indicate whether the considered risk mainly influences other risks or if other risks influence it or both, respectively. The contribution of each risk in the FCM can be regarded by determining its centrality, which indicates how connected the risk is to other risks and what the cumulative strength of these connections is. The obtained results of the proposed FCM show that blockchain immaturity has the highest impact. The high investment cost risk has been influenced greatly by other risks. Privacy has the highest centrality index. Then, the graph of the proposed FCM for presenting casual relationships between considered risks is analyzed by PAJEKT software depicted by Gephi software (see Figure 2).

Table 6. Final FCM results (per hundred)

	T2	T3	T4	T5	T6	T8	S2	S6	S9	F2	F5	F6	O4	O5	O8	E2	L1	L2	L3	L4	L9	C2	C3	C4
T2	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40
T3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	53	0	0	0	0	59	0	0	0	0	60	0	0	0	0	0	0	0	67	0
T6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	60	0	0
T8	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	60	0	0	0	0	0	0	0	0
S2	0	40	60	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0
S6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	60	0	0
S9	0	0	0	53	0	0	0	0	0	33	0	0	0	0	0	0	0	0	53	0	0	0	63	0
F2	0	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	0	0	0	0	0	0	0	0
F5	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0
F6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	27	0	60	0	0	0	0	0	0	0
O4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0
O5	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O8	0	0	71	0	0	0	67	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	22
E2	0	0	0	0	0	0	0	0	0	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	37	0	0	0	0	0
L2	0	0	0	0	0	0	0	67	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
L3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L4	57	37	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0
L5	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	0	0	0	0	0	0	0	0	67	0	0	0	0	0	0	0	0	0	0	70	0	0	0
C4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43	0	0	0	0	0	0

Table 7. FCM Measures Summary

	Risk	Out-degree	In-degree	Centrality
S6	Privacy	1.07	0.35	3.42
L2	Issues with Contract Law	1.17	2.23	3.40
F2	High investment cost	0.70	2.69	3.39
O8	Lack of management support	2.23	0.87	3.02
T5	Lack of technological maturity	2.39	0.53	2.92
L1	Compatibility risk	1.07	1.83	2.90
L4	Working within limitations of blockchain	1.43	1.40	2.83
S2	Cryptography, key management, and tokenization	1.63	1.13	2.76
C3	Lack of implement transparent structure	1.37	1.70	2.67
O5	Chain defense	0.37	0	2.07
S9	The complexity of the blockchain system compared to existing systems	2.03	1.30	2.03
L9	User identity	0.63	1.31	1.93
T4	Consensus mechanism and network management	0.57	1.10	1.88
E2	High energy Consumption	0.73	1.23	1.83
L3	Regulation	0.60	0.53	1.83
T6	Lack of customer awareness	1.20	0.70	1.73
F6	Technology Implementation and Acquisition	0.88	0.57	1.58
T2	Oracle Risk	0.87	0.77	1.44
T3	Speed and accuracy of Transactions	0.57	0	1.37
T8	Lack of sustainable energy infrastructure	1.30	0.60	1.30
C2	Blockchain myths	0.53	0.62	1.13
C4	Participatory persuasion	0.43	0.62	1.05
F5	Scalability and maintenance	0.72	0.33	1.05
O4	Applicability to use blockchain as a solution	0.63	0.01	0.64

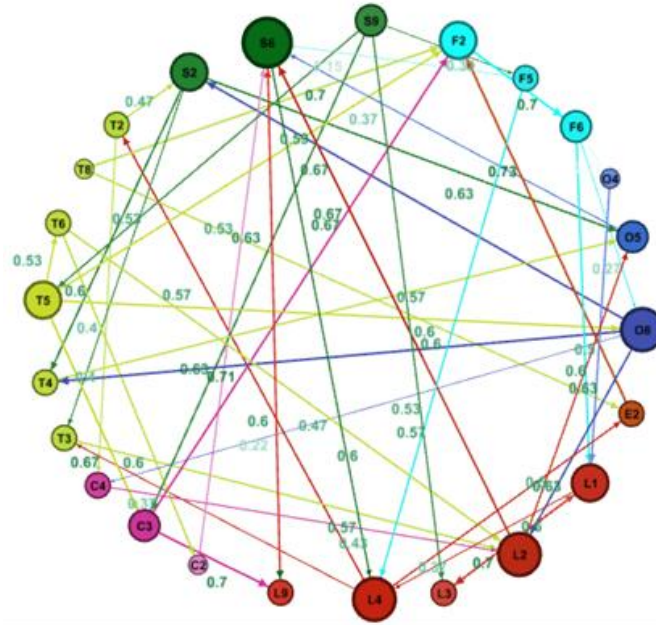


Figure 2. Proposed FCM for risks assessment of blockchain implementation

In figure 2, each weight of two risks (i.e., nodes) relation value is presented above the related arrow. The set of risks with high impacts are selected as the most important risks of blockchain implementation (see Table 8).

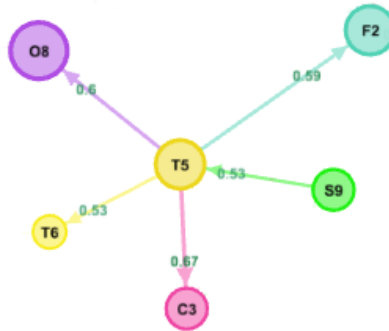


Figure 3. Cluster of lack of technological maturity risk (T5)

Analysis of the out-degree index shows that technological immaturity has the most impact on other risks. Therefore, figure 3, the network cluster includes the risks related to the technology immaturity risk that should be analyzed. The results indicate that to reduce the risk of lack of technological maturity is necessary to enhance the lack of management support (O8). Also, other factors that increase high investment cost (F2),

such as high energy consumption of electricity and other similar things, should be minimized. The lack of implementation of a transparent structure (C3) due to the newness and anonymity of blockchain technology for users is another risk that is affected by the immaturity of blockchain technology. Improving the users' knowledge of blockchain technology at the various organizational levels could be an efficient strategy for dealing with the mentioned risk.

Table 8. Final ranking of blockchain implementation risks

Risk	Rank	Risk	Rank
F2 High investment cost	1	F6 Technology Implementation and Acquisition	13
S6 Privacy	2	T3 Speed and accuracy of Transactions	14
L2 Issues with Contract Law	3	T2 Oracle Risk	15
O5 Chain defense	4	C4 Participatory persuasion	16
L1 Compatibility risk	5	C2 Blockchain myths	17
L4 Working within limitations of blockchain	6	T6 Lack of customer awareness	18
L9 User identity	7	T5 Lack of technological maturity	19
T4 Consensus mechanism and network management	8	F5 Scalability and maintenance	20
L3 Regulation	9	O4 Applicability to use blockchain as a solution	21
C3 Lack of implement transparent structure	10	O8 Lack of management support	22
S2 Cryptography, key management and tokenization	11	T8 Lack of sustainable energy infrastructure	23
E2 High energy Consumption	12	S9 The complexity of the blockchain system compared to existing systems	24



Figure 4. Cluster of high investment cost (F2)

Figure 4 shows high investment costs as the most influential risk of blockchain implementation, from other risks. The related cluster includes critical risks such as high energy consumption, lack of sustainable energy infrastructure, and lack of implement transparent structure. Providing more efficient and reliable energy resources could decrease the operational cost of using blockchain technology and diminish an organization's vulnerability to this technology implementation.

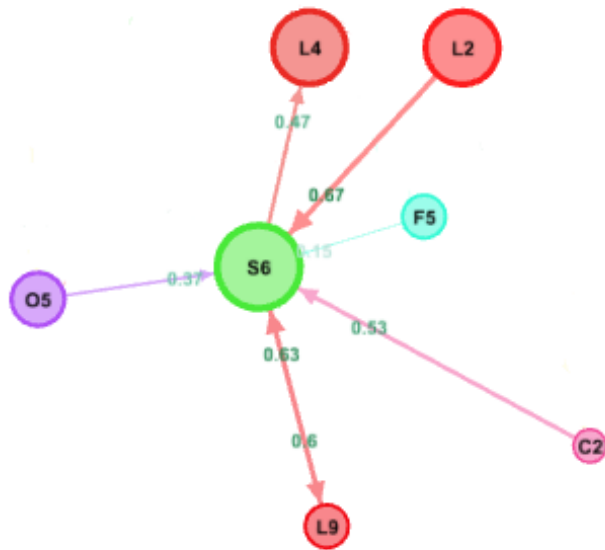


Figure 5. Cluster of privacy risk (*S6*)

Figure 5 presents the critical role of risks related to law issues such as contract law issues, working within blockchain limitations, and user identity. In addition, chain defence methods of mining pool attacks for blockchain security issues, network communication and smart contracts for blockchain security issues, and privacy thefts for blockchain privacy issues are so important in analyzing the privacy risk. Blockchain would be defined in three general types: public, private, and consortium. The private blockchain has a lot of supervision, which is only under certain individuals' control. Accordingly, a private blockchain would be used as much as possible to deal with the risk of privacy.

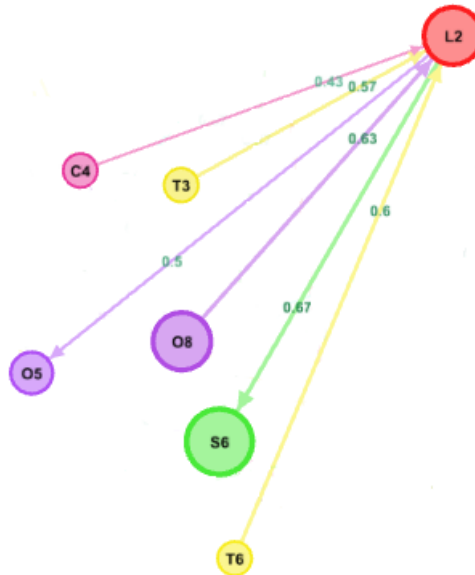


Figure 6. Cluster of risk of contract law (L2)

Figure 6 presents the cluster of contract law as a second important risk in term of the centrality index. Obtained results indicate that technological issues such as transaction speed and accuracy as well as customer awareness have the most impact on this critical risk. In addition, organizational and security aspects of an enterprise, such as chain defense and privacy, are affected by the risk of contract law.

5. Conclusion

Blockchain has a great potential for changing attitudes toward traditional businesses to be more cost-effective and reliable. Blockchain is an innovative technology which has been accepted widely by various industries. This emerging technology has several advantages, such as eliminating intermediaries, transparency, and traceability. Nowadays, businesses are exploring how to use this emerging technology to efficiently influence their enterprise and avoid the implementation of potential risks. Therefore, identifying and assessing the extensive implementation risks are very important and have a critical impact on organization performance. Risk management could be a more challenging task by increasing the number of risks and potential causal relationships between them. For this purpose, the fuzzy cognitive mapping technique has been used in this study to analyze the complex system of blockchain risk implementation as a disruptive technology. In this study, the evaluation and analysis of blockchain implementation risk are handled via the fuzzy cognitive mapping technique. For this purpose, after a comprehensive literature review, the potential risks of blockchain implementation have been identified and examined in eight general categories: technical, security, organizational, legal, financial, environmental, cultural, and social.

Finally, various risks have been identified and investigated based on the experts' opinions using the two-phase fuzzy Delphi method for determining the most important risks. Then, constructed map of risks are analyzed via FCM in terms of influencing other risks, affected by other risks, and impotence in the risks network.

Obtained results indicate that financial risk, including high investment cost is the most important implementation risk of blockchain as a revolutionary technology. This critical cost-based risk has been mentioned in other studies because of high energy dependence, the difficult process of integration, and the implementations high costs [34], [36]. By developing new generation of blockchain technology, these technologies based challenges would be more improved in term of operational cost of using blockchain. Law related risks have a significant role among the top ten assessed risks. This importance could be seen by analyzing the central index for critical risks such as privacy. In addition, issues with contract law have various critical impacts on the organizational and privacy risks and are impacted by technology risks. Regarding the environmental context, specific laws and regulatory support were considered as the most important factors [33], [35]. These key soft aspects should also be more developed in proper harmony with the common technological aspects of the blockchain technology, which have been of most noted until now. Although the immaturity of blockchain has a critical impact on other considered risks in term of the out-degree index, it is expected this influencing role decreases over time as a consequence of the further evolution of blockchain technology. The risk of the high investment cost of blockchain usage is affected by the novel as well as sustainable energy-providing approach. Required supportive infrastructures, including both technical and non-technical elements simultaneously, may stimulate the development, diffusion, commercialization, and penetration coefficient of new blockchain-based applications which could be integrated with others disruptive information technologies such as IoT, cloud-computing, and other cyber-physical systems [33].

Therefore, this threat with a high impact on other risks in term of out-degree index could be transformed into an opportunity by using more cost-efficient energy resources and outweighing obtained benefits by blockchain. Therefore, the assessed network of risks that have high dynamics should be analyzed by considering the effects of time on related issues and potential feedbacks over time. For this purpose, the systems dynamics analysis approach can be used for future researches.

References

1. Farshidi S, Jansen S, de Jong R, Brinkkemper S. A decision support system for software technology selection. *Journal of Decision systems*. 2018. 27(sup1): p. 98-110.
2. Efanov, D. and P. Roschin, The all-pervasiveness of the blockchain technology. *Procedia computer science*, 2018. 123: p. 116-121.
3. Böhme R, Christin N, Edelman B, Moore T. Bitcoin: Economics, technology, and governance. *Journal of economic Perspectives*. 2015. 29(2): p. 213-38.
4. Lu H, Huang K, Azimi M, Guo L. Blockchain technology in the oil and gas industry: A review of applications, opportunities, challenges, and risks. *Ieee Access*. 2019. 27(7): p. 41426-44.
5. White, B.S., C.G. King, and J. Holladay, Blockchain security risk assessment and the auditor. *Journal of Corporate Accounting & Finance*, 2020. 31(2): p. 47-53.
6. Swan, M., *Blockchain: Blueprint for a new economy*. 2015: " O'Reilly Media, Inc."

7. Vukolić, M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. in International workshop on open problems in network security. 2015. Springer.
8. Britchenko, I., T. Cherniavska, and B. Cherniavskiy, Blockchain technology into the logistics supply. 2018.
9. Öztürk, C. and A. Yildizbaşı, Barriers to implementation of blockchain into supply chain management using an integrated multi-criteria decision-making method: a numerical example. *Soft Computing*, 2020. 24(19): p. 14771-14789.
10. Hrishev, R. ERP systems and data security. in IOP Conference Series: Materials Science and Engineering. 2020. IOP Publishing.
11. Staples M, Chen S, Falamaki S, Ponomarev A, Rimba P, Tran AB, Weber I, Xu X, Zhu J. Risks and opportunities for systems using blockchain and smart contracts. Data61. CSIRO, Sydney. 2017.
12. Kim, K. and T. Kang. Does technology against corruption always lead to benefit? The potential risks and challenges of the blockchain technology. in Paper submitted to OECD's Anti-Corruption and Integrity Forum. <https://www.oecd.org/cleangovbiz/Integrity-Forum-2017-Kim-Kang-blockchain-technology.pdf>. 2017.
13. Zamani, E., Y. He, and M. Phillips, On the security risks of the blockchain. *Journal of Computer Information Systems*, 2020. 60(6): p. 495-506.
14. Harris, C.G. The risks and dangers of relying on blockchain technology in underdeveloped countries. in NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. 2018. IEEE.
15. B Bürer MJ, de Lapparent M, Pallotta V, Capezzali M, Carpita M. Use cases for blockchain in the energy industry opportunities of emerging business models and related risks. *Computers & Industrial Engineering*. 2019. 137:106002.
16. Norta, A., R. Matulevičius, and B. Leiding, Safeguarding a formalized blockchain-enabled identity-authentication protocol by applying security risk-oriented patterns. *Computers & Security*, 2019. 86: p. 253-269.
17. Sayeed, S. and H. Marco-Gisbert, Assessing blockchain consensus and security mechanisms against the 51% attack. *Applied Sciences*, 2019. 9(9): p. 1788.
18. Prewett, K.W., G.L. Prescott, and K. Phillips, Blockchain adoption is inevitable—Barriers and risks remain. *Journal of Corporate accounting & finance*, 2020. 31(2): p. 21-28.
19. Feng S, Wang W, Xiong Z, Niyato D, Wang P, Wang SS. On cyber risk management of blockchain networks: A game theoretic approach. *IEEE Transactions on Services Computing*. 2018. 14(5): p. 1492-504.
20. Biswas, B. and R. Gupta, Analysis of barriers to implement blockchain in industry and service sectors. *Computers & Industrial Engineering*, 2019. 136: p. 225-241.
21. Özkan B, Kaya İ, Erdoğan M, Karaşan A. Evaluating blockchain risks by using a MCDM methodology based on Pythagorean fuzzy sets. In *International conference on intelligent and fuzzy systems 2019*. 23: p. 935-943. Springer, Cham.
22. Drljevic, N., D.A. Aranda, and V. Stantchev, Perspectives on risks and standards that affect the requirements engineering of blockchain technology. *Computer Standards & Interfaces*, 2020. 69: p. 103409.
23. KPMG, Realizing blockchain's potential, in Retrieved from <https://assets.kpmg/content/dam/kpmg/xx/pdf/2018/09/realizing-blockchains-potential.pdf>. 2018.
24. Zetsche, D.A., R.P. Buckley, and D.W. Arner, The distributed liability of distributed ledgers: Legal risks of blockchain. *U. Ill. L. Rev.*, 2018: p. 1361.
25. Lindman, J., V.K. Tuunainen, and M. Rossi, Opportunities and risks of Blockchain Technologies—a research agenda. 2017.
26. Caron, F., Blockchain: Identifying risk on the road to distributed ledgers. *ISACA Journal*, 2017. 5: p. 1-6.
27. Tarr, J.-A., Distributed ledger technology, blockchain and insurance: Opportunities, risks and challenges. *Insurance Law Journal*, 2018. 29(3): p. 254-268.

28. Santhana, P. and A. Biswas, Blockchain risk management–risk functions need to play an active role in shaping blockchain strategy. Accessed: Dec, 2017. 7: p. 2019.
29. Wang Y. Research on supply chain financial risk assessment based on blockchain and fuzzy neural networks. *Wireless Communications and Mobile Computing*. 2021 Feb 17;2021.
30. Ishikawa, A., et al., The max-min Delphi method and fuzzy Delphi method via fuzzy integration. *Fuzzy sets and systems*, 1993. 55(3): p. 241-253.
31. Habibi, A., F.F. Jahantigh, and A. Sarafrazi, Fuzzy Delphi technique for forecasting and screening items. *Asian Journal of Research in Business Economics and Management*, 2015. 5(2): p. 130-143.
32. Papageorgiou E, Papageorgiou K, Dikopoulou Z, Mouhrir A. A web-based tool for Fuzzy Cognitive Map Modeling, 2018. Fort Collins, USA.
33. Esmaeilnezhad Tanha, F., Hasani, A., Hakak, S., Reddy Gadekallu, T. Blockchain-based cyber physical systems: Comprehensive model for challenge assessment, *Computers and Electrical Engineering*, 2022. 103, 2022, 108347.
34. Zhang, F., Song, W., Sustainability risk assessment of blockchain adoption in sustainable supply chain: An integrated method, *Computers & Industrial Engineering*, 2022. 171, 108378.
35. Nguyen, S., Shu-Ling Chen, P., Du, Y. Risk assessment of maritime container shipping blockchain-integrated systems: An analysis of multi-event scenarios, *Transportation Research Part E: Logistics and Transportation Review*, 2022. 163, 102764.
36. Sadeghi, M., Mahmoudi, A. and Deng, X. Blockchain technology in construction organizations: risk assessment using trapezoidal fuzzy ordinal priority approach, *Engineering, Construction and Architectural Management*, 2022. <https://doi.org/10.1108/ECAM-01-2022-0014>
37. Gorbunova, M., Masek, P., Komarov, M., Ometov, A. Distributed Ledger Technology: State-of-the-Art and Current Challenges. *Computer Science and Information Systems*, 2022. 19 (1), 65-85.

Samayeh Samsami is a graduate student in the master of business administration from the Shahrood University of Technology. Her current research interests include Risk Management and Applications of CPS in logistics.

Aliakbar Hasani is an associate professor at the department of industrial engineering and management, Shahrood University of Technology. His current research interests include Risk Management and System Analysis.

Saqib Hakak is an assistant professor at the Canadian Institute for Cybersecurity, faculty of computer science, University of New Brunswick. His current research interests include Risk Management and Blockchain Technology.

Fatemeh Esmaeilnezhad Tanha is a graduate student in the master of business administration from the Shahrood University of Technology. Her current research interests include Risk Management and applications of CPS in logistics.

Muhammad Khurran Khan is a global thought leader and influencer in cybersecurity. He is a Professor of Cybersecurity at the Center of Excellence in Information Assurance, King Saud University.

Received: March 08, 2022; Accepted: January 25, 2023.

RESNETCNN:an Abnormal Network Traffic Flows Detection Model

Yimin Li^{1,*}, Dezhi Han¹, Mingming Cui¹, Fan Yuan², and Yachao Zhou²

¹ College of Information Engineering,
Shanghai Maritime University, Shanghai201306, China
{202130310117, mmcui}@stu.shmtu.edu.cn
dzhan@shmtu.edu.cn

² Hangzhou Anheng Information Technology Co.,
LTD, Hangzhou310051, China
{frank.fan, anna.zhou}@dbappsecurity.com.cn

Abstract. Intrusion detection is an important means to protect system security by detecting intrusions or intrusion attempts on the system through operational behaviors, security logs, and data audit. However, existing intrusion detection systems suffer from incomplete data feature extraction and low classification accuracy, which affects the intrusion detection effect. To this end, this paper proposes an intrusion detection model that fuses residual network (RESNET) and parallel cross-convolutional neural network, called RESNETCCN. RESNETCNN can efficiently learn various data stream features through the fusion of deep learning and convolutional neural network (CNN), which improves the detection accuracy of abnormal data streams in unbalanced data streams, moreover, the oversampling method into the data preprocessing, to extract multiple types of unbalanced data stream features at the same time, effectively solving the problems of incomplete data feature extraction and low classification accuracy of unbalanced data streams. Finally, three improved versions of RESNETCNN networks are designed to meet the requirements of different traffic data processing, and the highest detection accuracy reaches 99.98% on the CICIDS 2017 dataset and 99.90% on the ISCXIDS 2012 dataset.

Keywords: Intrusion detection, RESNETCNN, Deep learning.

1. Introduction

With the digitalization of the Internet [13], cyberspace security issues have become increasingly complex and diverse. On June 22, 2022, Northwestern Polytechnical University released a statement about a cyber attack in which multiple Trojan samples were found in the university's information network. In recent years, tens of thousands of malicious cyber attacks on domestic Chinese network targets, resulting in network devices have been controlled and high-value data have been stolen. Therefore, cybersecurity protection is extremely imminent. Intrusion detection is the detection of intrusions or intrusion attempts on a system through operational behavior, security logs, audit data, or other available network information, etc. When running on the inspected system, the Intrusion Detection System (IDS) [31] is responsible for scanning the current network activity of

* Corresponding author

the system, monitoring and recording the network traffic of the system. Then IDS detects, records and judges the legitimacy of various processes running on the system, filtering abnormal network traffic from the host according to defined rules, and finally provides real-time alerts, which can effectively guarantee the network security.

Existing intrusion detection systems may suffer from overfitting, low classification accuracy, and high false alarm rate (FPR) when facing [32] with huge and diverse network data. Jun ho Bang et al. [1] used HsMM to model WSN and developed an LTE signaling attack detection scheme. Compared with other detection schemes, the attack detection effect was better, but the accuracy was lower. M. Nazari et al. [26] proposed an ARIMA time series model and a new DoS and DDoS attack detection algorithm, which improved the classification performance of abnormal traffic, but had a high false positive rate.

Machine learning [42] has been widely used to identify various types of cyber attacks. Yang et al. [38] proposed an abnormal network traffic detection algorithm that integrates mixed information entropy and SVM to detect abnormal network traffic in cloud computing environment. K. Li [15] combined principal component analysis (PCA) and random forest (RF) algorithm to extract and combine features from different network layers, reducing redundancy and noise caused by multi-layer combination. However, most traditional machine learning methods are shallow learning, that usually emphasizes feature engineering and feature selection, which cannot solve classification problems for large-scale data in real network environments. And the classification accuracy of multiple classification tasks decreases with the dynamic growth of the dataset. Thus shallow learning is not suitable for intelligent analysis and high-dimensional massive data learning.

In the face of network traffic data with large scale and high dimensions, deep learning has greater advantages. Zhang [40] proposed an intrusion detection model based on deep hierarchical network, which combined CNN and LSTM (CNN_LSTM for short) and achieved good performance on CICIDS2017 data set. Zhong [41] proposed HELAD, a network abnormal traffic detection algorithm integrating multiple deep learning technologies. Although HELAD has better adaptability and detection accuracy, its bit error rate is slightly higher. At present, the intrusion detection system based on deep learning has the following two problems: 1. With the further increase of network traffic, the current intrusion detection system with high-speed detection capability is not very ideal in terms of packet capture capability and detection performance. 2. Data imbalance in real environment seriously affects the detection accuracy of most current intrusion detection systems.

To alleviate the above problems, an intrusion detection model, referred to as RESNETCNN is proposed that fuses RESNET and a parallel cross-convolutional neural network. The main contributions of this paper are as follows.

(1) By introducing the oversampling method to process the ISCXIDS 2012 dataset, we can extract multiple types of unbalanced data stream features simultaneously, which effectively solves the problems of incomplete data feature extraction and low classification accuracy in unbalanced dataset.

(2) The top layer of the proposed model adopts RESNET network structure and the bottom layer of the network adopts traditional convolutional neural network (CNN) structure. The combination of the top and bottom layers can effectively perform feature fusion and improve the detection accuracy of abnormal data streams in unbalanced datasets.

(3) Three improved versions of RESNETCNN are proposed. Simulation experiments

are conducted on the CICIDS 2017 and ISCXIDS 2012 datasets, and the corresponding detection accuracy can reach 99.98% and 99.90%, respectively.

The rest of the paper is organized as follows. The second part describes the evolution of the imbalanced dataset processing method, RESNET network model, and parallel cross-convolutional neural network. The third part describes the data processing method and the network structure of RESNETCNN, afterwards, ablation experiments are conducted on the CICIDS 2017 and ISCXIDS 2012 datasets to evaluate the effectiveness of the proposed model in the fourth part. Finally, the fifth part concludes the paper.

2. Related Work

This section discusses the related work from three aspects, namely, unbalanced dataset processing methods, RESNET network models, and the evolution of parallel cross-convolutional neural networks.

2.1. Evolution of Methods for Processing Unbalanced Data Sets

The number of samples of each category in real intrusion detection datasets is often unbalanced, and the methods to deal with this problem mainly include feature selection-based, and resampling-based methods. The feature selection-based approach [35] first identifies highly relevant features from the source and target domains, then removes irrelevant or redundant features, and finally applies the highly relevant features to the target problem. This method has two problems: first, it does not consider the correlation between features and data; second, the selected feature data usually tend to be in the majority class with large amount of data. The resampling-based methods change the data distribution through specific data sampling to equalize the data distribution of different classes in the dataset. The common methods are divided into two categories: random downsampling and random oversampling. Random oversampling is performed by randomly sampling a small number of samples in the dataset, and then adding the sampled samples to the original dataset. Random oversampling can restore the data of the real scene and improve the detection rate of the small number of data in the unbalanced abnormal traffic.

2.2. Evolution of the RESNET Model

ImageNet classification with deep convolutional neural network(Alexnet) [14] unveiled the dominance of neural networks in computer vision by incorporating a variant of the neural network model in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. Visual geometry group(VGG) [27] investigated the effect of convolutional network's depth on the accuracy of large-scale image recognition, where (3×3) Convolutional filter architecture was used to evaluate the depth of the network. And the results showed that pushing the network's depth to 16-19 weight layers achieves a significant improvement over existing technology configurations. Going deeper with convolutions (InceptionV1) [29] is a 22-layer deep network, which won the first place in the 2014 ILSVRC challenge. It increases the depth and width of the network while keeping the computational budget the same. Its disadvantage is that the network is difficult to change, and the cost will increase four times if the number of filter sets is doubled.

Batch normalization: accelerating deep network training by reducing internal covariate shift(InceptionV2) [12] improves InceptionV1 with a Batch Normalization(BN) layer, achieving a top-5:4.9% result in the ILSVRC competition, but also increases the weight consumption by 25% and the computational consumption by 30%.Rethinking the inception architecture for computer vision (InceptionV3) [30] improves InceptionV2 by reducing the initial module and parallelizing the network structure. RESNET [10] improves InceptionV3 by introducing a residual structure that smoothly propagates the gradient from backward to forward, which extends the RESNET structure to thousands of layers and further alleviates the gradient disappearance problem in deep neural networks.

2.3. Evolution of Parallel Cross Convolutional Neural Networks

There is an imbalance of different categories of data in intrusion detection dataset. In order to improve the detection accuracy of few categories of data, many researchers have proposed parallel cross-convolutional neural network. It draws temporal, spatial, and semantic features of anomalous traffic dataset through feature fusion of two or three layers of different convolutional neural networks, greatly improving the classification accuracy of anomalous traffic dataset.

[39] proposed parallel cross-convolutional neural network based on deep hierarchical network [40],called PCCN. This network model fused fully convolutional networks for semantic segmentation(FCN) [25] and convolutional neural network (CNN). Although its overall accuracy reached 0.9991 at CICIDS 2017, there are problems of excessive amount of parameters, long training time ,and low classification accuracy. An anomaly network traffic detection model integrating temporal and spatial features(ITSN) [22] and an efficient hybrid parallel deep learning model(HPM) [4] are both parallel deep learning models that fuse long short-term memory(LSTM) [11] and CNN,which have the advantage of introducing LSTM to extract spatial features.Compared with PCCN, the classification accuracy of minority classes is improved, but the classification accuracy of GlodenEye, Hulk, slowhttp and slowloris in CICIDS 2017 dataset is lower.A network abnormal detection method(PCCS) [34] combines single-stage headless face detector algorithms(SSH) and parallel crossover neural network, and consists of two parallel convolutional network layers. It has the advantage that the overall accuracy is improved compared to ITSN, reaching 0.9996, but the classification accuracy is lower on GlodenEye, Hulk, slowhttp, and slowloris in the CICIDS 2017 dataset.

RESNETCNN adopts a parallel crossover network structure with RESNET at the top and CNN at the bottom. RESNET absorbs semantic features and CNN absorbs high-resolution features.After three stages of feature fusion, the overall accuracy reaches 99.96% at CICIDS 2017, and the classification accuracy of GlodenEye, Hulk, slowhttp, and slowloris in the dataset is greatly improved.Compared with the traditional work, RESNETCNN has fewer parameters, high classification accuracy and fast speed, which is suitable for the classification of large anomalous traffic datasets.

3. Data Pre-processing

This section mainly discusses the algorithm of data preprocessing and verifies the validity of random oversampling method.

The ISCXIDS 2012 dataset suffers from data imbalance in different categories, and the classification accuracy of different abnormal traffic categories varies greatly. Thus, a random oversampling method is used with the following algorithm.

First, the data classified by the confusion Matrix is marked as X_{ij} ($i=1 \dots m, j=1 \dots n$).

Step1 calculates the ratio of the number of incorrect classifications to the number of correct classifications for each category in the confusion Matrix (misclassification ratio).

$$ratio = \frac{\sum_{j=1}^n X_{ij}(j \neq i)}{X_{ii}} \quad (1)$$

Observe whether the proportion of misclassified data in most minority classes is less than the proportion of misclassified data in most classes, and if so, proceed to step2.

Step2 random oversampling, that is, the minority class data in the dataset is amplified according to the maximum data volume of the majority class.

(1) Integrates all categories of data in anomalous traffic datasets.
 (2) Classifies large data sets into m classes based on labeled features and store them in a collection of lists.

(3) lists stores the m -class data set, expands the data amount of minority class to the maximum data amount Max of majority class, and then stores it in the excell table.

The pseudo code for the oversampling algorithm is shown in Algorithm1. Description of the argument in Algorithm 1 is shown in table 1.

Algorithm 1 Resampling of a Small Number of Data Sets

Input: all data[];

Output: Expanded data;

```

1: temp_list=[]; # Store the final processed data
2: label_list=[]; # Store raw data label set
3: MAX;
4: length;# The length of raw data
5: for  $i$  in range(0,  $m$ ) do
5:     #Generate  $m$  empty list sets to temporarily store the generated  $m$  feature sets
    temp_list.append([]);
6: end for
7: for  $i$  in range(0, length) do
7:     #Get the index number of the tag
    data_index = label_list.index(all data[i]);
    temp_lists[data_index].append(all data[i]);
8: end for
9: #Expand the small number of data sets to at least Max, and then store them.
10: for  $i$  in range(0, temp_lists.length) do
11:     while  $len(temp\_lists[i]) < MAX$  do
11:         temp_list[i].expend(temp_list[i])
12:     end while
13: end for
14: #Save the descended data to the specified csv file

```

Table 1. Description of the argument in Algorithm 1

Argument	Description
all Data[]	Enter the csv file data in the ISCXIDS 2012 data set
temp_list[]	Generate m empty list sets to temporarily store the generated m feature sets
Label_list	Raw data label set
Max	Maximum data size for most classes
Length	The number of rows of all Data[]

We found that in the intrusion detection dataset ISCXIDS 2012, the misclassification ratio of most minority data is lower than that of most majority data, namely, the dataset is unbalanced. So we conducted random oversampling on ISCXIDS 2012. After random oversampling, we trained on the RESNETCNN3 model, and the accuracy increased by 0.0001. The results are shown in Fig. 1.

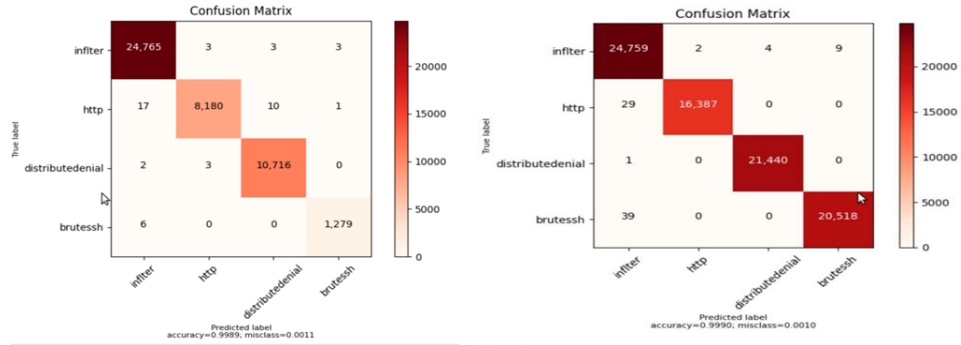


Fig. 1. (1) Data Set before Random Oversampling (2)Data Set after Random Oversampling

4. System Model

This section details the network structures of RESNETCNN and three improved versions of RESNETCNN.

4.1. Network Structure of RESNETCNN

As shown in Fig. 2, the basic idea of RESNETCNN is mainly inspired by the residual structure RESNET, and the model is divided into two layers, top branch and bottom branch. Top branch adopts the first three layers of RESNET version 18, while bottom branch adopts the traditional CNN structure, which consists of three convolutional pooling layers. In order to improve the detection accuracy of a few classes, top branch and

bottom branch are fused into three stages: the first two stages are add feature fusion and the third stage is concatenate feature fusion.

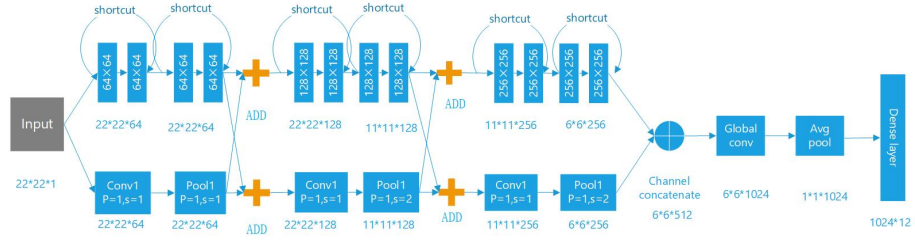


Fig. 2. Network Structure of RESNETCNN

1) TOP BRANCH

RESNET structure is applied to RESNETCNN upper layer. To solve the problem of network degradation, Kai-Ming He proposed the residual learning. Generally speaking, the deeper the network layer, the stronger the network fitting ability, and the smaller the network training error should be. But in fact, the deeper the network layer, the easier it is to overfit, and the network optimization is more difficult. RESNET takes a cue from LSTM, introduces a gating unit, and adds a computational path, shortcut mapping, to the traditional forward propagation. Shortcut mapping is beneficial to gradient propagation.

$$\frac{\partial H}{\partial X} = \frac{\partial F}{\partial X} + \frac{\partial X}{\partial X} = \frac{\partial F}{\partial X} + 1 \tag{2}$$

The constant mapping of Eq. (2) allows the gradient to propagate unimpeded from back to front, which enables the RESNET structure to expand to thousands of layers (1202 layers).

As shown in Fig. 3, the residual structure is stacked in two ways, Basic block and Bottleneck block. Basic block uses two 3 × 3 convolutional stacking mode. First input parameter x, then perform 3 × 3 convolution, relu function, and 3 × 3 convolution to get F(x), finally calculate H(x)=F(x)+x. Fitting F(x) makes the convolutional block easier to learn constant mapping. When F(x)=0, H(x)=x, in which case the precision of the deep network is higher than that of the shallow network and the network achieves constant mapping.

$$F(x) = W_2 \times Relu(W_1 \times x) \tag{3}$$

$$H(x) = F(x) + x = W_2 \times Relu(W_1 \times x) + x \tag{4}$$

Unlike the former, the number of channels in multiple network layers is like a bottleneck. The input channels change from large to small, and then from small to large. Bottleneck block uses 1 × 1 convolution kernel. The first 1 × 1 decreases the number of channels by 1/4 and the second 1 × 1 increases the number of channels by 4 times, which is more conducive to extracting advanced features by first descending and then ascending, and at the same time reduces computation and saves training time.

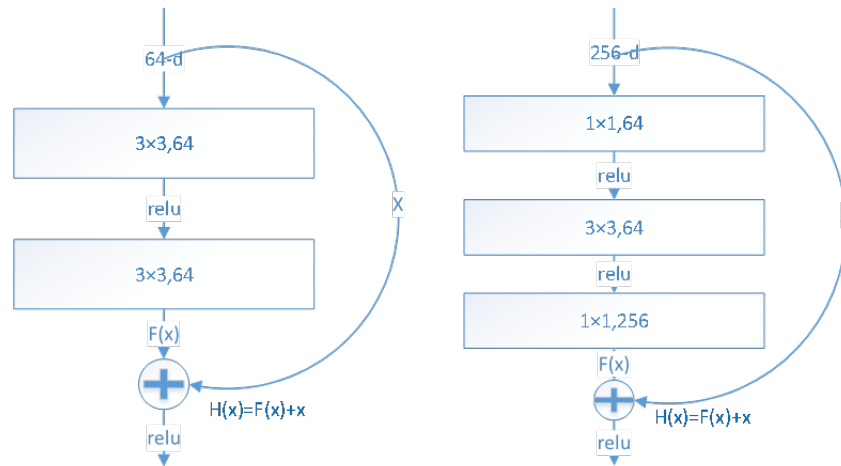


Fig. 3. Basic Block and Bottleneck Block

The upper layer is structured with RESNET18, which consists of 6 Basic blocks with 3×3 convolutional kernels, to learn the semantic features of unbalanced anomalous traffic.

2) BOTTOM BRANCH

The lower layer of RESNETCNN adopts CNN structure. CNN is a mathematical model that imitates the structure and function of biological neural network. CNN can be divided into three categories: convolutional layer, pooling layer, and fully-connected layer. The main role of convolutional layer and pooling layer are feature extraction and downsampling respectively. The pooling layer retains the most significant features and discards other useless information to reduce the operation. The introduction of pooling layer also ensures the translation invariance, i.e., the same image after flipping and deforming can get similar results. The fully-connected layer is mainly responsible for classifying the features derived from the previous convolution and pooling layers. After each neuron feedback a different weight, it will adjust the weight and network to get the classification results.

The lower layer of RESNETCNN consists of three sets of convolutional pooling layers stacked, which are used to extract high-level traffic features of unbalanced anomalous traffic.

3) FEATURE CROSS FUSION

Feature fusion is an important tool to improve classification performance. Low-level features have higher resolution, and contain more location and detail information. But they are less semantic and more noisy because they go through fewer convolutional layers. High layer features have stronger semantic information, but have very low resolution and poor perception of details. How to fuse the two is the key to improve the classification model. According to the order of fusion and prediction, they are divided into early fusion and late fusion. The former first fuses the features of multiple layers first, and then trains the predictor on the fused features. Two classical fusion methods are 1) concatenate: series

feature fusion, where two features are directly connected. If the dimension of two input features x and y are p and q , and the dimension of output feature z is $p+q$; 2) add:parallel strategy fusion, combining these two feature vectors into a complex vector. After features x and y are input, $z = x + i \times y$ will be output, where i is an imaginary unit. Late fusion improves detection performance by combining detection results from different layers in two ways: 1) multi-scale features are first predicted separately, and then the results are fused. 2) multi-scale features are first pyramid fused, and then the results are predicted. RESNETCNN uses concatenate and add of the early fusion.

In the first stage, the upper layer passes through two Basic blocks with 64 channels and 3×3 convolutional kernels in RESNET18, and the lower layer passes through two convolutional pooling layers with 64 channels, 3×3 convolutional kernels, padding=1 and stride=1 in CNN. Add fusion is used for feature fusion in the first and second stages. Compared with the first stage, in the second stage, the stride of the second block of the upper layer is changed to 2, the stride of the second pooling layer of the lower layer is changed to 2, and the size of the convolutional kernel is reduced by 1/2. The purpose of this approach is to extract richer semantic information through downsampling. The third stage of feature fusion uses concatenate fusion, where the number of channels is expanded twice. It realizes the complementary advantages between different features, and is more conducive to fully learning the intrinsic features of traffic data, thus weakening the impact of data imbalance on traffic data feature learning. In the fourth stage, after full convolution, average pooling and fc layer, the information extracted from the feature layer is classified, and the final accuracy reaches 99.96% on the CICIDS 2017 dataset.

4.2. Three Improved Versions of RESNETCNN

Version 1 is RESNETCNN1 (MINIRESET50 Cross Convolutional Neural Network), as shown in Fig. 4, which contains top branch and bottom branch. Top branch is the RESNET50 structure, consisting of six Bottleneck blocks, and bottom branch is the CNN structure, consisting of six convolutional and pooling blocks. The output of the two branches is fused with concatenate channels, and finally the classification features are output through the fully connected layer. Version 2 is RESNETCNN2 (RESNET50 Cross Convolutional Neural Network), as shown in Fig. 5. The top branch adopts RESNET50 structure and the bottom branch is CNN structure. Different from version 1, Version 2 performs feature fusion in three stages. The top two Bottleneck block and the lower two CNN convolutional pooling layers in the first and second stages perform ADD fusion and extract semantic features of the data. In the third stage, two Bottleneck blocks in the upper layer and two CNN convolutional pooling layers in the lower layer perform concatenate channel fusion to improve the detection performance of few classes of data in imbalanced datasets. Version 3 is RESNETCNN3 (RESNET Cross Convolutional Neural Network), as shown in Fig. 6, which differs from versions 1 and 2 in that (1) the top branch adopts the structure of RESNEXT [37]. RESNEXT introduces a grouped convolution with a grouping number of 32, where each block is divided into 32 groups and the convolutional kernel size is 4; (2) feature fusion is performed in two stages, each stage the upper three Bottleneck blocks and the lower two CNN convolutional pooling blocks are fused. The upper and lower layers of the first stage are add fused to extract low-level traffic features. In the second stage, the upper and lower layers perform concatenate fusion to extract high-level traffic features, and finally the final result is obtained through the classification layer.

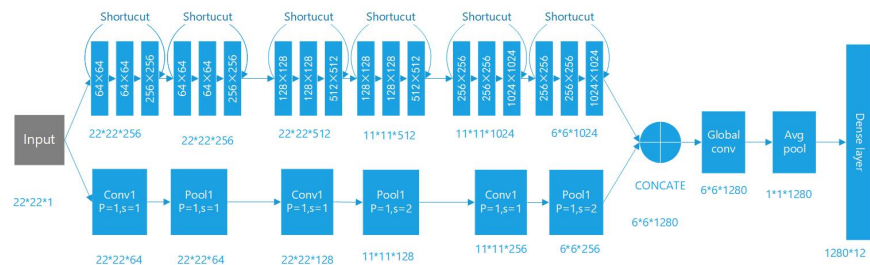


Fig. 4. Network Structure of RESNETCNN1

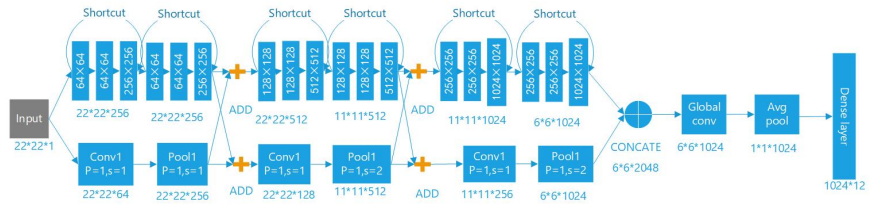


Fig. 5. Network Structure of RESNETCNN2

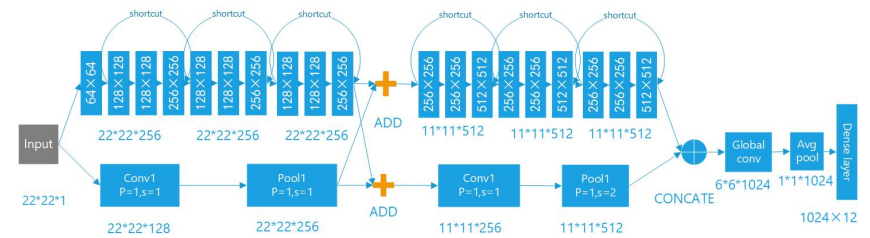


Fig. 6. Network Structure of RESNETCNN3

5. Experimental Results and Analysis

This section first describes the environment required for the experiments, including the hardware environment, the software environment, and the datasets. In addition, the evaluation metrics used in the experiments and the configuration of parameters during model training are presented. In the last part of this section, the content of the experiment is introduced in detail and the experimental results are analyzed.

5.1. Experimental Environment

The experimental environment is shown in Table 2.

Table 2. Experiment Environment

Equipment	Example
CPU	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
GPU	Quadro P4000
Memory	64GB
Hard Disk	2T
OS	Ubuntu 16.04
Compile software	PyCharm 2021.2.3
Python	3.8
Database	ISCXIDS 2012,CICIDS2017

5.2. Hyperparameter Setting

In RESNETCNN, we fix the size of the convolutional kernel to 3×3 . The model uses a batch size of 256 during training, where the momentum is fixed at 0.9 and the weight decay is set to 1×10^{-4} to prevent overfitting and the model falling into local optima. Also, we use a cross-entropy loss function to continuously optimize the model parameters and an Adam optimizer to accelerate the convergence of the network. Setting the learning rate too large or too small can affect the convergence of the model and cause the model to miss the optimal point. Therefore, a total of 10 iterations are designed in this paper. The learning rate settings for each iteration are shown in Table 3.

Table 3. Learning Rate Setting

Epoch	0	1	2	3	4	5	6	7	8	9
Learning Rate	0.0001	0.0001	0.0001	0.0001	0.0001	0.00002	0.00002	0.00002	0.000004	0.000004

5.3. Evaluation Indicators

The horizontal axis in the confusion matrix [33] is a count of the number of categories predicted by the model, and the vertical axis is a count of the number of true labels of the data. The diagonal line, represents the number of model predictions that agree with the data labels, so the sum of the diagonals divided by the total number of test sets is the accuracy rate. The larger the number on the diagonal, the better, and the darker the color in the visualization results, indicating the more accurate the model's prediction in that category.

True Positive (True, TP): predicts the positive class as the number of positive classes, True Negative (True Negative, TN): predicts the negative class as the number of negative classes, False Positive (False Positive, FP): predicts the negative class as the number of positive classes, and False Negative (False Negative, FN): predicts the positive class as the number of negative classes.

1. Precision

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

The fraction of predicted positive cases that are determined to be positive, as a percentage of all predicted positive cases.

2. Recall

$$Recallrate = \frac{TP}{TP + FN} \quad (6)$$

The fraction of cases predicted to be positive and that are indeed positive, as a percentage of all classes that are indeed positive.

3. Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

All positive and negative cases with correct predictions, as a percentage of all samples.

4. F1_Score

$$F1_score = 2 \times \left(\frac{Precision \times Recall}{precision + Recall} \right) \quad (8)$$

F1_Score, also known as the balanced F_score, is the summed average of Precision and Recall, which combines the results of Precision and Recall. F1_Score ranges from 0 to 1, with 1 and 0 representing the best and the worst output of the model respectively. The two metrics, Precision and Recall, are commonly used to evaluate the analytical effectiveness of two classification models. However, when these two metrics are in conflict, it is difficult to compare between models. For example, we have two models A and B. Model A has a higher recall than model B, but model B has a higher precision than model A. This is where F1_Score is used to judge the overall performance of the two models.

5.4. Experimental Analysis

To further explore the detection performance of RESNETCNN for unbalanced data streams in complex environments, we conducted ablation experiments on the CICIDS 2017 dataset. Fig. 7 and Table4-8 show the experimental results, where the data of PCSS are cited from

Tables 3, 4, 5, 6, and 7 in PCSS . From Fig. 7, it can be seen that the RESNETCNN series network model proposed in this paper outperforms the PCCN, PCSS, and ITSN models in the four evaluation indexes of average recall, average F1_score, average precision, and average accuracy. It shows that the PCSS and RESNETCNN are the same in average accuracy, but the average precision of RESNETCNN is lower than that of PCSS. In comparison, the three improved versions of RESNETCNN in the experiments are better than PCSS. Compared with RESNETCNN2 and RESNETCNN3 , RESNETCNN1 shows an increasing trend in average accuracy, average precision, and average F1_score, but the average recall of RESNETCNN1 is better than that of PCSS. RESNETCNN1 outperforms the other two models in average recall, indicating that multiple feature fusion will reduce the callback rate while improving the accuracy.

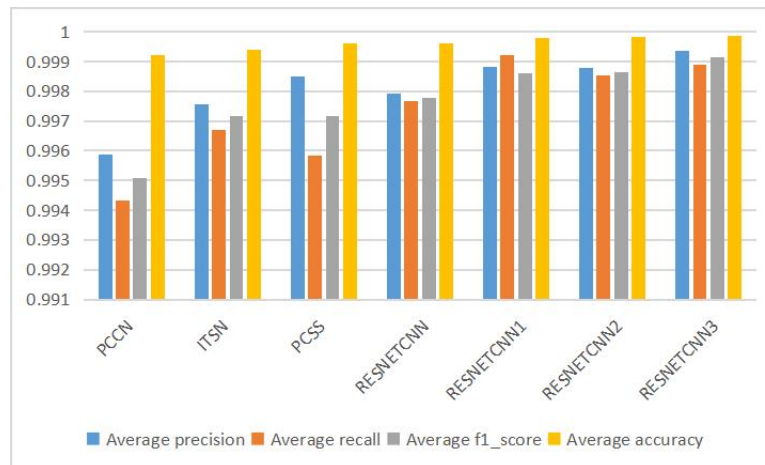


Fig. 7. Experimental Comparison of Various Abnormal Traffic Detection Algorithms

In Table 4, RESNETCNN3 has the highest overall accuracy, reaching a classification accuracy of 0.999876. Meanwhile, it can be seen that the overall accuracies of RESNETCNN, RESNETCNN1, RENETCNN2, and RENETCNN3 show a stepwise trend of increasing, indicating that the three improved versions proposed in this paper in the network model with RESNET50 to replace RESNET18 is effective. Tables 5 to 8 show the classification precision, F1_score, recall, and False-positive rate of 12 types of anomalous traffic data in the CICIDS 2017 dataset for PCCN, ITSN , PCCS, RESNETCNN, RESNETCNN1 , RESNETCNN2, and RESNETCNN3 on eight network models. In Table 6 and 7, except for the third category of anomalous traffic with recall and f1_score that reaches the optimum on PCSS, the other categories of anomalous traffic all achieve the optimum on RESNETCNN3. In Table 5 and 8, RESNETCNN3 achieves the optimum on precision, F1_score, recall, and False_positive rate. In order to prevent overfitting, the RESNETCNN3 model is validated on the ISCXIDS 2012 dataset, with an overall accuracy of 99.9%.

To understand the errors in the experimental results of the RESNETCNN family of models, the heat map shown in Figure 8 is generated according to the results of the experiments performed on the RESNETCNN3 model. The diagonal numbers represent the number of correct predictions and the remaining numbers are the number of incorrect predictions. It can be seen that the RESNETCNN3 network model proposed in this paper achieves a high detection success rate for monitoring twelve types of abnormal traffic in the CICIDS 2017 dataset, which proves that our proposed method is effective.

True label \ Predicted label	botnet	DDoS	GlodenEye	Hulk	slowhttp	slowloris	Ftppatator	heartbleed	infiltration	portscan	sshpatator	webattack
botnet	415	0	0	0	0	0	0	0	0	0	0	0
DDoS	0	52246	0	0	0	0	0	0	0	0	0	0
GlodenEye	0	0	4103	3	3	0	0	0	0	0	0	0
Hulk	0	0	2	94929	1	0	0	0	0	0	0	0
slowhttp	0	0	9	0	1338	9	0	0	0	2	0	0
slowloris	0	0	2	0	4	2102	0	0	0	0	0	0
Ftppatator	0	0	0	0	0	0	3989	0	0	0	0	0
heartbleed	0	0	0	0	0	0	0	1972	0	0	0	0
infiltration	0	0	0	0	0	0	0	0	1067	0	0	0
portscan	0	0	1	0	1	0	0	0	0	53925	0	1
sshpatator	0	0	0	0	0	0	0	0	0	0	5509	0
webattack	0	0	0	0	0	0	0	0	0	0	0	2109

Fig. 8. Confusion Matrix of RESNETCNN3

6. Conclusion

In this paper, an intrusion detection model (RESNETCCN) is proposed that fuses RESNET and parallel cross-convolutional neural networks, and three improved versions of the RESNETCNN network model are designed. In addition, a data oversampling method is introduced to improve the detection accuracy of imbalance data in the ISCXIDS 2012 dataset. The experimental results show that the four RESNETCNN network models proposed in this paper can effectively handle the unbalanced abnormal traffic data and provide an effective solution for network security intrusion detection systems.

Although the RESNETCNN network model achieves extremely high classification accuracy, the current network environment is complex and changing, resulting in the RESNETCNN model based on closed-set protocols cannot meet the new network anomaly traffic detection requirements. In our future work, we will introduce more new ideas such

as blockchain cryptography [8], [18], [9], [19], [16], alliance chain [36], [7], [20], visual Q&A [5], [28], transformer [21], panoramic image [17], reinforcement learning [3], internet of things [23], [24], shared data [6] in our model. We will continue to explore network intrusion detection methods in more areas such as unsupervised and semi-supervised [2] areas for network anomalous traffic data detection. In addition, we also try to introduce new evaluation metrics and establish systematic evaluation methods of intrusion detection.

References

1. Bang, J.h., Cho, Y.j., Kang, K.: Anomaly detection of network-initiated lte signaling traffic in wireless sensor and actuator networks based on a hidden semi-markov model 65, 108–120 (2017)
2. Cai, S., Han, D., Li, D.: A feedback semi-supervised learning with meta-gradient for intrusion detection. *IEEE Systems Journal* (2022)
3. Cai, S., Han, D., Li, D., Zheng, Z., Crespi, N.: An reinforcement learning-based speech censorship chatbot system. *The Journal of Supercomputing* 78(6), 8751–8773 (2022)
4. Cai, S., Han, D., Yin, X., Li, D., Chang, C.C.: A hybrid parallel deep learning model for efficient intrusion detection based on metric learning. *Connection Science* 34(1), 551–577 (2022)
5. Chen, C., Han, D., Chang, C.C.: Caan: Context-aware attention network for visual question answering. *Pattern Recognition* 132, 108980 (2022)
6. Cui, M., Han, D., Wang, J., Li, K.C., Chang, C.C.: Arfv: An efficient shared data auditing scheme supporting revocation for fog-assisted vehicular ad-hoc networks. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY* 69(12), 15815–15827 (2020)
7. Gao, N., Han, D., Weng, T.H., Xia, B., Li, D., Castiglione, A., Li, K.C.: Modeling and analysis of port supply chain system based on fabric blockchain. *COMPUTERS & INDUSTRIAL ENGINEERING* 172(A) (2022)
8. Han, D., Pan, N., Li, K.C.: 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 (2022)
9. Han, D., Zhu, Y., Li, D., Liang, W., Soury, A., Li, K.C.: A blockchain-based auditable access control system for private data in service-centric iot environments. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS* 18(5), 3530–3540 (2022)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* 9(8), 1735–1780 (1997)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International conference on machine learning*. pp. 448–456. PMLR (2015)
13. Ji, S.: *Research on network traffic intrusion detection based on deep learning* (2020)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60(6), 84–90 (2017)
15. Li, B., Zhang, S., Li, K.: Towards a multi-layers anomaly detection framework for analyzing network traffic 29 (2017)
16. Li, D., Han, D., Weng, T.H., Zheng, Z., Li, H., Liu, H., Castiglione, A., Li, K.C.: Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. *Soft Computing* 26(9), 4423–4440 (2022)
17. Li, D., Han, D., Zhang, X., Zhang, L.: Panoramic image mosaic technology based on sift algorithm in power monitoring. In: *2019 6th International Conference on Systems and Informatics (ICSAI)*. pp. 1329–1333. IEEE (2019)

18. Li, H., Han, D., Tang, M.: A privacy-preserving charging scheme for electric vehicles using blockchain and fog computing. *IEEE SYSTEMS JOURNAL* 15(3), 3189–3200 (2021)
19. Li, H., Han, D., Tang, M.: A privacy-preserving storage scheme for logistics data with assistance of blockchain. *IEEE INTERNET OF THINGS JOURNAL* 9(6), 4704–4720 (2022)
20. Li, J., Han, D., Wu, Z., Wang, J., Li, K.C., Castiglione, A.: A novel system for medical equipment supply chain traceability based on alliance chain and attribute and role access control. *Future Generation Computer Systems* 142, 195–211 (2022)
21. Li, M., Han, D., Li, D., Liu, H., Chang, C.C.: Mfvf: an anomaly traffic detection method merging feature fusion network and vision transformer architecture. *EURASIP Journal on Wireless Communications and Networking* 2022(1), 1–22 (2022)
22. Li, M., Han, D., Yin, X., Liu, H., Li, D.: Design and implementation of an anomaly network traffic detection model integrating temporal and spatial features. *Security and Communication Networks* 2021 (2021)
23. Liu, H., Han, D., Cui, M., Li, K.C., Souri, A., Shojafar, M.: Idenmultisig: Identity-based decentralized multi-signature in internet of things. *IEEE Transactions on Computational Social Systems* pp. 1–11 (2023)
24. Liu, H., Han, D., Li, D.: Fabric-iot: A blockchain-based access control system in iot. *IEEE Access* 8, 18207–18218 (2020)
25. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 3431–3440 (2015)
26. Nezhad, S.M.T., Nazari, M., Gharavol, E.A.: A novel dos and ddos attacks detection algorithm using arima time series model and chaotic system in computer networks 20(4), 700–703 (2016)
27. Sercu, T., Puhersch, C., Kingsbury, B., LeCun, Y.: Very deep multilingual convolutional neural networks for lvcsr. In: *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. pp. 4955–4959 (2016)
28. Shen, X., Han, D., Guo, Z., Chen, C., Hua, J., Luo, G.: Local self-attention in transformer for visual question answering. *APPLIED INTELLIGENCE* (2022)
29. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1–9 (2015)
30. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2818–2826 (2016)
31. Tian, Q., Han, D., Li, K.C., Liu, X., Duan, L., Castiglione, A.: An intrusion detection approach based on improved deep belief network. *APPLIED INTELLIGENCE* 50(10), 3162–3178 (2020)
32. Tian, Q., Han, D., Li, K.C., Liu, X., Duan, L., Castiglione, A.: An intrusion detection approach based on improved deep belief network. *Applied Intelligence* 50(10), 3162–3178 (2020)
33. Visa, S., Ramsay, B., Ralescu, A.L., Van Der Knaap, E.: Confusion matrix-based feature selection. *MAICS* 710, 120–127 (2011)
34. Wang, Z., Han, D., Li, M., Liu, H., Cui, M.: The abnormal traffic detection scheme based on pca and ssh. *Connection Science* 34(1), 1201–1220 (2022)
35. Wasikowski, M., Chen, X.w.: Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on knowledge and data engineering* 22(10), 1388–1400 (2009)
36. Xiao, T., Han, D., He, J., Li, K.C., de Mello, R.F.: Multi-keyword ranked search based on mapping set matching in cloud ciphertext storage system. *CONNECTION SCIENCE* 33(1), 95–112 (2021)
37. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1492–1500 (2017)

38. Yang, C.: Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment 22(4), S8309–S8317 (2019)
39. Zhang, Y., Chen, X., Guo, D., Song, M., Teng, Y., Wang, X.: Pccn: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows. *IEEE Access* 7, 119904–119916 (2019)
40. Zhang, Y., Chen, X., Jin, L., Wang, X., Guo, D.: Network intrusion detection: Based on deep hierarchical network and original flow data. *IEEE Access* 7, 37004–37016 (2019)
41. Zhong, Y., Chen, W., Wang, Z., Chen, Y., Wang, K., Li, Y., Yin, X., Shi, X., Yang, J., Li, K.: Helad: A novel network anomaly detection model based on heterogeneous ensemble learning (2020)
42. Zhou, Z.H.: *Machine learning*. Prentice Hall, Springer Nature (2021)

Yimin Li received the B.S degree from Hubei University of Economics, Wuhan, China. She is currently working toward the M.S. degree with Shanghai Maritime University, Shanghai, China. Her main research interests include intrusion detection, cloud computing security, machine learning, and deep learning.

Dezhi Han received the B.S. degree from Hefei University of Technology, Hefei, China, the M.S. and Ph.D. degrees from Huazhong University of Science and Technology, Wuhan, China. He is currently a Professor of computer science and engineering with Shanghai Maritime University, Shanghai, China. His specific interests include storage architecture, blockchain technology, cloud computing security, and cloud storage security technology.

Mingming Cui received the B.S. degree in Computer Science and Technology from the Anhui University of Finance and Economics, Bengbu, China. She is currently pursuing the Ph.D. degree in Information management and information systems from Shanghai Maritime University, Pudong, China. She is currently a Visiting Ph.D. student in the Nanyang Technological University, Singapore. Her current research interests include cryptology, blockchain, data privacy protection, network security, VANETS security, and Internet of things.

Yuan Fan received the B.S. degree from Nanjing University of Posts and telecommunications, Nanjing, China, and the M.S. degree from San Jose State University, California, the U.S.A. He is the Chairman of DBAPP Security Co., Ltd. His specific interests include cybersecurity and data security. In recognition of his great contribution, he has been granted the special allowance of the State Council, selected into the National Million Talent Project, he is also a member of the 10th National Committee of China Association for Science and Technology.

Yachao Zhou received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, the M.S. degree from Tsinghua University, Beijing, China, and Ph.D. degree from Dublin City University, Dublin, Ireland. She is the chief scientist of DBAPP Security Co., Ltd. Shanghai headquarters. Her specific interests include cloud computing, deep packet inspection and IoT security.

Received: November 24, 2022; Accepted: February 10, 2023.

Logical dependencies: extraction from the versioning system and usage in key classes detection

Adelina Diana Stana and Ioana Şora

Politehnica University, Piaţa Victoriei Nr. 2,
300006 Timişoara, jud. Timiş, România
stana.adelina.diana@gmail.com
ioana.sora@cs.upt.ro

Abstract. The version control system of every software product can provide important information about how the system is connected. In this study, we first propose a language-independent method to collect and filter dependencies from the version control, and second, we use the results obtained in the first step to identify key classes from three software systems. To identify the key classes, we are using the dependencies extracted from the version control system together with dependencies from the source code, and also separate. Based on the results obtained we can say that compared with the results obtained by using only dependencies extracted from code, the mix between both types of dependencies provides small improvements. And, by using only dependencies from the version control system, we obtained results that did not surpass the results previously mentioned, but are still acceptable. We still consider this an important result because this might open an important opportunity for software systems that use dynamically typed languages such as JavaScript, Objective-C, Python, and Ruby, or systems that use multiple languages. These types of systems, for which the code dependencies are harder to obtain, can use the dependencies extracted from the version control to gain better knowledge about the system.

Keywords: logical dependencies, logical coupling; mining software repositories, versioning system, key classes, co-changing entities, software evolution.

1. Introduction

The version control (also known as source control) system that tracks changes in source code during software development can provide useful information about the system's details. The usage of information extracted from the version control system is not new. Previous works have used version control information to detect design issues [27], predict fault incidence among modules [13], [3], reconstructing software development methods [14] or guide software changes [9], [2]. In software engineering literature, concepts like evolutionary coupling, evolutionary dependencies, logical dependencies, or logical coupling refer to the same sort of relationship among software entities. That relationship is extracted from the version control system and can mean that the entities from the source code files change together, evolve together, and might depend on one another. Studies show that dependency relationships found in the source code overlap only in a small percentage with dependency relationships found in the version control system, and suggest

that these two types of relationships can be used together [15], [1]. But, in practice, dependencies extracted from the version management system are rarely used because of the size of the information extracted [19]. A relatively small source code repository with roundabout one thousand commits can lead to millions of connections. In this paper, by applying a set of filters with different thresholds to the information extracted, we intend to speed up the processing time, reduce the size of connections extracted from the version control and increase the confidence that the connections obtained might be related. To validate the results obtained, and to see if the filtering methods had or had not a favorable effect on the final result, we want to identify the key classes of different systems. The identification of key classes has been previously performed by using structural dependencies, so we intend to use the results obtained together with structural dependencies, and also separate, and see how the final results fluctuate.

In this work we perform our analysis on three open source projects: Ant, Tomcat Catalina, and Hibernate. And we answer the following research questions:

RQ1: Can logical dependencies combined with structural dependencies enhance the results obtained by using only structural dependencies in key class detection? Since previous researches and our studies show that logical dependencies overlap only in a small percentage with structural dependencies, we intend to combine both types of dependencies and provide them as input for key classes identification tools.

RQ2: Can logical dependencies provide good results if they are used instead of structural dependencies in key class detection? In this research question, we want to focus more on the use of logical dependencies as stand-alone dependencies. Even though logical dependencies are not the same as structural dependencies, they can provide enough information about the system to be successfully used as input for tools like key classes detection tools.

RQ3: Does the connection strength filter has a favorable impact on the detection of key classes? In this paper, we use a new type of filter: the connection strength filter. This filter, together with the commit size filter, will be used to filter co-changes into logical dependencies. Now, the question is if this new filter will indeed do what we expect and provide better results.

The paper is organized as follows: Section 2 introduces the concepts of logical dependencies and the methods of obtaining them. Section 3 introduces the concept of key classes and the metrics used for results evaluation. The new approach of using logical dependencies to detect key classes can be found in section 4. Section 5 defines the data set used and presents the new results obtained with the data set. In section 6 we present the threats to the validity of the results presented in this paper. And finally, section 7 discusses the conclusions based on the results obtained.

2. The concept of logical dependencies

This section presents the definition of logical dependencies, previous works involving logical dependencies, and our contribution to this topic. In subsection 2.1, we define what logical dependencies are and the previous research involving them. In subsection 2.2, we present our approach for identifying logical dependencies and the work we have done around this subject. Related to subsection 2.2, sections 2.3, and 2.4 present more details about the filtering techniques we used to identify logical dependencies.

2.1. Definition and previous related research

The concept of logical coupling (dependency) was first introduced by Gall et al. [12]. They defined the logical dependency between two software entities (classes, modules, interfaces, etc.) as the fact that the entities repeatedly change together during the historical evolution of a software system. Since then, logical dependencies have been used in multiple areas of software engineering, most commonly in fault and change prediction. Besides the studies on how logical dependencies can help gain knowledge about software systems, some studies also focused on the interplay between logical and structural dependencies. Ajiienka et al. and Olivia et al. studied the interplay between structural and logical dependencies, and they concluded that, in most cases, structural dependencies do not lead to logical dependencies [15], [16], [1]. The above affirmation is also supported by Lanza et al., who consider that logical dependencies are important because they can reveal dependencies that are not visible via code analysis [8].

In previous research, the *support* and *confidence* metrics were used to measure the strength of a logical dependency. The logical dependencies are commonly represented as directed association rules [16], [1], [27]. The association rule between A and B ($A \rightarrow B$) means that changes in entity A cause changes in entity B, where A is the antecedent, and B is the consequent of the rule. The support metric counts the number of commits in which both entities of an association rule change together. The confidence metric is the ratio between the support metric and the total number of commits in which the antecedent of the rule was involved.

By applying different thresholds to the metrics presented above, the logical dependencies to further use were selected.

2.2. Our approach for logical dependencies identification

To avoid confusion, we call *co-changing pairs* all the association rules of one system. The association rules are formed between two software entities that update together in the same commit. For example, a commit that contains seven entities will generate 21 co-changing pairs ($C_k^n = \frac{n!}{k!(n-k)!} = \frac{7!}{2!(5)!} = 21$).

The *logical dependencies* are the association rules whose metrics fulfill certain conditions. So, the logical dependencies are a subset of the co-changing pairs.

The conditions that need to be met by a co-changing pair to be considered a logical dependency are called *filters*. Like in other research regarding logical dependencies, our filters are thresholds applied to the metrics of association rules.

Previously, we tried to filter logical dependencies from co-changing pairs by applying filters like the occurrence filter and commit size filter [21], [22]. The commit size filter, presented in more detail in section 2.3, and used by other authors [1], proved to be helpful, and it will be also used for this paper. But we cannot say the same for the occurrence filter. The filter consisted of different thresholds applied to the support metric and proved to not work well for systems with few commits.

Currently, we aim to refine the filtering method with a new filter applicable to all sorts of commit history sizes. This new filter, presented in section 2.4, will be used together with the commit size filter to filter logical dependencies from co-changing pairs. The entire process of extracting co-changing pairs from the versioning system, filtering them to obtain logical dependencies, and exporting the results, is done with a tool written in Python. The workflow is presented in figure 1.

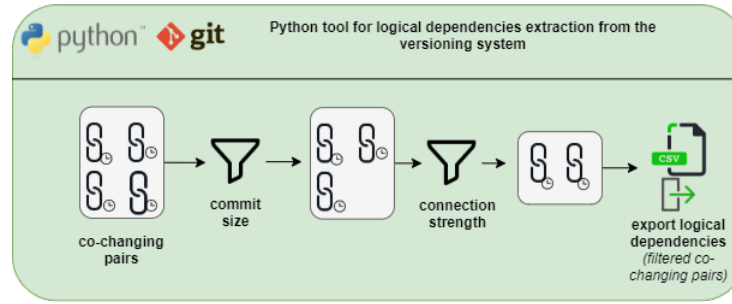


Fig. 1. Workflow for logical dependencies extraction

2.3. Logical dependencies filtering: The commit size filter

The commit size filter filters out all co-changing pairs from commits with more than 10 files changed. We consider that commits with more than 10 files changed tend to be code unrelated; we studied the commit size trend from several git open-source repositories, and we concluded that most of the commits contain less than 10 files. On average, only 10% of the total commits have more than 10 files changed.

This filter will also prevent the volume of data processed from going out of proportion. In some of the repositories studied, we found commits with more than 1000 files; these commits could generate over half a million co-changing pairs if the commit size filter is not applied [22], [21].

2.4. Logical dependencies filtering: The connection strength filter

The connection strength filter is new for our research regarding logical dependencies identification, and it is based on our experience with the occurrence filter. An important conclusion drawn from the results obtained with the occurrence filter is that setting a hard threshold for a filter is not always a good idea. A certain threshold can work well with a medium/large-sized system, but when applied to a small-sized system, it can reduce the co-changes filtered to 0. To avoid this kind of situation, we evaluated a filter that considers the system's specifications.

As we previously mentioned, a filter has two components: the metrics computed for each co-changing pair (association rule) and the threshold values. The connection strength metric derives from the support and the confidence metrics.

For an association rule (co-changing pair) formed from the antecedent A and consequent B ($A \rightarrow B$), the support count is the total number of commits in which both entities are involved,

$$\text{support}(A \rightarrow B) = \text{freq}_{\text{total commits}}(A \cup B) \quad (1)$$

and the confidence is the ratio between the support and the frequency of the antecedent of the rule.

$$\text{confidence}(A \rightarrow B) = \frac{\text{support}(A \rightarrow B)}{\text{freq}_{\text{total commits}}(A)} \quad (2)$$

The only problem with the confidence metric, as it is defined above, is that it does not include the big picture of the system. The best value for the confidence metric is 1, meaning that in all commits in which entity A is present, entity B is also there. If, for example, we have a co-changing pair $A \rightarrow B$, and A updates only once in the entire history, and in that time, updates together with B, then the confidence metric associated with the co-changing pair will be 1 (the best value possible). That is not a fair value compared with other scenarios. For example, we can have the co-changing pair $A \rightarrow B$, and A updates 100 in the entire history from which 80 times updates together with B, leading to a confidence value of 0.8. Even though in the second scenario we have a confidence value smaller than in the first scenario, the second scenario could lead to a more trustworthy connection.

Figures 2, 3 and 4 intend to offer a big picture of systems. The dots represent the maximum number of updates of one entity with another, and the black line represents the average occurrence value of the system. It can be observed that all systems have multiple entities that update only once, meaning that we might have many confidence values of 1 (the highest value possible) for entities that update only once together. We plotted only the maximum occurrences between entities to not overcrowd the plot. Even with only the maximum occurrences plotted, it can be observed that most of the points are at the bottom of the graphic. So, plotting all the points wouldn't change the overall picture of the system. The excluded points will only create a line of points even lower at the bottom of the graphic.

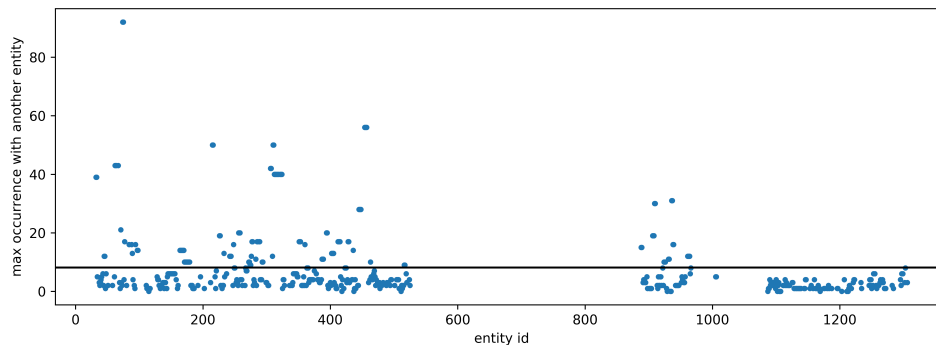


Fig. 2. Overview of the number of occurrences in Ant

To take into account the big picture of the system, we defined a new metric for a co-changing pair (association rule), called the *connection strength metric*.

The connection strength metric is computing the same ratio as the confidence metric, a.k.a the ratio between the support metric and the frequency of the antecedent. And additionally, it multiplies it with a system factor and with 100. The *system factor* calculates the ratio between the support metric and the mean value for updates. The *system mean* is the mean value of all the support values for all the association rules from the system. We multiply with 100 because we want to scale the metric's values to structural dependencies metric's values that have, in most cases, supraunitary values. And we want both metrics

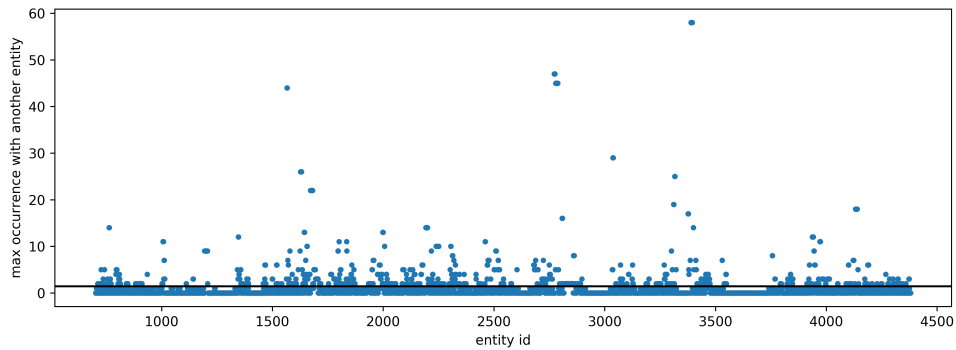


Fig. 3. Overview of the number of occurrences in Hibernate

to be comparable. The values obtained are clipped between 0 and 100, where 100 is the best value and 0 is the worst.

$$system\ factor\ for(A \rightarrow B) = \frac{support(A \rightarrow B)}{system\ mean} \tag{3}$$

$$strength(A \rightarrow B) = \frac{support(A \rightarrow B) * 100}{freq_{total\ commits}(A)} * system\ factor \tag{4}$$

By using the strength metric, if we consider again the two scenarios presented above, and a system mean value of 10, we will have the following values: for the scenario in which the entities A and B update only once, and in that one update, they update together, the strength metric value is 10. For the scenario in which entity A updates 100 times in the entire history from which 80 times updates together with B, the strength metric value is 100.

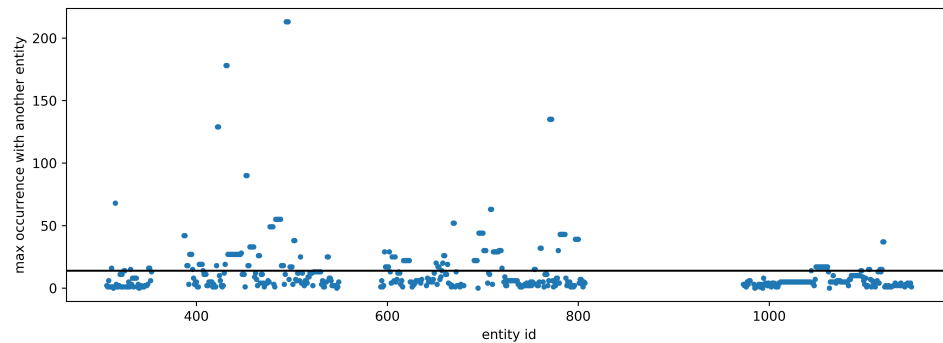


Fig. 4. Overview of the number of occurrences in Catalina

Since the values can vary from 0 to 100, the filter threshold values begin at 10 and are repeatedly incremented by 10, until 100. We do not settle for one value because we want to see how the threshold values affect the number of remaining co-changing pairs and the output of their usage.

In figure 5 we plotted for two systems (one small-sized and one medium-sized) the number of structural dependencies, co-changing pairs before filtering, and co-changing pairs after filtering. With the connection strength filter, the small-sized system didn't lose all the co-changing pairs once with the filtering. We compare the number of remaining co-changing pairs with the number of structural dependencies because, according to surveys [19], [11], the main reason why logical dependencies (filtered co-changes) are not used together with structural dependencies is their size. So, it is essential to get an overview of the comparison between the number of co-changing pairs and the number of structural dependencies at each filtering step.

We call the co-changing pairs that remain after filtering, logical dependencies. After this step, we will use the logical dependencies obtained with different threshold values and see which threshold value performs the best. Up until now, we only looked at the size of the resulting logical dependencies and decided if a filter and its threshold are good or not. Now, we can also look at the results obtained by using the logical dependencies and decide.

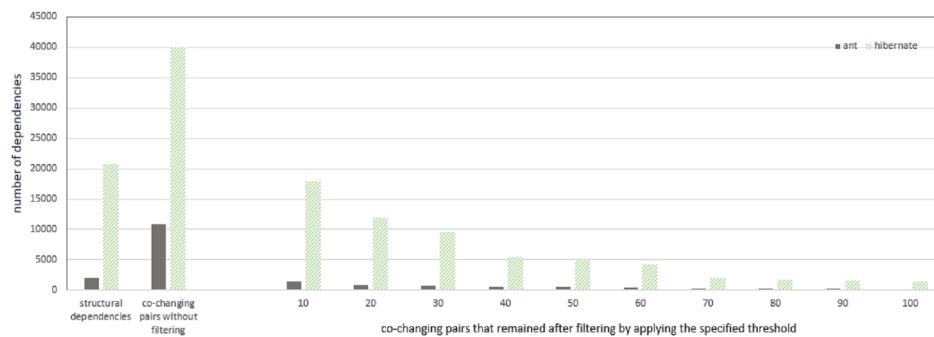


Fig. 5. Overview of the impact of connection strength filtering on the number of co-changing pairs

3. The concept of key classes

This section presents the key classes definition, previous research regarding key classes identification, and metrics used to evaluate the results obtained. In section 3.1, we present a summary of previous researchers and their approaches to key classes identification. Section 3.2 focuses on one previous research that we consider as our baseline research in key classes identification. With the results of the baseline research, we will compare our results. In section 3.3, we present the metrics used in previous research to evaluate the results obtained.

3.1. Definition and previous research

Zaidman et al. [26] was the first to introduce the concept of key classes and it refers to classes that can be found in documents written to provide an architectural overview

of the system or an introduction to the system structure. Tahvildari and Kontogiannis have a more detailed definition regarding the key classes concept: “Usually, the most important concepts of a system are implemented by very few key classes which can be characterized by the specific properties. These classes, which we refer to as key classes, manage many other classes or use them in order to implement their functionality. The key classes are tightly coupled with other parts of the system. Additionally, they tend to be rather complex, since they implement much of the legacy system’s functionality” [23].

The key class identification can be done by using different algorithms with different inputs. In the research of Osman et al., the key class identification is made by using a machine learning algorithm and class diagrams as input for the algorithm [17]. Thung et al. built on top of Osman et al.’s approach and added network metrics and optimistic classification to detect key classes [24].

Zaidman et al. used a web mining algorithm and dynamic analysis of the source code to identify the key classes [26].

3.2. Baseline approach

We use the research of I. Şora et al. [29] as a baseline for our research involving the usage of logical dependencies to find key classes.

Şora et al. used the static analysis of the source code, a page ranking algorithm and other class attributes to find key classes [5], [28], [6], [20],[29]. The page ranking algorithm is a customization of PageRank, the algorithm used to rank web pages [18], and it works based on a recommendation system. If one node has a connection with another node, then it recommends the second node. In previous research, connections are established based on structural dependencies extracted from static code analysis. If A has a structural dependency with B, then A recommends B, and also B recommends A.

The ranking algorithm ranks all the classes from the source code of the system, according to their importance. To identify the important classes from the rest, a threshold for the top classes from the top of the ranking is set. We call this TOP threshold, and its value can range from 1 to the total number of classes found in the system.

3.3. Metrics for results evaluation

To evaluate the quality of the key classes ranking algorithm and solution produced, the key classes found by the algorithm are compared with a reference solution. The reference solution is extracted from the developer documentation. The classes mentioned in the documentation are considered key classes and form the reference solution (ground truth) used for validation [25].

For the comparison between both solutions, a classification model is used. The quality of the solution produced is evaluated by using the Receiver Operating Characteristic Area Under Curve (ROC-AUC) metric, a metric that evaluates the performance of a classification model.

The ROC graph is a two-dimensional graph that has on the X-axis plotted the false positive rate and on the Y-axis the true positive rate. By plotting the true positive rate and the false positive rate at thresholds that vary between a minimum and a maximum possible value, we obtain the ROC curve. The area under the ROC curve is called Area Under the Curve (AUC).

The true positive rate of a classifier is calculated as the division between the number of true positive results identified, and all the positive results identified:

$$\text{True positive rate}(TPR) = \frac{TP}{TP + FN} \quad (5)$$

The false positive rate of a classifier is calculated as the division between the number of false positive results identified, and all the negative results identified:

$$\text{False positive rate}(FPR) = \frac{FP}{FP + TN} \quad (6)$$

The True Positives (TP) are the classes found in the reference solution and also in the top TOP ranked classes. False Positives (FP) are the classes that are not in the reference solution, but are in the TOP ranked classes. True Negatives (TN) are classes that are found neither in the reference solution nor in the TOP ranked classes. False Negatives (FN) are classes that are found in the reference solution, but are not found in the TOP ranked classes.

In related research, the ROC-AUC metric has been used to evaluate the results for finding key classes of software systems. For a classifier to be considered good, its ROC-AUC metric value should be as close to 1 as possible. When the value is 1, then the classifier is considered to be perfect. A metric value between 0.8 and 0.9 means that the classifier is excellent. Between 0.8 and 0.7 means acceptable results, and between 0.7 and 0.5 means poor results [10].

4. Key classes identification using logical dependencies

This section presents our approach for key classes identification by using logical dependencies. In section 4.1, we describe the experimental setup and how we intend to integrate logical dependencies with the baseline approach. Section 4.2 presents our investigation plan and how this plan can respond to the research questions we enunciated at the beginning of this paper.

4.1. Current approach

The baseline approach uses a tool that takes as an input the source code of the system and applies ranking strategies to rank the classes according to their importance. We modified the tool used by the baseline approach to take also the logical dependencies as input; the rest of the workflow is the same as in the baseline approach (figure 6).

Below are some of the class metrics used in the baseline approach and in our current research to rank the classes according to their importance.

The class metrics used can be separated into two categories: class connection metrics and class PageRank values. The class connection metrics are CONN-TOTAL-W, which is the total weight of all connections of the class, and CONN-TOTAL, the total number of distinct classes that a class uses or is using the class [29].

Previous research used PageRank values computed on both directed and undirected, weighted and unweighted graphs. In the current research, we use the PR, which is the PageRank value computed on the directed and unweighted graph, the PR-U, which is the value computed on the undirected and unweighted graph, and the PR-U2-W, the value computed on the weighted graph with back-recommendations [5], [28], [29], [20].

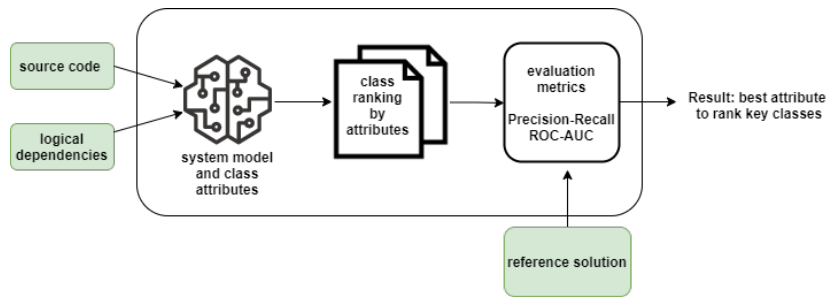


Fig. 6. Overview of the current approach

4.2. Experimental plan

Our study aims to check whether logical dependencies can be usable. Previous research focused on filtering the co-changes extracted from the versioning system and studying how filtering affects their size or how they overlap with structural dependencies. We intend to use the resulting information after co-changes filtering (the logical dependencies) in a tool that usually receives structural dependencies as input.

Our research questions are the following: **RQ1:** *Can logical dependencies combined with structural dependencies enhance the results obtained by using only structural dependencies in key class detection?* **RQ2:** *Can logical dependencies provide good results if they are used instead of structural dependencies in key class detection?* **RQ3:** *Does the connection strength filter has a favorable impact on the detection of key classes?*

To answer the research questions, we defined the following experimental plan: We will use the tool mentioned in the above section. Previously the tool had as input the structural dependencies of the system and the reference solution, and the output was the ROC-AUC score. The closer the ROC-AUC score is to 1, the better the results. With the slightly modified version of the tool, we are able to receive as input also logical dependencies.

For **RQ1**, we will give as input to the tool the structural and the logical dependencies, and we will compare the ROC-AUC scores obtained with the results obtained by using only structural dependencies.

Hypothesis: Key classes detection is better when we provide both types of dependencies as input to the tool. The output has a higher ROC-AUC score than the base approach.

Our findings for this research question can be found in section 5.3.

For **RQ2**, we will give as input to the tool only logical dependencies, and we will compare the ROC-AUC scores obtained with the results obtained by using only structural dependencies, and by using structural and logical dependencies combined. We do not expect that the results will be better compared to the results previously obtained, but we expect results that have a ROC-AUC score close to those values. That would mean that logical dependencies can provide enough information to detect most of the key classes of the system.

Hypothesis: The output has a ROC-AUC score between 0.7 and 1.

Our findings for this research question can be found in section 5.4.

For **RQ3**, we will generate two sets of logical dependencies. One set will be generated with the connection strength filter and one with the confidence filter. We will then use each

set in two different scenarios: one when we only use logical dependencies to detect key classes and one when we use logical and structural dependencies for detection. Finally, we will compare the results obtained. We expect that the connection strength filter will generate better results.

Hypothesis: The results obtained by using the connection strength filter are better than the ones obtained with the confidence filter.

Our findings for this research question can be found in section 5.5.

5. Experimental results using logical dependencies

As presented in section 3, the key class detection was previously done only by using the structural dependencies of the system. In this section, we use the same tool used in the baseline approach presented in section 3, and we add a new input to it, the logical dependencies.

In subsection 5.1, we present the data set used to generate new results and, in subsection 5.2, we present the previously obtained results. Subsection 5.3 presents the conclusions and results obtained by using logical and structural dependencies together, and subsection 5.4 presents the conclusions and results obtained by using only logical dependencies. Subsection 5.5 presents a comparison between results obtained with the confidence metric versus results obtained with the strength metric. And finally, subsection 5.6 presents a comparison between the results obtained in the current paper and the results of other researchers.

5.1. Data set used

The research of I. Sora et al. take into consideration structural dependencies that were extracted using static analysis techniques and were performed on the object-oriented systems presented in table 1 [29].

The requirements for a system to qualify as suited for investigations using logical dependencies are: has to be version controlled by Git, has to have releases for different code versions (previous research was done only on specific versions), and also has to have a significant number of commits. From the total of 14 object-oriented systems listed in the baseline [29], 13 of them have repositories in git 1, and from the found repositories, only 6 repositories have the same release tag as the specified version in previous research. The commit number found on the remaining 6 repositories varies from 19108 commits for Tomcat Catalina to 149 commits for JHotDraw. In order to have more accurate results, we need a significant number of commits (more than 5000 commits), so we concluded to use only 3 systems from the initial candidates for key classes detection using logical dependencies: Ant, Hibernate, and Tomcat Catalina.

5.2. Measurements using only the baseline approach

In table 2 are presented the ROC-AUC values for different attributes computed for the systems Ant, Tomcat Catalina, and Hibernate by using the baseline approach. We compare these values with the new values obtained by using also logical dependencies in key class detection.

Table 1. Systems and versions of the systems found in Git

ID	System	Version	Release Tag name	Commits number
S1	Apache Ant	1.6.1	rel/1.6.1	6713
S2	Argo UML	0.9.5	not found	0
S3	GWT Portlets	0.9.5 beta	not found	0
S4	Hibernate	5.2.12	5.2.12	6733
S5	javaclient	2.0.0	not found	0
S6	jEdit	5.1.0	not found	0
S7	JGAP	3.6.3	not found	0
S8	JHotDraw	6.0b.1	not found	149
S9	JMeter	2.0.1	v2_1_1	2506
S10	Log4j	2.10.0	v1_2_10-recalled	634
S11	Mars	3.06.0	not found	0
S12	Maze	1.0.0	not found	0
S13	Neuroph	2.2.0	not found	0
S14	Tomcat Catalina	9.0.4	9.0.4	19108
S15	Wro4J	1.6.3	v1.6.3	2871

Table 2. ROC-AUC metric values extracted

Metrics	Ant	Tomcat Catalina	Hibernate
PR_U2_W	0.95823	0.92341	0.95823
PR	0.94944	0.92670	0.94944
PR_U	0.95060	0.93220	0.95060
CONN_TOTAL_W	0.94437	0.92595	0.94437
CONN_TOTAL	0.94630	0.93903	0.94630

5.3. Measurements using combined structural and logical dependencies

The tool used in the baseline approach runs a graph-ranking algorithm on a graph that contains all the structural dependencies extracted from static source code analysis. Each edge in the graph represents a dependency. The entities that form a structural dependency are represented as vertices in the graph. As mentioned in section 3, we modified the tool to take structural and logical dependencies as input. For this subsection's measurements, we add the logical dependencies in the graph that contains all structural dependencies. Since it is a weighted graph, if a structural dependency is also a logical dependency, then the final weight of the connection is the sum of the weight computed for the structural dependency and the connection strength metric associated with the logical dependency.

In tables 3, 4, and 5, on each line, we have the computed the key class metric generated with logical dependencies extracted with the connection strength threshold that is specified in the columns header.

We started with logical dependencies that have a connection strength metric greater than 10, then we repeatedly increased the value by 10 until we reached 100. The last column of the table contains the results previously obtained by the tool by only using structural dependencies (the results presented in section 5.2). So, to answer *RQ1: Can logical dependencies combined with structural dependencies enhance the results obtained by using only structural dependencies in key class detection?*: The results obtained by combining structural and logical dependencies are close to the previously registered values but, in most cases, do not surpass them. Underlined are the values that are better than the previously registered values. We can observe that for all 3 systems, the best values obtained are for connection strength between 40-70.

Table 3. Measurements for Ant using structural and logical dependencies combined

Metrics	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100	Baseline
PR_U2_W	0.877	0.880	0.883	0.888	0.884	0.880	0.901	0.924	0.900	0.891	0.929
PR	<u>0.955</u>	<u>0.932</u>	<u>0.936</u>	<u>0.936</u>	<u>0.880</u>	<u>0.884</u>	<u>0.887</u>	<u>0.889</u>	<u>0.888</u>	<u>0.890</u>	0.855
PR_U	0.933	<u>0.937</u>	<u>0.936</u>	<u>0.939</u>	<u>0.940</u>	<u>0.939</u>	<u>0.941</u>	<u>0.943</u>	<u>0.942</u>	<u>0.940</u>	0.933
CON_T_W	0.841	0.839	0.836	0.838	0.835	0.849	0.859	0.872	0.870	0.874	0.934
CON_T	0.920	0.919	0.921	0.923	0.923	0.932	0.934	0.939	0.937	0.937	0.942

Table 4. Measurements for Tomcat Catalina using structural and logical dependencies combined

Metrics	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100	Baseline
PR_U2_W	0.862	0.883	0.898	0.901	0.907	0.909	0.910	0.916	0.918	0.918	0.923
PR	0.879	0.885	0.888	0.882	0.869	0.869	0.863	0.863	0.863	0.863	0.927
PR_U	0.924	0.930	0.931	0.932	0.932	0.932	0.932	0.932	0.932	0.932	0.932
CON_T_W	0.868	0.888	0.901	0.909	0.914	0.917	0.918	0.923	0.925	0.925	0.926
CON_T	0.925	0.934	0.937	0.938	0.938	0.938	0.938	0.938	0.938	0.938	0.939

Table 5. Measurements for Hibernate using structural and logical dependencies combined

Metrics	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100	Baseline
PR_U2_W	0.903	0.909	0.916	0.928	0.930	0.932	0.946	0.947	0.947	0.949	0.958
PR	0.956	0.959	0.961	0.962	0.962	0.962	0.953	0.953	0.953	0.954	0.949
PR_U	0.937	0.941	0.943	0.947	0.948	0.948	0.950	0.950	0.950	0.950	0.951
CON_T_W	0.864	0.872	0.879	0.896	0.898	0.900	0.929	0.930	0.931	0.934	0.944
CON_T	0.920	0.927	0.932	0.940	0.940	0.940	0.945	0.945	0.945	0.945	0.946

Some other details about the systems are presented in tables 6 and 7 . In table 6 are the overlappings between structural and logical dependencies expressed in percentages. Each column represents the percentage of logical dependencies that are also structural. The values obtained confirm that, indeed, the logical dependencies overlap with structural dependencies in a small percentage, and they must be treated as different dependencies.

In table 7 are the ratio numbers between structural dependencies and logical dependencies. We added this table to highlight how different the numbers of both dependencies are.

Table 6. Percentage of logical dependencies that are also structural dependencies

System	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100
Ant	17.628	19.872	20.461	20.858	21.078	23.913	24.688	21.807	20.000	19.776
Tomcat Catalina	10.331	14.931	15.862	16.221	16.427	16.302	16.598	18.336	19.207	19.149
Hibernate	8.005	8.971	9.755	12.060	12.348	12.254	18.426	19.105	18.836	19.371

In most cases, for all systems, the results tend to become better once with increasing the value of the connection strength threshold up until one point, after which the results obtained begin to drop. If we look at table 6, we can observe that the bigger the threshold for the connection strength filter, the smaller the number of total logical dependencies becomes. For example, in Hibernate, the value 70 for the connection strength threshold makes the structural dependencies outnumber 10 times the logical dependencies.

Table 7. Ratio between structural and logical dependencies (SD/LD)

System	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100
Ant	1.373	2.251	2.870	3.133	3.461	4.604	5.282	6.598	7.060	7.903
Tomcat Catalina	0.445	0.936	1.302	1.543	1.660	1.967	2.218	3.057	3.376	3.440
Hibernate	1.159	1.747	2.184	3.867	4.283	4.877	10.547	11.920	12.464	14.851

We can identify 3 scenarios based on tables 3, 4, 5 and 7. In the 1st scenario, the connection strength threshold is too small, and we remain with a lot of logical dependencies after filtering. The high volume of logical dependencies introduced in the graph might cause an erroneous detection of the key classes, in consequence, less performing measurements/results. This affirmation is sustained by the fact that, when the threshold begins to be more restrictive, and the total number of logical dependencies begins to decrease, the key classes detection starts to improve. The 2nd scenario assumes that the connection strength threshold is too big, significantly decreasing the number of logical dependencies. In this case, the logical dependencies introduced in the graph are too few to improve the detection, and, instead, will create noise in the graph and less performing results. This leads us to the 3rd scenario, in which the connection strength threshold is 'just right'. Not too small, because it will introduce too many logical dependencies in the graph and produce less performing results. And not too high, because it will decrease too much the number of logical dependencies, producing less performing results.

The 'just right' value can differ from one system to another, depending on the size of the system. If we look at Ant (the smaller size system), we can see that the results begin to decrease sooner than for Hibernate. On average, all the systems perform well between 40 and 70 for the connection strength threshold value.

5.4. Measurements using only logical dependencies

In the previous subsection, we added the logical and structural dependencies in the graph based on which the ranking algorithm works. Currently, we add only the logical dependencies to the graph.

In tables 8, 9, and 10 are presented the results obtained by using only logical dependencies to detect key classes.

For the second research question: '*RQ2: Can logical dependencies provide good results if they are used instead of structural dependencies in key class detection?*', the initial hypothesis is confirmed by the results obtained.

The measurements obtained are not as good as the ones using logical and structural dependencies combined or using only structural dependencies. But the values obtained are above 0.7, which means that a good part of the key classes is detected by using only logical dependencies. As mentioned in section 3.3, a classifier is good if it has the ROC-AUC value as close to 1 as possible.

One explanation for the less performing results is that the key classes may have a better design than the rest of the classes, which means that are less prone to change. If the key classes are less prone to change, then the associated connection strength metric has a lower value than other entities.

Tables 11, 13 and 12, provide us a better overview of the update behavior of key classes in the versioning system. The selected classes from all three tables are the key classes extracted from developer documentation [29]. The commit count column presents

Table 8. Measurements for Ant using only logical dependencies

Metrics	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100	Baseline
PR_U2_W	0.679	0.695	0.738	0.799	0.822	0.883	0.890	0.901	0.846	0.862	0.929
PR	0.868	0.776	0.767	0.825	0.822	0.850	0.834	0.863	0.844	0.860	0.855
PR_U	0.801	0.792	0.757	0.806	0.822	0.854	0.856	0.867	0.848	0.860	0.933
CON_T_W	0.819	0.825	0.818	0.817	0.813	0.828	0.843	0.861	0.845	0.854	0.934
CON_T	0.856	0.836	0.819	0.803	0.801	0.816	0.831	0.855	0.840	0.851	0.942

Table 9. Measurements for Tomcat Catalina using only logical dependencies

Metrics	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100	Baseline
PR_U2_W	0.775	0.810	0.834	0.828	0.819	0.815	0.805	0.816	0.820	0.813	0.923
PR	0.813	0.813	0.836	0.831	0.820	0.814	0.804	0.816	0.820	0.813	0.927
PR_U	0.772	0.815	0.835	0.831	0.820	0.814	0.804	0.816	0.819	0.813	0.932
CON_T_W	0.805	0.823	0.842	0.835	0.822	0.815	0.805	0.817	0.820	0.813	0.926
CON_T	0.787	0.812	0.835	0.832	0.821	0.814	0.804	0.817	0.820	0.813	0.939

Table 10. Measurements for Hibernate using only logical dependencies

Metrics	≥ 10	≥ 20	≥ 30	≥ 40	≥ 50	≥ 60	≥ 70	≥ 80	≥ 90	≥ 100	Baseline
PR_U2_W	0.721	0.733	0.743	0.700	0.700	0.703	0.741	0.742	0.744	0.751	0.958
PR	0.735	0.747	0.756	0.704	0.702	0.706	0.745	0.745	0.746	0.752	0.949
PR_U	0.738	0.740	0.749	0.699	0.701	0.704	0.744	0.743	0.745	0.752	0.951
CON_T_W	0.730	0.739	0.747	0.701	0.702	0.706	0.746	0.747	0.748	0.754	0.944
CON_T	0.740	0.743	0.750	0.700	0.700	0.704	0.746	0.746	0.747	0.753	0.946

the number of commits in which the entity was involved. The column 'Max occurrence with another entity' contains the maximum number of updates with another entity from the system (the strongest connection with another entity).

It can be observed that some key classes change a lot in the versioning system, for example, Configuration for Hibernate, ProjectHelper for Ant and StandardContext for Catalina. Also, some classes create strong connections with other entities, like IntrospectionHelper for Ant, Table for Hibernate and StandardContext for Catalina. But, in most cases, the key classes are not the entities that update the most in the versioning system. So, by setting too high the connection strength threshold, we risk filtering out the key classes.

Table 11. Ant key classes update overview

Key class name	Commit count	Max occurrence with another entity
org.apache.tools.ant.Task	40	13
org.apache.tools.ant.Target	39	16
org.apache.tools.ant.IntrospectionHelper	52	43
org.apache.tools.ant.RuntimeConfigurable	38	16
org.apache.tools.ant.ProjectHelper	67	17
org.apache.tools.ant.TaskContainer	6	2
org.apache.tools.ant.Main	56	21
org.apache.tools.ant.UnknownElement	47	16
org.apache.tools.ProjectHelper2\$ElementHandler	21	14

Table 12. Hibernate key classes update overview

Key class name	Commit count	Max occurrence with another entity
org.hibernate.Query	9	1
org.hibernate.engine.spi.SessionFactoryImplementor	26	10
org.hibernate.SessionFactory	20	3
org.hibernate.mapping.Table	39	25
org.hibernate.criterion.Projection	2	0
org.hibernate.criterion.Criterion	2	0
org.hibernate.engine.spi.SessionImplementor	16	2
org.hibernate.cfg.Configuration	88	9
org.hibernate.mapping.Column	16	3
org.hibernate.type.Type	10	0
org.hibernate.Transaction	9	0
org.hibernate.engine.ConnectionProvider	2	0
org.hibernate.Session	25	14
org.hibernate.Criteria	10	1

Table 13. Tomcat Catalina key classes update overview

Key class name	Commit count	Max occurrence with another entity
org.apache.catalina.Session	15	8
org.apache.catalina.Loader	7	1
org.apache.catalina.startup.Catalina	46	32
org.apache.catalina.Pipeline	9	3
org.apache.catalina.core.StandardHost	55	38
org.apache.catalina.Container	25	16
org.apache.catalina.Wrapper	18	13
org.apache.catalina.core.StandardService	42	12
org.apache.catalina.startup.HostConfig	60	43
org.apache.catalina.core.StandardContext	242	213
org.apache.catalina.core.StandardServer	46	12
org.apache.catalina.Realm	17	8
org.apache.catalina.connector.CoyoteAdapter	153	129
org.apache.catalina.core.StandardWrapper	82	25
org.apache.catalina.Valve	9	2
org.apache.catalina.connector.Request	208	178
org.apache.catalina.Context	91	68
org.apache.catalina.connector.Connector	80	18
org.apache.catalina.Server	15	8
org.apache.catalina.connector.Response	102	28
org.apache.catalina.core.StandardEngine	30	17
org.apache.catalina.startup.Bootstrap	26	5
org.apache.catalina.Host	19	11
org.apache.catalina.LifecycleListener	5	1
org.apache.catalina.core.StandardPipeline	25	6
org.apache.catalina.Manager	22	15
org.apache.catalina.Service	15	8
org.apache.catalina.Engine	4	1

5.5. Comparison between results obtained with strength versus confidence metric

As mentioned in section 2.4, we did not use the confidence metric because it does not consider the big picture of the system. A co-changing pair $A \rightarrow B$, where A updates only once in the entire history, and when it updates, it updates together with B , will have the best confidence value that we can get. This is why we introduced the strength metric, to

balance the metric in the favor of those which update more frequently. Since both metrics require the same inputs and only the calculation method is different, we computed with our tool the confidence metric and applied the same threshold to it as to the strength metric. The only difference from how other authors computed the metric is that we multiplied its value by 100. So, the confidence values can fluctuate between 0 and 100. In the graph used by the key classes detection tool, the structural dependencies weights are supraunitary values. So, we multiplied with 100 the confidence value to scale it to the structural dependencies weights. Otherwise, if we add a subunitary value (confidence value) to a high value (the structural weight), it will not make a difference, so we will not be able to see the impact of the logical dependencies in the graph.

Table 14. Average results obtained with strength versus confidence metric

Metric used	Using	
	Only logical dependencies	Structural and logical dependencies
Average values obtained for all systems		
strength	0.791	0.916
confidence	0.731	0.893
Average values obtained for Ant		
strength	0.826	0.903
confidence	0.741	0.873
Average values obtained for Tomcat Catalina		
strength	0.816	0.910
confidence	0.752	0.878
Average values obtained for Hibernate		
strength	0.732	0.935
confidence	0.699	0.929

The comparison between the average values obtained by using the confidence metric and the strength metric can be found in table 14.

These results help us answer the third research question: *RQ3: Does the connection strength filter has a favorable impact on the detection of key classes?*. As we expected in our initial hypothesis and now based on the results, we can say that the connection strength metric is more suited for logical dependencies detection. So, by considering the mean update frequency of the entire system in the filtering process, we improve the detection of logical dependencies.

5.6. Comparison and discussion on the obtained results

In this subsection, we compare the results obtained in the current paper with results previously obtained by other researchers.

Even though the approaches were not the same, most of them used the ROC-AUC metric to evaluate the quality of the results, the same as ours. Osman et al. obtained in their research an average ROC-AUC score of 0.750 [17]. Thung et al. obtained an average ROC-AUC score of 0.825 [24] and Şora et al. (our baseline approach) obtained an average ROC-AUC score of 0.894 [29].

In the current research, we obtained an average ROC-AUC score of 0.916 when using logical and structural dependencies combined and a score of 0.791 when using only logical dependencies to detect key classes.

So, when using both dependencies combined, we can obtain a slightly better ROC-AUC score than the one from the baseline approach. And, when using only logical dependencies, even though we do not obtain a better score than the baseline approach, we obtain results that can be compared with results obtained by other researchers.

6. Threats to validity

We will present in this section some aspects that can constitute threats to the validity of the results from this paper. First, we extract co-changes only from main (master) branch commits. Development branches are not taken into consideration, which might cause some information loss. Especially if there are many branches in development with many commits in them. On the other hand, if we designed the tool that extracts co-changes to consider also development branches, that would have constituted another threat to the validity. Some branches are just for trial and error or prototyping, or sometimes they never get integrated into the main branch, which means that we risk analyzing information that does not reflect the reality of the system. Another threat to the validity of the results is that we do not consider the age of the co-changes. It can happen that two entities updated together a lot because they were also structurally related entities, but, at some point in time, that connection was removed from the code, and they do not update together any longer. In this case, it can happen that the tool still considers them logical dependencies due to the frequency of updates. In future works, we will try to identify outdated connections.

7. Conclusions

In this paper, we studied the filtering and the usage of co-changes extracted from the versioning system. In the first part of the paper, we focused on the co-changes filtering, and in the second part, we used the filtered co-changes (logical dependencies) to detect key classes. For co-changes filtering, we applied the commit size filter and the filter based on connection strength. The co-changes that remained after filtering, called logical dependencies, were provided as input for a tool that detects key classes.

We approached two scenarios to detect key classes by using logical dependencies. In the 1st scenario, we used logical dependencies together with structural dependencies, and in the 2nd, we used only logical dependencies to detect the key classes. We modified the tool used in the baseline approach for detecting key classes from structural dependencies [29], to use also logical dependencies.

Based on the results obtained, compared with the baseline results, we saw a slight improvement in key class detection when both logical and structural dependencies were used together. The best results were obtained with a connection strength threshold of 40-70. Also, our connection strength metric performs better than the confidence metric used in related works.

When we used only logical dependencies to detect key classes, the results were less performing than our results when using only structural or structural and logical dependencies combined, but they were comparable with results of related work using structural dependencies. We consider this a very positive result because this research uses a different type of input than the previous ones, the logical dependencies. It is also an open door for

new research in multiple fields that use structural dependencies to gain knowledge about software systems. And since logical dependencies are easy and fast to extract from the versioning system and do not depend on the language of the software system, the cost of integrating them is small.

To sum up the findings of this paper, logical dependencies can be used to gain knowledge about software systems. We consider that the advantage of using only logical dependencies is that it only uses data extracted from the versioning system and can be generalized to various programming languages.

In the future, we want to check if other areas can be improved by using logical dependencies, like software clustering [7], [19], [4].

References

1. Ajenka, N., Capiluppi, A.: Understanding the interplay between the logical and structural coupling of software classes. *Journal of Systems and Software* 134, 120–137 (2017), <https://doi.org/10.1016/j.jss.2017.08.042>
2. Ajenka, N., Capiluppi, A., Counsell, S.: An empirical study on the interplay between semantic coupling and co-change of software classes. *Empirical Software Engineering* 23(3), 1791–1825 (2018), <https://doi.org/10.1007/s10664-017-9569-2>
3. Cataldo, M., Mockus, A., Roberts, J.A., Herbsleb, J.D.: Software dependencies, work dependencies, and their impact on failures. *IEEE Transactions on Software Engineering* 35, 864–878 (2009)
4. Şora, I.: Software architecture reconstruction through clustering: Finding the right similarity factors. In: *Proceedings of the 1st International Workshop in Software Evolution and Modernization - Volume 1: SEM, (ENASE 2013)*. pp. 45–54. INSTICC, SciTePress (2013)
5. Şora, I.: Helping program comprehension of large software systems by identifying their most important classes. In: *Evaluation of Novel Approaches to Software Engineering - 10th International Conference, ENASE 2015, Barcelona, Spain, April 29-30, 2015, Revised Selected Papers*. pp. 122–140. Springer International Publishing (2015)
6. Şora, I.: Helping program comprehension of large software systems by identifying their most important classes. In: Maciaszek, L.A., Filipe, J. (eds.) *Evaluation of Novel Approaches to Software Engineering*. pp. 122–140. Springer International Publishing, Cham (2016)
7. Şora, I., Glodean, G., Gligor, M.: Software architecture reconstruction: An approach based on combining graph clustering and partitioning. In: *Computational Cybernetics and Technical Informatics (ICCC-CONTI), 2010 International Joint Conference on*. pp. 259–264 (May 2010)
8. D’Ambros, M., Lanza, M., Lungu, M.: The evolution radar: Visualizing integrated logical coupling information. pp. 26–32 (01 2006)
9. D’Ambros, M., Lanza, M., Lungu, M.: Visualizing co-change information with the evolution radar. *IEEE Transactions on Software Engineering* 35(5), 720–735 (2009)
10. David W. Hosmer, S.L.: *Assessing the Fit of the Model*, chap. 5, pp. 143–202. John Wiley and Sons, Ltd (2000), <https://onlinelibrary.wiley.com/doi/abs/10.1002/0471722146.ch5>
11. Ducasse, S., Pollet, D.: Software architecture reconstruction: A process-oriented taxonomy. *IEEE Transactions on Software Engineering* 35(4), 573–591 (July 2009)
12. Gall, H., Hajek, K., Jazayeri, M.: Detection of logical coupling based on product release history. In: *Proceedings of the International Conference on Software Maintenance*. pp. 190–. ICSM ’98, IEEE Computer Society, Washington, DC, USA (1998), <http://dl.acm.org/citation.cfm?id=850947.853338>
13. Graves, T., Karr, A., Marron, J., Siy, H.: Predicting fault incidence using software change history. *IEEE Transactions on Software Engineering* 26(7), 653–661 (2000)

14. Jankovic, M., Zitnik, S., Bajec, M.: Reconstructing de facto software development methods. *Computer Science and Information Systems* 16, 38–38 (01 2018)
15. Oliva, G.A., Gerosa, M.A.: On the interplay between structural and logical dependencies in open-source software. In: *Proceedings of the 2011 25th Brazilian Symposium on Software Engineering*. pp. 144–153. SBES '11, IEEE Computer Society, Washington, DC, USA (2011), <https://doi.org/10.1109/SBES.2011.39>
16. Oliva, G.A., Gerosa, M.A.: Experience report: How do structural dependencies influence change propagation? an empirical study. In: *26th IEEE International Symposium on Software Reliability Engineering, ISSRE 2015, Gaithersbury, MD, USA, November 2-5, 2015*. pp. 250–260 (2015), <https://doi.org/10.1109/ISSRE.2015.7381818>
17. Osman, M.H., Chaudron, M.R.V., v. d. Putten, P.: An analysis of machine learning algorithms for condensing reverse engineered class diagrams. In: *2013 IEEE International Conference on Software Maintenance*. pp. 140–149 (2013)
18. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (November 1999), <http://ilpubs.stanford.edu:8090/422/>, previous number = SIDL-WP-1999-0120
19. Shtern, M., Tzerpos, V.: Clustering methodologies for software engineering. *Adv. Soft. Eng.* 2012, 1:1–1:1 (Jan 2012), <http://dx.doi.org/10.1155/2012/792024>
20. Şora, I.: A PageRank based recommender system for identifying key classes in software systems. In: *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics (SACI)*. pp. 495–500 (May 2015)
21. Stana, A.D., Şora, I.: Analyzing information from versioning systems to detect logical dependencies in software systems. In: *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. pp. 000015–000020 (2019)
22. Stana, A.D., Şora, I.: Identifying logical dependencies from co-changing classes. In: *Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE*. pp. 486–493. INSTICC, SciTePress (2019)
23. Tahvildari, L., Kontogiannis, K.: Improving design quality using meta-pattern transformations: a metric-based approach. *J. Softw. Maintenance Res. Pract.* 16, 331–361 (2004)
24. Thung, F., Lo, D., Osman, M.H., Chaudron, M.R.V.: Condensing class diagrams by analyzing design and network metrics using optimistic classification. In: *Proceedings of the 22nd International Conference on Program Comprehension*. p. 110–121. ICPC 2014, Association for Computing Machinery, New York, NY, USA (2014), <https://doi.org/10.1145/2597008.2597157>
25. Yang, X., Lo, D., Xia, X., Sun, J.: Condensing class diagrams with minimal manual labeling cost. In: *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*. vol. 1, pp. 22–31 (2016)
26. Zaidman, A., Demeyer, S.: Automatic identification of key classes in a software system using webmining techniques. *Journal of Software Maintenance and Evolution: Research and Practice* 20(6), 387–417 (2008)
27. Zimmermann, T., Weisgerber, P., Diehl, S., Zeller, A.: Mining version histories to guide software changes. In: *Proceedings of the 26th International Conference on Software Engineering*. pp. 563–572. ICSE '04, IEEE Computer Society, Washington, DC, USA (2004), <http://dl.acm.org/citation.cfm?id=998675.999460>
28. Şora, I.: Finding the right needles in hay - helping program comprehension of large software systems. In: *Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE*. pp. 129–140. INSTICC, SciTePress (2015)
29. Şora, I., Chirila, C.B.: Finding key classes in object-oriented software systems by techniques based on static analysis. *Information and Software Technology* 116, 106176 (2019), <https://www.sciencedirect.com/science/article/pii/S0950584919301727>

Ioana Şora is an Associate Professor in the Department of Computer and Information Technology at the Politehnica University of Timisoara, Romania. She received the Phd degree in Computer Science in 2003 from the Politehnica Univerity of Timisoara. Her research interests are in the following domains: Software architecture and reconstuction; Program analysis and program comprehension; Recommender systems for software engineering; Dynamic and self-adaptive software architectures; Component based software engineering.

Adelina Diana Stana is a Ph.D. student at the Politehnica University of Timisoara. In 2016 she received her bachelor's degree at Politehnica, and in 2018 her master's degree. Since 2014 she is also employed at Continental Automotive Group, her current role is Software Architect. Her research interests are focusing on software evolution and maintenance. Her work has been on improving the understanding and design of software systems.

Received: May 18, 2022; Accepted: February 28, 2023.

A Hierarchical Federated Learning Model with Adaptive Model Parameter Aggregation

Zhuo Chen¹, Chuan Zhou¹ and Yang Zhou²

¹ College of Computer Science and Engineering, Chongqing University of Technology
Chongqing, China

chenzhuo@cqut.edu.cn

czhou@2020.cqut.edu.cn

² Department of Computer Science and Software Engineering, Auburn University
Auburn, USA

yangzhou@auburn.edu

Abstract. With the proposed Federated Learning (FL) paradigm based on the idea of “data available but invisible”, participating nodes which create or hold data can perform local model training in a distributed manner, then a global model can be trained only by continuously aggregating model parameters or intermediate results from different nodes, thereby achieving a balance between data privacy protection and data sharing. However, there are some challenges when deploying a FL model. First, there may be hierarchical associations between participating nodes, so that the datasets held by each node are no longer independent of each other. Secondly, due to the possible abnormal delay of data transmission, it can seriously influence the aggregation of model parameters. In response to the above challenges, this paper proposes a newly designed FL framework for the participating nodes with hierarchical associations. In this framework, we design an adaptive model parameter aggregation algorithm, which can dynamically decide the aggregation strategy according to the state of network connection between nodes in different layers. Additionally, we conduct a theoretical analysis of the convergence of the proposed FL framework based on a non-convex objective function. Finally, the experimental results show that the proposed framework can be well applied to applications in different network connections, and can achieve faster model convergence efficiency while ensuring the accuracy of the model prediction.

Keywords: Parameter Aggregation, Federated Learning, Internet of Things, Privacy Computing.

1. Introduction

Traditional machine learning models usually need to collect data generated in distributed locations into a central storage point (e.g., a cloud data center) for model training. However, with the increase in the number of mobile terminals and Internet of Things (IoT) sensors, the data generated is not only more diverse in data types and formats, but also the scale of data is also proliferating. While larger-scale data can help train better machine learning models, transferring large amounts of data consumes more network resources [1]. In addition, there is a non-negligible risk of information leakage in the process of data collection, transmission and storage [2]. Furthermore, data holders are increasingly

reluctant to transfer data to other uncontrolled locations for the purpose of privacy protection. These issues pose challenges for building machine learning models in an environment where data is increasingly fragmented and isolated. In recent years, service nodes deployed close to users have been greatly improved in terms of computing power, storage resources, and network transmission capabilities, which have laid the foundation for building distributed machine learning models based on these distributed nodes [3,4]. In particular, Federated Learning (FL) [5] proposed based on the concept of “data does not move while model moves”, enables collaborative learning among multiple participating nodes without the data leaving the place they are generated, which is called as FedAvg. One global model is trained only by continuously aggregating model parameter or intermediate results, thereby achieving a balance between data privacy protection and data sharing. This new type of machine learning paradigm has recently received continuous attention from academic community.

A typical FL model can be regarded as a two-layers FL framework[6] composed of a parameter server (PS) with sufficient computing power (e.g., a server deployed in cloud data center) and multiple clients with acceptable computing capability (e.g., edge service nodes) . The operation process of a typical FL model is demonstrated in Fig.1. The client independently performs local model training based on local dataset, and global model is optimized through the exchange of model parameters under the encryption mechanism. Then, the global model is transferred to the clients for facilitating local training next time. The whole process continues until the model converges, or reaches the maximum number of iterations. Since PS obtains model gradients or model parameters rather than raw data from clients, the purpose of protecting the privacy can be achieved [6,7]. However, in the practical scenarios, there is often a more complex hierarchical relationship between the data holders, and the parameters exchange between clients and PS may be fulfilled by IoT network with unstable transmission quality. When faced with such kind of scenario, the existing work lacks some considerations on two issues: 1) The data generated by multiple clients may have explicit or implicit correlations. This means that the data features generated by nodes at the lower layers of the hierarchical relationship will affect data distribution at higher layers, which no longer makes the dataset on each participating node completely independent. 2) Model quality and model convergence may be affected by the quality of network transmission. Specifically, when the network transmission is abnormal, the model parameters cannot be transferred between PS and the clients in time. If PS uses the synchronous aggregation method [8], the training time will be prolonged. In contrast, if the PS adopts a purely asynchronous aggregation method [9], although the training time can be reduced, the convergence of the model is unsatisfactory.

To address the above issues, a hierarchical federated learning (HFL) framework is proposed in this paper. In this framework, multiple nodes participating in collaborative learning are logically divided into multiple layers. There is an association relationship between the data generated by the nodes at the lower layer and the data generated by the nodes at the upper layer. In addition, the nodes in the middle layer (named Intermediate Aggregation Node, IAN) will not only aggregate the model parameters passed by its lower-layer nodes, but also perform local training based on the data generated by itself. We continually propose an adaptive parameter aggregation strategy. Based on this strategy, the IAN can adaptively adjust the aggregation method according to the quality of IoT-based data transmission to improve the model convergence performance and reduce

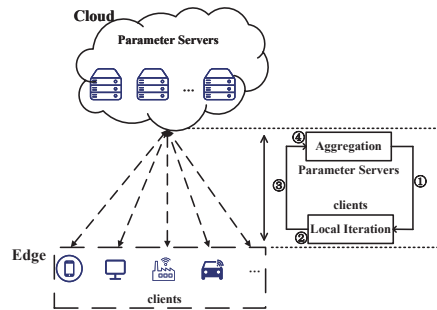


Fig. 1. The operation procedures for a typical FL model

the training time. Furthermore, we conduct a theoretical analysis of the convergence for the proposed HFL model with non-convex objective function, demonstrating that the convergence of the proposed HFL framework depends on aggregation frequencies and the total number of training rounds. Finally, using the HFL model, we achieve water demand forecasting in sub-regional and hierarchical water supply scenarios involving multiple water supply companies. The main contributions of this paper are summarized as follows.

- We propose a HFL model that integrates adaptive parameter aggregation algorithm. Under different network transmission delay, this framework can help multiple nodes that have hierarchical relationships and participate in joint learning to achieve better model training performance.
- We conduct an in-depth analysis of the model convergence and examine key parameters that have important impact on convergence.
- We establish the proposed HFL model to realize the prediction of urban water demand. Furthermore, through comparing with similar work, we finally verify the effectiveness and efficiency of the model.

The rest of the paper is organized as follows. Related work is discussed in Section 2. Section 3 proposes the HFL framework and presents an adaptive aggregation strategy. In Section 4, we conduct an in-depth convergence analysis of the proposed HFL framework. Section 5 describes the case of realizing urban water demand prediction based on the HFL, and the experimental results of the model are presented. Finally, the paper is concluded in section 6.

2. Related Work

In this section, we will discuss the representative HFL frameworks and the existing efforts on performance improvement of HFL.

Some HFL models have recently been proposed for the so-called “Terminal-Edge-Cloud” network service architecture. Reference [10] builds a layered FL model by relying on terminals, edge nodes and cloud servers as participants in joint learning. By taking advantage of the respective advantages of edge nodes and cloud server, the rational use of

computing and communication resources can be realized. Z. Wang et al.[11] treat cloud servers and edge nodes as global aggregator and cluster aggregator, then an asynchronous aggregation method and a synchronous aggregation method are adopted to achieve parameter aggregation, respectively. This work only performs a convergence analysis based on a convex objective function. In addition, W. Lim et al.[12] design a resource allocation mechanism and an incentive mechanism for a hierarchical FL architecture. The above-mentioned HFL frameworks do not consider the correlation between the data generated by multiple nodes participating in joint learning when designing the model. Although the above mentioned works proposed multiple hierarchical structure based FL models to balance the local iterations and runtime, it has not taken into account the potential relationship between the data held by the participating nodes.

Different from the centralization-based machine learning model, the training datasets of different scales held by participating nodes, the non-independent and identically distributed (non-IID) characteristics between different datasets, and the unstable network transmission quality are all potential factors that may affect the performance of FL models. To address these challenges, McMahan et al. [5] proposed the FedAvg algorithm, which enables participants to perform gradient descent independently, and finally the aggregation node averages the staged gradient values of clients to achieve model aggregation. Furthermore, Li Tian et al. [13] proposed an algorithm called FedProx, which can be effectively applied to highly heterogeneous environments and obtained satisfactory convergence. In [14], the synchronous aggregation mechanism is adopted to realize the parameter interaction between PS and clients, but it is difficult to applied to the scenarios with unstable network environment. In addition, for IoT networks with unstable network links, Chen et al. [15] proposed a lightweight node selection strategy based on an asynchronous FL model, which can improve the model training efficiency. H. Zhu et al. [16] considered availability and fairness in the client nodes scheduling process, and designed an asynchronous aggregation algorithm to improve the convergence of the model. C. Chen et al. [17] proposed an adaptive parameters transmission algorithm. The model parameters that are temporarily stable will not participate in the network transmission process, thereby reducing network bandwidth consumption. J. Liu et al. [18] combined Deep Reinforcement Learning (DRL) to propose an adaptive algorithm that adjusts the number of nodes participating in joint learning, and intelligently adjusts the local updates delivered to the PS according to the network state during each round of aggregation. J. Jin et al. [19] applied an adaptive optimization algorithm to FL to accelerate model convergence. These existing works are trying to improve the performance of HFL from the selection of nodes involved in the aggregation, the improvement of model training efficiency, the transmission of training parameters, and the design of the local update mechanism, etc. However, as far as we know, they have not considered adaptive parameters aggregation in the case of complex associations between multiple participating nodes in the collaborative learning.

3. A Hierarchical Federated Learning Framework

In this section, we first introduce the proposed HFL framework in detail, and then propose an adaptive parameter aggregation algorithm for HFL.

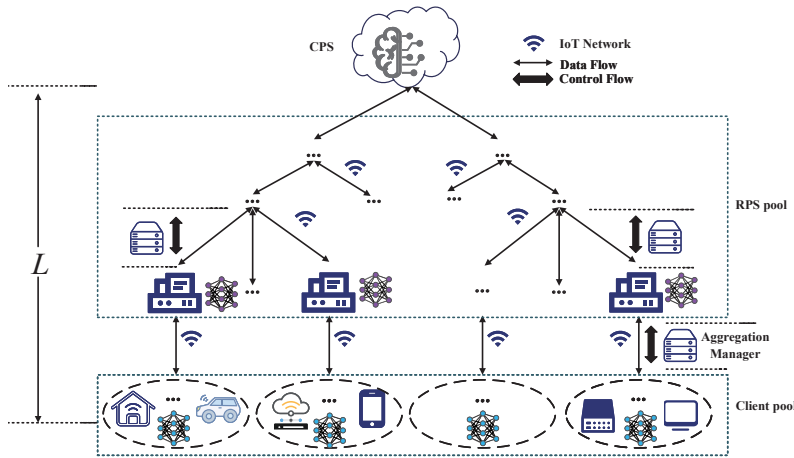


Fig. 2. The architecture of the proposed HFL

3.1. The description of the proposed HFL Model

As shown in Fig.2, we propose a newly designed HFL framework. In this framework, nodes are logically divided into multiple layers, and nodes between two adjacent layers can be interconnected through IoT-based network to realize data transmission. These participating nodes, from the role of performing FL, consist of Central Parameter Server (CPS), Regional Parameter Server (RPS) and clients. Since the RPS is located in the middle layer (or layers) of the proposed framework, it can connect CPS and clients at the same time. Therefore, a RPS actually plays the role of an IAN. Since then, we will use RPS to replace IAN to make further description. Specifically, the CPS is usually one cloud server with powerful computing capability. The CPS trains the global model and can interact with multiple RPSs. One RPS is typically one edge server with IoT connectivity that not only generates or collects data, but also trains regional model. At the same time, the RPS can aggregate the parameter updates of other RPSs or clients. A client is usually acted as an IoT terminal or a light-wight edge service node with acceptable computing capabilities. A client mainly performs local model training and interacts with RPSs with model parameters. These three types of participating nodes cooperate with each other to aggregate the model parameters and complete the FL model training. In addition, the framework also includes an Aggregation Manager (AM), which can periodically check the quality of the current IoT network. AM is the basis for the adaptive aggregation strategy with CPS and RPS. In a practical scenario, AM may be monitoring nodes managed by telecom operators responsible for operating the IoT network.

We assume that the area where HFL will be deployed can be logically divided into S sub-regions, that is, the HFL framework includes S RPSs. There are K clients in each sub-region. For the distinction in description, the parameter aggregation on the RPS, the model obtained by RPS aggregation, the local training on the RPS and the corresponding training model are called regional aggregation, regional model, regional training and regional updated model, respectively. Additionally, the aggregation on the CPS and the model aggregated are called global aggregation and global model, respectively. The local

Table 1. Summary of main notations

Notation	Description
\mathbf{K}, K	The set of clients associated with a RPS, the number of clients associated with a RPS
\mathbf{S}, S	The set of all RPSs, the number of RPSs
\mathcal{D}_k, D_k	Local dataset of client k , the size of \mathcal{D}_k
\mathcal{D}_s, D_s	Local dataset of RPS s , the size of \mathcal{D}_s
\mathcal{D}, D	The whole dataset, the size of \mathcal{D}
Γ^r	The size of $\bigcup_{k \in \alpha_c \mathbf{K}} \mathcal{D}_k$ on round r
B	Global aggregation interval
K_1	The whole number of training rounds for a client ($K_1 = B\kappa_1\kappa_2$)
H	The number of performing iterations per round
K_2	The whole number of training rounds for a RPS ($K_2 = B\kappa_3$)
r	Index of local training rounds
t	Index of regional training rounds
α_c	A certain fraction of \mathbf{K}
α_s	A certain fraction of \mathbf{S}
f	Global loss function on dataset $\bigcup_{s \in \alpha_s \mathbf{S}} \mathcal{D}_s$
F_s	Edge loss function on dataset \mathcal{D}_s
f_s	Edge loss function on dataset $\bigcup_{k \in \alpha_c \mathbf{K}} \mathcal{D}_k$
f_k	client loss function on dataset \mathcal{D}_k
Q	The true transmission delay to evaluate the network quality
T	The accepted transmission delay

training on the client and the corresponding model are called local training and local updated model, respectively. The global model w_0 with traditional FL is usually initialized in a random manner and broadcast by central server to others. However, in the HFL, the initial global model w_0 is learned from the common features of each node's dataset by CPS, and w_0 is broadcast to RPSs and clients. RPSs and clients start the training of local models based on the local dataset and initial weights. The training of the local model is performed in a parallel and distributed manner. The way of local update is performed as follows

$$\begin{aligned}
 H \text{ iterations} & \begin{cases} w_{r,1}^k = w_0 - \eta \nabla f_1^k(w_0) \\ w_{r,2}^k = w_{r,1}^k - \eta \nabla f_2^k(w_{r,1}^k) \\ \dots \end{cases} \\
 \Rightarrow w_{r,H}^k & = w_0 - \eta \sum_{i=1}^H \nabla f_i^k(w_{r,i}^k),
 \end{aligned} \tag{1}$$

where $w_{r,H}^k$ is the local update obtained by node k in the round r after H local iterations, and η is the learning rate. In particular, when $r = H = 0$, $w_{r,H}^k = w_0$. Then, integrating the iterative results of H times, the final equation in Eq.(1) can be obtained. The optimization method adopted in Eq.(1) is SGD, and Adam optimizer can also be applied [20]. After the local updates from clients are obtained by RPSs, FedAvg is used to obtain the regional model w_r^s , and the aggregation method can be represented as follows

$$w_r^s = \sum_{k=1}^{\alpha_c K} \frac{D_k}{\Gamma_s^r} w_{r,H}^k, \quad (2)$$

where $\alpha_c(\alpha_c \in \{\frac{1}{K}, \frac{2}{K}, \dots, 1\})$ represents the proportion of clients selected to participate in the aggregation, the dataset on client k is represented by \mathcal{D}_k , the size of \mathcal{D}_k is D_k ($D_k \triangleq |\mathcal{D}_k|$, where $|\cdot|$ denotes the cardinality). The size of the whole dataset in sub-region s with $\alpha_c K$ clients is $\Gamma_s^r = |\bigcup_{k \in \alpha_c K} \mathcal{D}_k|$.

After clients and RPS have completed $\kappa_1 \kappa_2$ rounds of local training and κ_2 times of regional aggregations respectively, then RPS performs iterative training based on its own dataset and generates a regional update w_s . The iterative process is the same as Eq.(1). Unlike clients, the number of iteration in one RPS is needed to execute κ_3 rounds to end. After each round of iterative execution, the RPS uploads w_s to the upper-layer RPS or CPS for aggregating a wider range of regional model or global model. In the proposed HFL model, the evolution of local weight $w_{r,i}^k$ of client k can be represented as follow

$$w_{r,i}^k = \begin{cases} w_{r,i-1}^k - \eta \nabla f_i^k(w_{r,i-1}^k), & \text{if } i \geq 1, r \mid \kappa_1 \neq 0 \\ \frac{\sum_{k \in \alpha_c K} D_k (w_{r,H-1}^k - \eta \nabla f_H^k(w_{r,H-1}^k))}{\Gamma_s}, & \text{if } r \mid \kappa_1 = 0 \\ \frac{\sum_{s \in \alpha_s S} D_s (w_{t,H-1}^s - \eta \nabla f_H^s(w_{t,H-1}^s))}{D}, & \text{if } r \mid \kappa_1 \kappa_2 = r \mid \kappa_3 = 0 \end{cases} \quad (3)$$

The update process of the weight w_s of the RPS in the sub-region s is similar to Eq.(3). According to the description of Fig.2, the parameter aggregation and training process of the proposed HFL is shown in Algorithm 1. Especially, the loss function at CPS is

$$\min_{w \in \mathbb{R}} f(w) = \sum_{s=1}^{\alpha_s S} \frac{D_s}{D^t} F_s(w), \quad (4)$$

where $F_s(w) = 1/D_s \cdot \sum_p^{D_s} \mathcal{F}_s(w_s, \zeta_p)$, $D_s \triangleq |\mathcal{D}_s|$ and $D^t = |\bigcup_{s \in \alpha_s S} \mathcal{D}_s|$. Especially, $\mathcal{F}_s(w_s, \zeta_p)$ is the loss function of the p -th data sample. Furthermore, as shown in Eq.(5), there is a weight w_r^s that minimizes the regional loss function.

$$w_r^s = \arg \min f_s = \arg \min \sum_{k=1}^{\alpha_c K} \frac{D_k}{\Gamma_s^r} f_k. \quad (5)$$

3.2. An adaptive parameter aggregation method for HFL

In this part, an adaptive parameter aggregation algorithm is proposed, which dynamically integrates synchronous and asynchronous aggregation method into the proposed HFL framework, so as to enable different nodes in the HFL framework (i.e., between CPS and RPSs, between RPS and RPS, and between RPS and clients) can perform adaptive parameter aggregation according to current connection state of the wireless IoT network. The monitoring of the connection status is performed by the AM. The AM informs the corresponding nodes of the connection status information of the IoT network to dynamically adjust the aggregation strategy adopted between the corresponding nodes (i.e.,

Algorithm 1 The parameter aggregation and training process of HFL

Input: the initial global model w_0 , the number of clients that belong to one RPS: K , the number of RPS: S , the learning rate η , other parameters are related to training rounds: B, κ_1, κ_2 .

Output: the final global model w .

- 1: **for** each round $r = 1, 2, 3, \dots, B\kappa_1\kappa_2$ **do**
- 2: **for** each client $k = 1, 2, \dots, K$ in parallel **do**
- 3: $w_r^k = w_{r-1}^k - \eta \nabla f_k$ /* Processing at the clients*/.
- 4: **end for**
- 5: **if** $r|\kappa_1 = 0$ **then**
- 6: Send w_r^k back to related RPS by client k .
- 7: **for** each RPS $s = 1, 2, \dots, S$ in parallel **do**
- 8: Aggregate the local models by order in which clients arrive according to Eq.(2) for getting w_r^s /*Processing at the RPS*/.
- 9: Send w_r^s to related clients again.
- 10: **end for**
- 11: **end if**
- 12: **if** $r|\kappa_1\kappa_2 = 0$ **then**
- 13: **for** each RPS $s = 1, 2, \dots, S$ in parallel **do**
- 14: Aggregate the local models by order in which clients arrive to get w_s^0 .
- 15: **for** $j = 1, 2, \dots, \kappa$ **do**
- 16: $w_s^j \leftarrow w_s^{j-1} - \eta \nabla f_s$.
- 17: **end for**
- 18: Send w_s^j back to CPS by RPS.
- 19: **end for**
- 20: Aggregate the regional models by order in which RPS arrive to get w /*Processing at the CPS*/.
- 21: Send w to all edge devices(RPS,client) as the new w_0 for the next round.
- 22: **end if**
- 23: **end for**

between CPS and RPS, between RPS and RPS, and between RPS and clients). We use Fig.3 to represent the complete training process performed by different types of nodes in HFL. First, a specific client will complete κ_1 rounds of local training, and then send the local model parameters to the corresponding RPS, so that the RPS can complete the aggregation of the regional model. When the above process is performed κ_2 times, the RPS will use the aggregated regional model parameters and its own private data as input, and then continue to complete κ_3 rounds of regional training. The obtained regional model parameters are then uploaded to the CPS through the IoT network to complete the training of the global model. Until the end of the global model training, the total number of iterations of the process is B , in which the total number of local training rounds in each client is K_1 ($K_1 = B\kappa_1\kappa_2$), and the total number of regional training rounds in each RPS is K_2 ($K_2 = B\kappa_3$). Furthermore, the threshold T is defined to represent the minimum acceptable transmission delay during the model training. Before RPS and CPS perform parameter aggregation, they will obtain the information of transmission delay between nodes located in two adjacent layers periodically detected by AM. The delay is represented by the parameter Q . When $Q > T$, it means that the current communication quality is unsatisfactory. Therefore, an asynchronous aggregation mechanism is adopted

to reduce the overall training time of the model. On the contrary, a synchronous aggregation strategy is adopted to ensure the convergence stability of the global model.

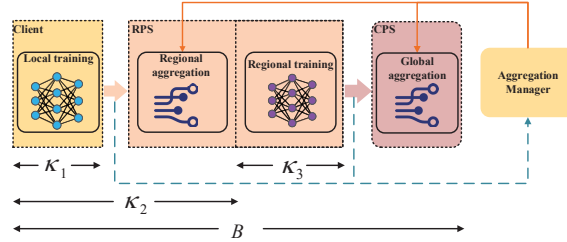


Fig. 3. The training process of the proposed HFL Framework

If we take a three-layer HFL model as an example, the adaptive synchronous and asynchronous aggregation decision will produce four aggregation combinations, i.e., “synchronous-synchronous”, “asynchronous-asynchronous”, “synchronous-asynchronous”, “asynchronous-synchronous”. For the synchronous aggregation mechanism, when $\alpha_c = 1$ or $\alpha_s = 1$, it means that it is necessary to wait for all nodes at the corresponding level to complete training and upload to the CPS (or the RPS) before triggering parameter aggregation. In contrast, if the asynchronous aggregation strategy is adopted, when the number of nodes that complete model training and upload the model reaches the specified threshold, the server (CPS or RPS) can be triggered to perform parameter aggregation, but the server only broadcasts the model aggregated to the nodes contributing to current model aggregation for next training. It is worth noting that asynchronous aggregation strategy must take into account: the server receives the local model parameters of node in the r_c -th round, while the node receives the model aggregated from the server in the r_s -th round, and they are often inconsistent [21], i.e., $\lambda = r_s - r_c \neq 0$. Therefore, this paper defines a parameter v_r^k to measure the staleness of node k in the r -th round, where $v_r^k = \rho^\lambda$, ($0 < \rho < 1$) and ρ is a constant. In particular, if there is no staleness for model update, that is, it is equivalent to a synchronous aggregation mechanism at this time. Therefore, for the asynchronous aggregation strategy, each model parameter $w_{r,H}^k$ owned by node k received by the server will be processed according to Eq.(6) to reduce the impact of nodes with poor staleness on the aggregation model, and then participate in the aggregation process.

$$w_{k,H}^r = v_r^k w_{k,H}^r + (1 - v_r^k) \bar{w}_s^{r^*}, 0 < v_r^k < 1, \quad (6)$$

where $\bar{w}_s^{r^*} = \frac{1}{\Gamma^{r^*}} \sum_k^{\alpha_c K} D_k \cdot w_{k,H}^{r^*}$ is the regional model after the r^* -th round for aggregation. The modification for $w_{k,H}^r$ will be completed based on $\bar{w}_s^{r^*}$ to participate in the model aggregation at $(r^* + 1)$ -th round. If the transmission delay of IoT connection is serious, v_r^k will decrease significantly with the increase of λ , resulting in Eq.(6), the weight $w_{k,H}^r$ of the model participating in the aggregation tends to be close to the aggregation result $\bar{w}_s^{r^*}$ of the previous round, so as to maintain the stability of the global model. The process between CPS and RPS is similar to the above process. Theoretically, the proposed adaptive parameter aggregation algorithm can be extended to one HFL framework with L -layer ($L > 3$) correlation, and correspondingly 2^L kinds of synchronous

and asynchronous parameter aggregation schemes can be formed. The proposed adaptive aggregation method is shown in Algorithm 2.

Algorithm 2 The adaptive parameter aggregation method

Input: parameter received time t_r , parameter sent time t_s

Output: one aggregation strategy, the number of client for aggregation $\alpha_c K$ in RPS, the number of RPS for aggregation $\alpha_s S$ in CPS, the model parameters corrected $w_{k,H}^r$.

- 1: The AM calculates the actual transmission delay Q according to t_r and t_s periodically.
 - 2: **if** Q is larger than T **then**
 - 3: Telling RPS or CPS to use asynchronous solution.
 - 4: $\alpha_c K = K - 1$ in RPS or $\alpha_s S = S - 1$ in CPS.
 - 5: The RPS or the CPS calculates the update delay λ and records it.
 - 6: The RPS corrects related model parameters by Eq.(6) to reduce the impact of staleness, so do CPS.
 - 7: **else**
 - 8: Telling RPS or CPS to use synchronous solution.
 - 9: $\alpha_c K = K$ in RPS or $\alpha_s S = S$ in CPS.
 - 10: **end if**
-

4. The Analysis of Convergence

For ease of convergence analysis, we denote the number of local training rounds and regional training rounds as $r(1 \leq r \leq B\kappa_1\kappa_2)$ and $t(1 \leq t \leq B\kappa_3)$, respectively. We assume that an unbiased estimate of $\nabla f_k(w)$ is $g_j(w, \zeta_k^r)$, i.e., $\nabla f_k(w) = \mathbb{E}_{\zeta_k^r \sim \mathcal{D}_k} g_j(w; \zeta_k^r)$. Also, we assume the loss function is non-convex and smooth. Then we introduce the following assumptions.

Assumption 1:(Lipschitz Gradient). The function f_k, f_s, F_s, f are L -smooth, i.e., $\|\nabla f_k(w) - \nabla f_k(w')\| \leq L \|w - w'\|$, $\|\nabla f_s(w) - \nabla f_s(w')\| \leq L \|w - w'\|$, $\|\nabla F_s(w) - \nabla F_s(w')\| \leq L \|w - w'\|$, $\|\nabla f(w) - \nabla f(w')\| \leq L \|w - w'\|$.

Assumption 2:(Bounded Variance). The divergences satisfy: $\|\nabla F_s(w) - \nabla f(w)\|^2 \leq \epsilon_s^2$, $\|\nabla f_k(w) - \nabla f_s(w)\|^2 \leq \epsilon_k^2$, $\|g_k(w, j) - \nabla f_k(w)\|^2 \leq \sigma^2, \forall s, k, j, w$.

The above assumptions are widely used in non-convex optimization theory [20]. Particularly, the parameter ϵ_s^2 and ϵ_k^2 can quantify the similarity of objective functions. Note $\epsilon_s^2 = 0$ or $\epsilon_k^2 = 0$ corresponds to the IID setting.

Theorem 1: Given the learning rate $\eta \leq \frac{1}{L}$, $1 - 3\eta^2 L^2 \geq 0$ and the optimal global model and regional model are respectively \bar{w}^*, \bar{w}_s^* . When the synchronous aggregation method is adopted, the upper bound of the average regional gradient deviation is given as

follows

$$\begin{aligned} \frac{1}{B\kappa_1\kappa_2} \sum_{r=1}^{B\kappa_1\kappa_2} \mathbb{E} \|\nabla f_s(\bar{\mathbf{w}}_s^{r-1})\|^2 &\leq \frac{2}{B\kappa_1\kappa_2\eta} (\mathbb{E} f_s(\bar{\mathbf{w}}_s^0) - f_s(\bar{\mathbf{w}}_s^*)) + \eta\sigma^2 LK^2 \\ &+ \frac{KL^2\eta^2}{B\kappa_1\kappa_2(1-3\eta^2L^2)\|\Gamma^r\|^2} \sum_{k \in V_K} \|D_k\|^2 (\Phi_1 + \Phi_2) \end{aligned} \quad (7)$$

$$\text{where } \Phi_1 = 2^{B\kappa_1\kappa_2+4}\sigma^2 \left(1 + K \sum_{k \in V_K} \|D_k\|^2\right), \Phi_2 = 3 * 2^{B\kappa_1\kappa_2+3}\epsilon_k^2$$

Proof. Due to the proposition of Lipschitz smooth, the expectation of f_s can be expressed as

$$\begin{aligned} \mathbb{E} f_s(\bar{\mathbf{w}}_s^r) &= \mathbb{E} f_s[\bar{\mathbf{w}}_s^{r-1} - \eta \nabla f_s(\bar{\mathbf{w}}_s^{r-1})] = \mathbb{E} f_s \left[\bar{\mathbf{w}}_s^{r-1} - \eta \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right] \\ &\leq \underbrace{\mathbb{E} f_s(\bar{\mathbf{w}}_s^{r-1}) - \eta \mathbb{E} \left\langle \nabla f_s(\bar{\mathbf{w}}_s^{r-1}), \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\rangle}_{A_1} + \underbrace{\frac{\eta^2 L}{2} \mathbb{E} \left\| \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2}_{A_2} \end{aligned} \quad (8)$$

We further express the bound of A_1 as follows:

$$\begin{aligned} -\eta \mathbb{E} \left\langle \nabla f_s(\bar{\mathbf{w}}_s^{r-1}), \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\rangle &= \frac{\eta}{2} \mathbb{E} \left\| \nabla f_s(\bar{\mathbf{w}}_s^{r-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2 \\ -\frac{\eta}{2} \mathbb{E} \|\nabla f_s(\bar{\mathbf{w}}_s^{r-1})\|^2 - \frac{\eta}{2} \mathbb{E} \left\| \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2 \end{aligned} \quad (9)$$

Then the bound of A_2 can be represented as

$$\begin{aligned} \mathbb{E} \left\| \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-1}) \right\|^2 &= \mathbb{E} \left\| \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2 \\ + \mathbb{E} \left\| \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k (g_k(w_k^{r-1}) - \nabla f_k(w_k^{r-1})) \right\|^2 &\leq \frac{K\sigma^2}{\|\Gamma^r\|^2} \sum_{k \in V_K} D_k^2 + \mathbb{E} \left\| \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2 \end{aligned} \quad (10)$$

By replacing A_1 and A_2 in (8) with (9) and (10) respectively, then we can get

$$\begin{aligned} \mathbb{E} f_s(\bar{\mathbf{w}}_s^r) &\leq \mathbb{E} f_s(\bar{\mathbf{w}}_s^{r-1}) + \frac{\eta^2 L}{2} \left(\frac{K\sigma^2}{\|\Gamma^r\|^2} \sum_{k \in V_K} D_k^2 \right) - \left(\frac{\eta}{2} - \frac{\eta^2 L}{2} \right) \mathbb{E} \left\| \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2 \\ &+ \frac{\eta}{2} \left(\mathbb{E} \left\| \nabla f_s(\bar{\mathbf{w}}_s^{r-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2 - \mathbb{E} \|\nabla f_s(\bar{\mathbf{w}}_s^{r-1})\|^2 \right) \\ &\stackrel{(a)}{\leq} \mathbb{E} f_s(\bar{\mathbf{w}}_s^{r-1}) + \frac{\eta^2 L}{2} \left(\frac{K\sigma^2}{\|\Gamma^r\|^2} \sum_{k \in V_K} D_k^2 \right) \\ &+ \frac{\eta}{2} \left(\underbrace{\mathbb{E} \left\| \nabla f_s(\bar{\mathbf{w}}_s^{r-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2}_{A_3} - \mathbb{E} \|\nabla f_s(\bar{\mathbf{w}}_s^{r-1})\|^2 \right) \end{aligned} \quad (11)$$

Since $\eta \leq \frac{1}{L}$ is assumed, then (a) is obtained. Additionally, the following inequality can be derived based on Assumption 1.

$$\mathbb{E} \left\| \nabla f_s(\bar{\mathbf{w}}_s^{r-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k \nabla f_k(w_k^{r-1}) \right\|^2 \leq \frac{KL^2}{\|\Gamma^r\|^2} \sum_{k \in V_K} D_k^2 \underbrace{\mathbb{E} \|\bar{\mathbf{w}}_s^{r-1} - w_k^{r-1}\|^2}_{A_4} \quad (12)$$

According to SGD and FedAvg, the bound of A_4 can be further represented as

$$\begin{aligned} \mathbb{E} \|\bar{w}_s^{r-1} - w_k^{r-1}\|^2 &= \mathbb{E} \|\bar{w}_s^{r-2} - w_k^{r-2} + \eta g_k(w_k^{r-2}) - \eta \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-2})\|^2 \\ &= \mathbb{E} \left\| (\bar{w}_s^{r-2} - w_k^{r-2}) + \eta g_k(w_k^{r-2}) - \eta \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-2}) \right\|^2 \\ &\stackrel{(b)}{\leq} 2\mathbb{E} \left\| \eta g_k(w_k^{r-2}) - \eta \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-2}) \right\|^2 + 2\mathbb{E} \|\bar{w}_s^{r-2} - w_k^{r-2}\|^2 \end{aligned} \quad (13)$$

The above inequality (b) can be obtained from the mean value inequality. After expanding (13), we obtain

$$\begin{aligned} \mathbb{E} \|\bar{w}_s^{r-1} - w_k^{r-1}\|^2 &\leq 2 \left[\mathbb{E} \left\| \eta g_k(w_k^{r-2}) - \eta \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-2}) \right\|^2 + \mathbb{E} \|\bar{w}_s^{r-2} - w_k^{r-2}\|^2 \right] \\ &\leq \underbrace{\eta^2 \sum_{i=1}^r 2^i \mathbb{E} \left\| g_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-i-1}) \right\|^2}_{A_5} \end{aligned} \quad (14)$$

We continue to drive the bound of A_5 as follows

$$\begin{aligned} &\sum_{i=1}^r 2^i \mathbb{E} \left\| g_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-i-1}) \right\|^2 \\ &\leq \sum_{i=1}^r 2^{i+1} \mathbb{E} \left\| g_k(w_k^{r-i-1}) - \nabla f_k(w_k^{r-i-1}) + \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k f_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k g_k(w_k^{r-i-1}) \right\|^2 \\ &+ \sum_{i=1}^r 2^{i+1} \mathbb{E} \left\| \nabla f_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k f_k(w_k^{r-i-1}) \right\|^2 \end{aligned} \quad (15)$$

We now further bound the term A_5 by mean inequality, we get

$$\begin{aligned} A_5 &\leq \sum_{i=1}^r 2^{i+2} \mathbb{E} \|g_k(w_k^{r-i-1}) - \nabla f_k(w_k^{r-i-1})\|^2 + \sum_{i=1}^r 2^{i+1} \mathbb{E} \left\| \nabla f_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k f_k(w_k^{r-i-1}) \right\|^2 \\ &+ \sum_{i=1}^r \frac{2^{i+2}K}{\|\Gamma^r\|^2} \sum_{k \in V_K} \|D_k\|^2 \mathbb{E} \|(g_k(w_k^{r-i-1}) - f_k(w_k^{r-i-1}))\|^2 \\ &\leq \sigma^2 \left(\sum_{i=1}^r 2^{i+2} + \sum_{i=1}^r 2^{i+2}K \sum_{k \in V_K} \|D_k\|^2 \right) + \underbrace{\sum_{i=1}^r 2^{i+1} \mathbb{E} \left\| f_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k f_k(w_k^{r-i-1}) \right\|^2}_{A_6} \end{aligned} \quad (16)$$

Similarly, the upper bound of A_6 can be derived as follows

$$\sum_{i=1}^r 2^{i+1} \mathbb{E} \left\| \nabla f_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in V_K} D_k f_k(w_k^{r-i-1}) \right\|^2 \leq 3 \sum_{i=1}^r 2^{i+1} \left[\mathbb{E} \|\nabla f_k(w_k^{r-i-1}) - \nabla f_k(\bar{w}_s^{r-i-1})\|^2 + \epsilon_k^2 \right] \quad (17)$$

With the results of (14), (16) and (17), we can obtain (18),

$$\mathbb{E} \|\bar{w}_s^{r-1} - w_k^{r-1}\|^2 \leq 2^{r+3} \eta^2 \sigma^2 \left(1 + K \sum_{k \in V_K} \|D_k\|^2 \right) + 3\eta^2 L^2 \sum_{i=1}^r 2^{i+1} \mathbb{E} \left(\|w_k^{r-i-1} - \bar{w}_s^{r-i-1}\|^2 \right) + 3\eta^2 \epsilon_k^2 2^{r+2} \quad (18)$$

By averaging the results of $B\kappa_1\kappa_2$ trainings, we can get

$$\begin{aligned} \frac{1}{B\kappa_1\kappa_2} \sum_{r=1}^{B\kappa_1\kappa_2} \mathbb{E} \|\bar{w}_s^{r-1} - w_k^{r-1}\|^2 &\leq \frac{2^{B\kappa_1\kappa_2+4}}{B\kappa_1\kappa_2} \eta^2 \sigma^2 \left(1 + K \sum_{k \in V_K} \|D_k\|^2 \right) + \frac{3^* 2^{B\kappa_1\kappa_2+3} \eta^2 \epsilon_k^2}{B\kappa_1\kappa_2} \\ &+ \frac{3\eta^2 L^2}{B\kappa_1\kappa_2} \sum_{r=1}^{B\kappa_1\kappa_2} \sum_{i=1}^r 2^{i+1} \mathbb{E} \left(\|w_k^{r-i-1} - \bar{w}_s^{r-i-1}\|^2 \right) \leq \Phi_1 + \Phi_2, 1 > 1 - 3\eta^2 L^2 \geq 0 \end{aligned} \quad (19)$$

Through (11), (12), (18) and (19), we can obtain **Theorem 1**, which completes the proof.

$$\begin{aligned} \frac{1}{B\kappa_1\kappa_2} \sum_{r=1}^{B\kappa_1\kappa_2} \mathbb{E} \|\nabla f_s(\bar{\mathbf{w}}_s^{r-1})\|^2 &\leq \eta \sigma^2 L K^2 + \frac{2}{B\kappa_1\kappa_2 \eta} (\mathbb{E} f_s(\bar{\mathbf{w}}_s^0) - f_s(\bar{\mathbf{w}}_s^*)) \\ &+ \frac{K L^2 \eta^2}{B\kappa_1\kappa_2 (1 - 3\eta^2 L^2) \|\Gamma^r\|^2} \sum_{k \in V_K} \|D_k\|^2 (\Phi_1 + \Phi_2) \end{aligned} \quad (20)$$

where $\eta \leq \frac{1}{L}$, $1 - 3\eta^2 L^2 \geq 0$.

Following **Theorem 1**, the similar result can be obtained for the upper bound of the average global gradient deviation and the upper bound of the average regional gradient deviation when synchronous FL is used.

Furthermore, if asynchronous aggregation method is adopted, **Theorem 2** can be obtained under the premise of the above assumptions.

Theorem 2: Given the learning rate $\eta \leq \frac{1}{L}$, $1 - 6\eta^2 L^2 \geq 0$. The upper bound of the average regional gradient deviation is given as follows,

$$\begin{aligned} \frac{1}{B\kappa_1\kappa_2} \sum_{r=1}^{B\kappa_1\kappa_2} \mathbb{E} \|\nabla f_s(\bar{\mathbf{w}}_s^{r-1})\|^2 &\leq \frac{2}{\eta} (\mathbb{E} f_s(\bar{\mathbf{w}}_s^0) - \mathbb{E} f_s(\bar{\mathbf{w}}_s^r)) + \eta L \sigma^2 \alpha_c^2 K^2 \\ &+ \alpha_c K L^2 \sum_{k \in M_{K,r}} \frac{2\eta^2 \sigma^2 (1 + \alpha_c^2 K^2) + 6\eta^2 L^2 \epsilon_k^2}{B\kappa_1\kappa_2 (1 - 6\eta^2 L^2) (1 - 2v_k^{\lambda_k})} \left(2B\kappa_1\kappa_2 v_k^{\lambda_k} - \frac{(2v_k^{\lambda_k})^2 \left(1 - (2v_k^{\lambda_k})^{B\kappa_1\kappa_2+1} \right)}{1 - 2v_k^{\lambda_k}} \right) \end{aligned} \quad (21)$$

Proof. Since some proof procedure can be found in **Theorem 1**, we only show the differences from **Theorem 1**: Firstly, we assume that in the $r^t h$ round, the server receives the model from the set $M_{K,r}$ which is denoted as the clients sending local model to the

server. We have

$$\begin{aligned} \mathbb{E}f_s(\bar{\mathbf{w}}_s^r) &\leq \mathbb{E}_s(\bar{\mathbf{w}}_s^{r-1}) + \frac{\eta^2 L}{2} \left(\alpha_c K \sigma^2 \sum_{k \in M_{K,r}} \left\| \frac{D_k}{\Gamma^r} \right\|^2 \right) \\ &+ \frac{\eta}{2} \left(\underbrace{\mathbb{E} \left\| \nabla f_s(\bar{\mathbf{w}}_s^{r-1}) - \sum_{k \in M_{K,r}} \frac{1}{\Gamma^r} D_k \nabla f_k(w_k^{r-1}) \right\|^2}_{A_7} - \mathbb{E} \|f_s(\bar{\mathbf{w}}_s^{r-1})\|^2 \right) \end{aligned} \quad (22)$$

Through the use of smoothness, the upper bound of A_7 can be expressed as follows

$$\begin{aligned} \mathbb{E} \left\| \nabla f_s(\bar{\mathbf{w}}_s^{r-1}) - \sum_{k \in M_{K,r}} \frac{1}{\Gamma^r} D_k \nabla f_k(w_k^{r-1}) \right\|^2 &\leq \\ \alpha_c K L^2 \sum_{k \in M_{K,r}} \left\| \frac{D_k}{\Gamma^r} \right\|^2 \underbrace{\mathbb{E} \|(\bar{\mathbf{w}}_s^{r-1} - w_k^{r-1})\|^2}_{A_8} \end{aligned} \quad (23)$$

Similar to the method adopted in (14) and (16), we can further obtain

$$\begin{aligned} \mathbb{E} \|\bar{w}_s^{r-1} - w_k^{r-1}\|^2 &\leq 2 \|v_k^{\lambda_k}\|^2 \left(\mathbb{E} \|\bar{w}_s^{r-2} - w_k^{r-2}\|^2 + \mathbb{E} \left\| \eta g_k(w_k^{r-2}) - \eta \sum_{k \in M_{K,r}} \frac{1}{\Gamma^r} D_k g_k(w_k^{r-2}) \right\|^2 \right) \\ &\leq \underbrace{\eta^2 \sum_{i=1}^r 2^i \|v_k^{\lambda_k}\|^{2i} \mathbb{E} \left\| g_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in M_{K,r}} D_k g_k(w_k^{r-i-1}) \right\|^2}_{A_9} \end{aligned} \quad (24)$$

Similarly, we can get

$$\begin{aligned} A_9 &\leq \sigma^2 \left(\sum_{i=1}^r 2^{i+1} \|v_k^{\lambda_k}\|^{2i} + \sum_{i=1}^r 2^{i+1} \|v_k^{\lambda_k}\|^{2i} \alpha_c K \sum_{k \in M_{K,r}} \left\| \frac{D_k}{\Gamma^r} \right\|^2 \right) \\ &+ \underbrace{\sum_{i=1}^r 2^{i+1} \|v_k^{\lambda_k}\|^{2i} \mathbb{E} \left\| \nabla f_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r} \sum_{k \in M_{K,r}} D_k f_k(w_k^{r-i-1}) \right\|^2}_{A_{10}} \end{aligned} \quad (25)$$

Finally, based on (22)-(25), the conclusion of **Theorem 2** can be obtained.

5. Performance Evaluation

In this section, we experimentally evaluate the effectiveness and performance of the proposed HFL framework. We firstly describe experimental settings and evaluation metric. Then, we conduct two experiments to illustrate the influence of key parameters on model convergence and demonstrate the efficiency of the proposed model, respectively.

Table 2. The statistics of datasets in the experiment

Area	Nodes	Min_O(divergence)	Max_I(divergence)	Min_I(divergence)
RPS 1	6	0.12412	0.02429	0.01558
RPS 2	4	0.26799	0.07153	0.01426
RPS 3	7	0.16471	0.07686	0.03633
RPS 4	4	0.20628	0.09204	0.03170
RPS 5	5	0.22823	0.09762	0.02344

5.1. Experimental Setup

The HFL framework proposed in this paper is formulated on the basis of analyzing the hierarchical relationship between water supply regions and the historical water supply datasets in Chongqing, China. The datasets used for experiments include the data of daily water supply in different water supply regions from July 2019 to July 2021. We firstly performed noise reduction and smoothing on the data, which is to reduce the impact of noisy data on subsequent model training to a certain extent. Also, the datasets used for training not only includes water supply data at different time points in different water supply areas, but also includes holiday information and environmental data in each water supply area, such as: air temperature, air humidity, rainfall, wind speed and direction, which are taken into account. At the same time, due to the fact that the IoT-based data collection terminals in various regions may lose packets or be interfered by communication during the actual network transmission process, there is a small amount of data missing in the 2-year datasets. According to the relationship between the water supply regions and the HFL structure shown in Fig.2, the entire water supply area includes one CPS for training and aggregating global model, and then the whole region is divided into 5 sub-regions, each sub-region includes one RPS and multiple clients, RPS is responsible for model aggregation in the sub-region, and client is responsible for its local model training. It is worth noting that, we refer to the method adopted in [22, 23] to further divide the original data set into multiple sub-datasets, and increase the scale of participating nodes while ensuring that the original distribution of the data is not changed. The deployment of CPS, RPSs and clients is shown in Fig.4. Furthermore, we use the divergence to measure the similarity between different regions. If the larger the divergence, the smaller the similarity. The data distribution for each region is combined in pairs, and then the corresponding divergence values are then calculated. The statistics datasets and the results on divergence between them are summarized in Table 2. Specifically, *Nodes* represents the number of clients included in the area covered by an RPS. In addition, *Min_O* represents the minimum divergence between the RPS and the clients outside the RPS area. *Max_I* represents the maximum divergence between the clients and RPS in the same RPS area, and *Min_I* indicates the minimum divergence in a specific RPS area. It can be seen from Table 2 that the data similarity in the same region is high, while the data similarity is low outside the area. Finally, according to the ratio of 4:1, each dataset is divided into training dataset and test dataset. In addition, in order to intuitively measure the accuracy of the model, we introduce a commonly used indicator explained variance score(EVC). When the indicator is close to 1, it means that the fitting effect of the model will be better, other-

wise the opposite. The experiments are deployed on a deep learning workstation equipped with NVIDIA GeForce RTX 3090 GPU, and the HFL model proposed is built based on TensorFlow [24].

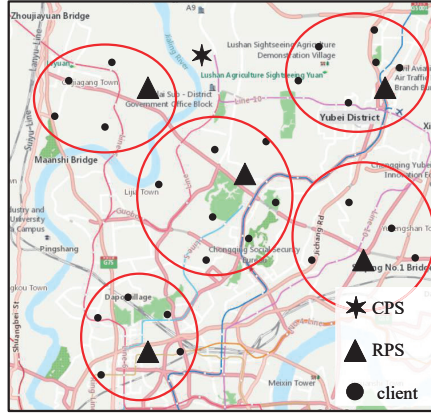


Fig. 4. Deployment of CPS, RPSs and clients in an entire water supply area

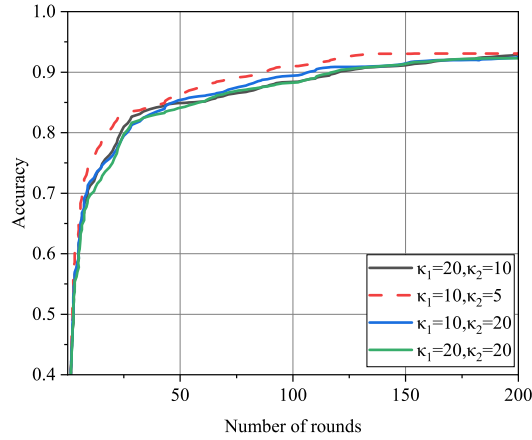


Fig. 5. Convergence under different κ_1 and κ_2

5.2. The Analysis of Convergence

During the training process of the client and the RPS with K_1 and K_2 rounds respectively, the convergence of the global model can be compared ($K_1 = B\kappa_1\kappa_2$, $K_2 = B\kappa_3$) by adjusting the value of κ_1 and κ_2 . When the current number of training rounds r satisfies $r|\kappa_1\kappa_2 = 0$, RPS will aggregate local models belonged to the clients in the covered sub-region and obtain the regional model \bar{w}_s^r , and further train the regional model of the RPS. The initial model parameter of the RPS is \bar{w}_s^r , that is, $w_0^s = \bar{w}_s^r$. Finally, the regional

model of the RPS participates in the global model aggregation at the CPS. It can be seen that the convergence of the global model is directly affected by the RPS from Algorithm 1, while the client indirectly affects the convergence of the global model by affecting the initial model weight of the RPS. Additionally, we assume the minimum acceptable delay $T \rightarrow +\infty$ at this time. It can be also seen from **Theorem1** and **Theorem2** that the smaller the values of κ_1 and κ_2 , the better the convergence of the model, which is not only applicable to the regional model but also the global model. As shown in Fig.5, the global model convergence is evaluated under different combinations of κ_1 and κ_2 . It can be seen that the global model converges best when $\kappa_1 = 10$ and $\kappa_2 = 5$, mainly because the values of these two parameters are the smallest. In contrast, when $\kappa_1 = 20$ and $\kappa_2 = 20$, the model convergence performance is the worst.

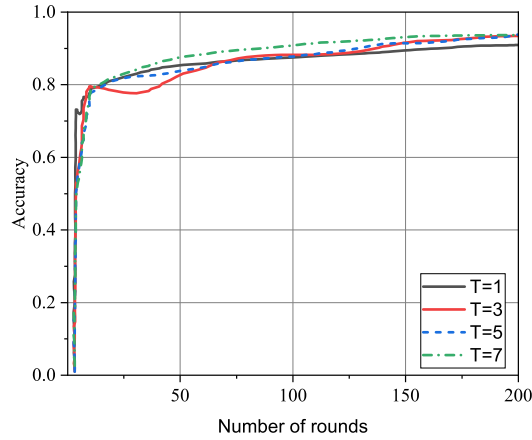


Fig. 6. The prediction accuracy of the model under different values of T

5.3. Performance Under Different Network Delay

In order to observe the impact of the heterogeneity of the devices and the data on model performance, we first construct the ideal case for network connection. Under the conditions of $\kappa_1 = 10$ and $\kappa_2 = 5$, we design four different T to compare the influence of system heterogeneity on the model convergence. The values of T are 1, 3, 5, and 7, respectively. The number of training rounds for both the RPS and the client is 200, that is, $B\kappa_1\kappa_2 = B\kappa_3 = 200$. The results on model prediction accuracy with different T are shown in Fig.6. It can be found that due to the difference in computing capability of each node participating in the HFL, different convergence trends of the global model will appear. In particular, when T is set to a small value, it can simulate the situation that the node cannot tolerate the existing delay being large during the parameter transmission process. In Fig.6, when $T = 1$, nodes will select asynchronous aggregation for model training more time. In addition, since the number of nodes participating in asynchronous aggregation during a certain round of training is less than the total number of nodes, in the early stage of model training, the accuracy curve corresponding to $T = 1$ fluctuates relatively significantly.

Table 3. Model prediction accuracy and switching times after 200 rounds of training under different T

	HFL(T=1)	HFL(T=3)	HFL(T=5)	HFL(T=7)
Running Time(sec)	7869.408	11079.721	12547.851	13093.137
EVC	0.909	0.934	0.937	0.936
Async_Freq	577	50	8	0
Sync_Freq	464	915	973	987

Table 4. Prediction accuracy of RPS under different T

	RPS 1	RPS 2	RPS 3	RPS 4	RPS 5
	EVC				
T = 1	0.827	0.931	0.846	0.887	0.918
T = 3	0.825	0.930	0.842	0.892	0.921
T = 5	0.830	0.934	0.844	0.892	0.923
T = 7	0.831	0.934	0.844	0.890	0.921

Under the above ideal network connection, we record running time, EVC for the global model, and switching times of synchronous aggregation method ($Sync_Freq$) and asynchronous aggregation method ($Async_Freq$) corresponding to four sets of T . The results are shown in Table 3. It can be observed that $Sync_Freq$ and $Async_Freq$ computed by CPS with $T = 1$ are relatively close. Indicating that in ideal circumstances, the delay caused by other influencing factors (such as heterogeneous computing power) averages close to 1. With the increase of T , $Sync_Freq$ for CPS will be also increased. Correspondingly, the time to complete the entire training task will be also increased, because the synchronous aggregation needs to wait for all participants to complete their local task. In particular, when $T = 7$, then $Async_Freq = 0$, it means that arbitrary delay can be tolerated, so asynchronous aggregation will not be adopted, that is, the HFL is equivalent to the traditional synchronous FL. Conversely, the HFL is more sensitive to delay when $T \rightarrow 1$. More extremely, the HFL is similar to traditional asynchronous FL when $T \rightarrow 0$. Similarly, RPS also has its $Sync_Freq$ and $Async_Freq$, as shown in Fig.7, where R_i ($Async$) and R_i ($Sync$) are represented as $Async_Freq$ and $Sync_Freq$ of RPS i respectively. The global model after convergence with different T in Fig.6 are applied to the test datasets of all RPSs respectively, then corresponding prediction accuracy (i.e, EVC) can be calculated, as shown in Table 4. It can be found that the difference in terms of test accuracy on the same dataset for different T is almost small.

However, most of the nodes actually participating in FL are mobile devices with limited network bandwidth resources, so there will be a certain communication delay in parameter transmission process, and the delay for uploading parameters is generally larger than that for downloading parameters [25].

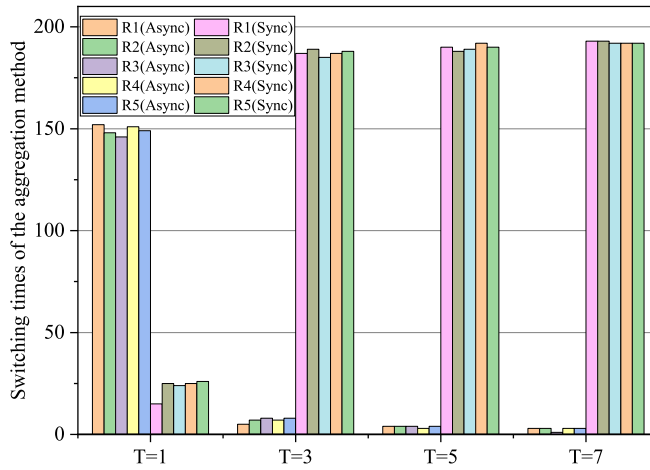


Fig. 7. Switching times of the aggregation method (async or sync) for RPS(1,2,3,4,5) under different T

The specific communication delay is measured as follow [26],

$$T_0 = \frac{\Theta_i}{b_i^r \log \left(1 + o_i |G_i^r|^2 / \psi \right)} \tag{26}$$

where b_i^r is the allocated bandwidth for node i at round r from the total bandwidth B , i.e. $\sum_i b_i^r = B$, o_i is the transmit power of node i , G_i^r is the average channel gain between node i and its upper server, ψ is the background noise and Θ_i is the size of local model of node i . To simplify the calculation, we assume $B = 20MHz$, $\psi = 10^{-19}$, $o_i = 20dBm$ and $G_i^r = 10$ [27].

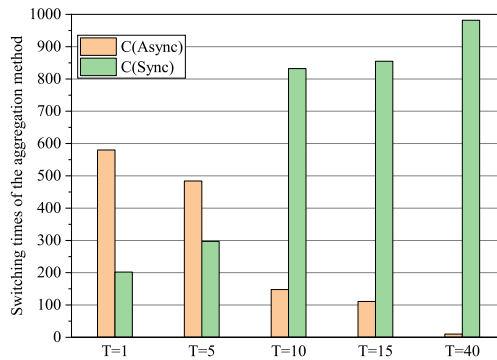


Fig. 8. Switching times of the aggregation method (Async or Sync) for CPS under different T

We further set five different T ($T = 1, T = 5, T = 10, T = 15, T = 40$) to verify the validity of the HFL. After 200 rounds local training on clients, multiple *Async.Freq*

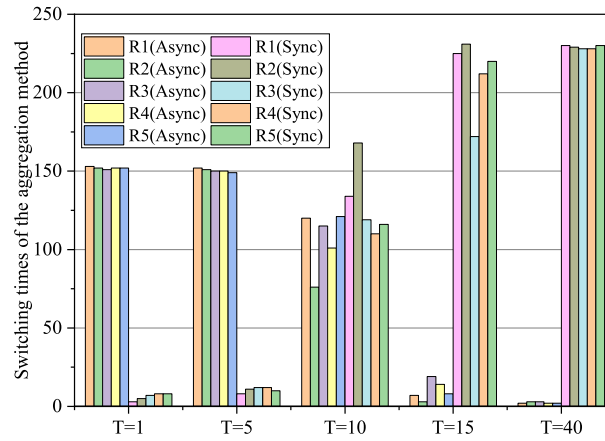


Fig. 9. Switching times of the aggregation method (Async or Sync) for RPS(1,2,3,4,5) under different T

and $Sync_Freq$ can be obtained by CPS and RPS respectively, as shown in Fig.8 and Fig.9. And “C(Async)” and “C(Sync)” respectively represent asynchronous(Async_Freq) or synchronous(Sync_Freq) aggregation performed by CPS. When T becomes larger, $Sync_Freq$ will be increased synchronously, but $Async_Freq$ will be the opposite. In particular, if $1 < T < 40$, CPS will use both synchronous and asynchronous aggregation method during the entire training process, that is, the operation for adopting synchronous or asynchronous aggregation is a dynamic procedure, instead of statically adopting either one approach among them. Similarly, with the increase of T , RPS will also undergo similar synchronous and asynchronous aggregation strategy adjustment during the whole training process. Particularly, $Async_Freq$ is close to 0 at $T = 40$ in Fig.9, indicating that the sum of communication and calculation delay is closed to 40, and the overall training process of the HFL will be affected hardly by the delay.

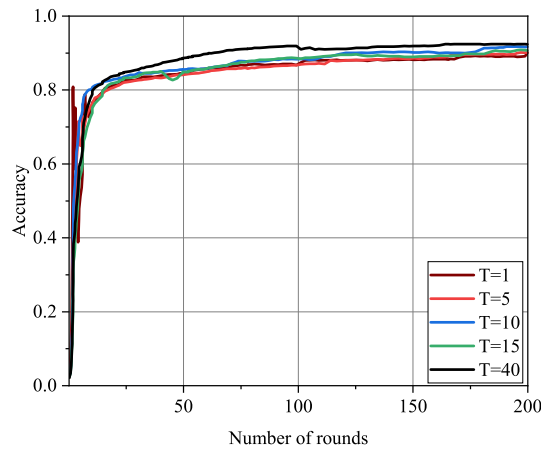


Fig. 10. The prediction accuracy of the model under different T

The prediction accuracy curves of global model corresponding to the above five T are shown in Fig.10. When T is increased, it will reduce the impact of communication delay on the HFL and increase $Sync_Freq$, which makes the model convergence more stable and faster, such as the accuracy curve with $T = 40$, but the time for training model will be also increased. However, the gap of prediction accuracy among them after model convergence is between 0.7% and 2.4%.

Furthermore, the framework proposed in this paper (denoted by Adaptive-HFL) is also compared with some representative models (i.e. ECHFL[10], ADFO[20], AHFL[26]) to demonstrate the its effectiveness. Among them, ECHFL and AHFL are both belong to hierarchical federated learning models, and ADFO is a federated version of adaptive optimizer. When $T = 10$, the comparison result about prediction accuracy is shown in Fig.11, due to the small number of local models of a single server in the hierarchical-based learning structure, there will be certain fluctuations for training, and the curve corresponding to ADFO is relatively smoother because ADFO is designed based on “server-client” network service architecture and adaptive optimization algorithm. However, in general, the hierarchical structure can achieve better prediction effect than ADFO after convergence. At the same time, since the hierarchical correlation and adaptive parameter aggregation scheme are considered in Adaptive-HFL, and more constraints are considered by AHFL model, so the higher prediction accuracy can be achieved. In addition, the accuracy of Adaptive-HFL is about 3.4% higher than that of ADFO. Finally, under the same rounds of training, the results of time required for the four models are shown in Fig.12. It can be seen that the time required for Adaptive-HFL has decreased by 12.6%, 8.4%, and 9.8% than ADFO, AHFL and ECHFL, respectively. Among them, because Adaptive-HFL can adjust the aggregation method between the different layers according to factors such as network communication delay, Adaptive-HFL realizes the model accuracy close to AHFL, but saves 8.4% of the training time required for convergence. This results show that Adaptive-HFL has a significant advantage in convergence performance compared to other three models.

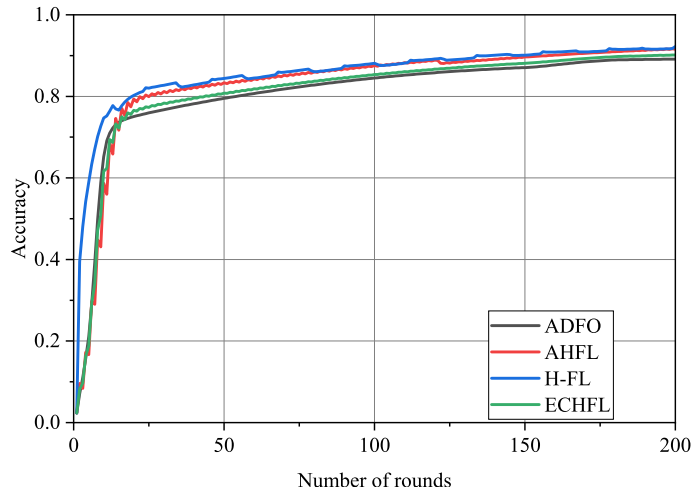


Fig. 11. The accuracy of different algorithms

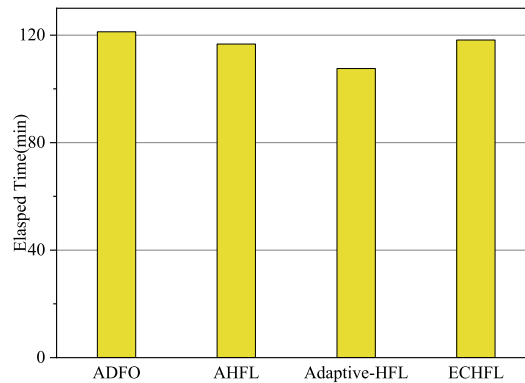


Fig. 12. The comparison of time required for model convergence

6. Conclusion

This paper first proposes a hierarchical federated learning framework, which realizes the joint learning with hierarchical associations between multiple data holders. Meanwhile, we propose a model parameter aggregation algorithm for selecting dynamically asynchronous aggregation/synchronous aggregation to improve the efficiency for training global model. This paper not only conducts a theoretical analysis on the convergence of the proposed model, but also comprehensively evaluates the effectiveness and performance of the proposed framework by taking the prediction of water demand in the urban multi-regional water supply scenario as an experiment. In our future work, we will investigate the matching problem between the efficiency of model training and node selection based on the HFL proposed framework.

Acknowledgments. This work is supported in part by Scientific and Technological innovation project of scientific research institutions of Chongqing (No.cstc2021jxjl20010), in part by Chongqing Technology Innovation and Application Development Key Project under Grant 2022TIAD-KPX0048 and Grant 2022TIAD-KPX0053, in part by Banan District Scientific and Technological Achievements Transformation and Industrialization Project, in part by Graduate Student Innovation Program of Chongqing University of Technology under Grant (No.gzlxc20222061).

References

1. K. M. Ahmed, A. Imteaj, M. H. Amini, Federated deep learning for heterogeneous edge computing, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1146–1152.
2. M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, H. V. Poor, Distributed learning in wireless networks: Recent progress and future challenges, *IEEE Journal on Selected Areas in Communications* 39 (12) (2021) 3579–3605.
3. L. Chettri, R. Bera, A comprehensive survey on internet of things (iot) toward 5g wireless systems, *IEEE Internet of Things Journal* 7 (1) (2020) 16–32.

4. W. Y. B. Lim, Z. Xiong, J. Kang, D. Niyato, C. Leung, C. Miao, X. Shen, When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries, *IEEE Transactions on Industrial Informatics* 18 (1) (2022) 457–466.
5. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.
6. A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, *arXiv preprint arXiv:1811.03604* (2018).
7. J. Domingo-Ferrer, A. Blanco-Justicia, J. Manjón, D. Sánchez, Secure and privacy-preserving federated learning via co-utility, *IEEE Internet of Things Journal* 9 (5) (2022) 3988–4000.
8. X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of fedavg on non-iid data, in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net*, 2020.
9. C. Xie, S. Koyejo, I. Gupta, Asynchronous federated optimization, *CoRR abs/1903.03934* (2019). *arXiv:1903.03934*.
10. L. Liu, J. Zhang, S. Song, K. B. Letaief, Client-edge-cloud hierarchical federated learning, in: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
11. Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, Y. Zhao, Resource-efficient federated learning with hierarchical aggregation in edge computing, in: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.
12. W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, C. Miao, Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning, *IEEE Transactions on Parallel and Distributed Systems* 33 (3) (2022) 536–550.
13. A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, V. Smith, On the convergence of federated optimization in heterogeneous networks, *CoRR abs/1812.06127* (2018). *arXiv:1812.06127*.
14. B. Luo, X. Li, S. Wang, J. Huang, L. Tassiulas, Cost-effective federated learning design, in: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.
15. Z. Chen, W. Liao, K. Hua, C. Lu, W. Yu, Towards asynchronous federated learning for heterogeneous edge-powered internet of things, *Digital Communications and Networks* 7 (3) (2021) 317–326.
16. H. Zhu, M. Yang, J. Kuang, H. Qian, Y. Zhou, Client selection for asynchronous federated learning with fairness consideration, in: *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2022, pp. 800–805.
17. C. Chen, H. Xu, W. Wang, B. Li, B. Li, L. Chen, G. Zhang, Communication-efficient federated learning with adaptive parameter freezing, in: *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 2021, pp. 1–11.
18. J. Liu, H. Xu, L. Wang, Y. Xu, C. Qian, J. Huang, H. Huang, Adaptive asynchronous federated learning in resource-constrained edge computing, *IEEE Transactions on Mobile Computing* (2021) 1–1.
19. J. Jin, J. Ren, Y. Zhou, L. Lyu, J. Liu, D. Dou, Accelerated federated learning with decoupled adaptive optimization, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 10298–10322.
20. S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, H. B. McMahan, Adaptive federated optimization, in: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net*, 2021.
21. Y. Zhao, X. Gong, Quality-aware distributed computation for cost-effective non-convex and asynchronous wireless federated learning, in: *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, IEEE, 2021, pp. 1–8.

22. C. Zhou, J. Liu, J. Jia, J. Zhou, Y. Zhou, H. Dai, D. Dou, Efficient device scheduling with multi-job federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 9971–9979.
23. Q. Li, Y. Diao, Q. Chen, B. He, Federated learning on non-iid data silos: An experimental study, in: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 965–978.
24. Tensorflow, Tensorflow, <https://www.tensorflow.org> (2022).
25. S. Q. Zhang, J. Lin, Q. Zhang, A multi-agent reinforcement learning approach for efficient client selection in federated learning, Proceedings of the AAAI Conference on Artificial Intelligence 36 (8) (2022) 9091–9099. doi:10.1609/aaai.v36i8.20894.
26. B. Xu, W. Xia, W. Wen, P. Liu, H. Zhao, H. Zhu, Adaptive hierarchical federated learning over wireless networks, IEEE Transactions on Vehicular Technology 71 (2) (2022) 2070–2083.
27. W. Shi, S. Zhou, Z. Niu, M. Jiang, L. Geng, Joint device scheduling and resource allocation for latency constrained wireless federated learning, IEEE Transactions on Wireless Communications 20 (1) (2021) 453–467.

Zhuo Chen received Ph.D. degree in communication and information systems from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He is currently an Associate Professor with the College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. His current research interests include distributed machine learning and IoT application.

Chuan Zhou received B.E. degree in Internet-of-Things engineering from Chongqing University of Education, Chongqing, China, in 2020. He currently is pursuing his master's degree in Computer science and technology at the College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. His research interests include Federated Learning and big data.

Yang Zhou received Ph.D. degree in the College of Computing at the Georgia Institute of Technology. He is currently an Assistant Professor in the Department of Computer Science and Software Engineering at the Auburn University. His current research interests include trustworthy machine learning, parallel, distributed, and Federated Learning.

Received: September 30, 2022; Accepted: March 10, 2023.

Point of Interest Coverage with Distributed Multi-Unmanned Aerial Vehicles on Dynamic Environment

Fatih Aydemir^{1,2} and Aydin Cetin²

¹ STM Defence Technologies Engineering and Trade. Inc.
06530 Ankara Turkiye
fatih.aydemir@stm.com.tr

² Gazi University, Faculty of Technology, Department of Computer Engineering
06560 Ankara Turkiye
acetin@gazi.edu.tr

Abstract. Mobile agents, which learn to optimize a task in real time, can adapt to dynamic environments and find the optimum locations with the navigation mechanism that includes a motion model. In this study, it is aimed to effectively cover points of interest (PoI) in a dynamic environment by modeling a group of unmanned aerial vehicles (UAVs) on the basis of a learning multi-agent system. Agents create an abstract rectangular plane containing the area to be covered, and then decompose the area into grids. An agent learns to locate on a center of grid that are closest to it, which has the largest number of PoIs to plan its path. This planning helps to achieve a high fairness index by reducing the number of common PoIs covered. The proposed method has been tested in a simulation environment and the results are presented by comparing with similar studies. The results show that the proposed method outperforms existing similar studies and is suitable for area coverage applications.

Keywords: unmanned aerial vehicle, multi-agent system, reinforcement learning, dynamic-area coverage, grid decomposition.

1. Introduction

1.1. Background, Definition, and Motivation

Point of interest coverage (PoI) is a problem that has a huge scope, where a group of agents is tasked with exploring and mapping [16] an unknown environment while also monitoring [9] as many PoI as possible. The aim of the PoI coverage is to ensure that a group of agents visit and monitor predefined locations, known as PoIs. These agents, which could be robots, drones, or even people, should move across a dynamic environment in order to cover a set of PoI. Topics such as how much of the PoI is monitored, how fast the process is managed, and how high the fault tolerance is, determine the quality of coverage. In applications where non-mobile agents are used, the agent remains stable after initial setup. This makes it difficult to adapt to changes in the environment and reduces the fault tolerance of the coverage process. It is needed the mobile agent to be designed with complex control modules if a single mobile agent is used. However, the same capability can be achieved with the cooperation of a group of simple mobile agents.

A group of mobile agents can be designed as a multi-agent system (MAS) that handle coverage problem in a coordinated, interactive, dependent, or independent manner [5]. These systems have defined a collection of agents that produce a common goal-oriented behavior called “collective intelligence” [30]. Methods developed with collective intelligence which are based on centralized algorithms can be computationally expensive and inflexible in dynamic environments. There has been growing curiosity about using distributed multi-agent reinforcement learning (MARL) to deal with the PoI coverage problem [32]. In MARL, each agent learns to make decisions independently, based on its local observations and interactions with the environment. This helps to create more scalable and adaptable solutions since the agents learn to coordinate and collaborate without the need for a centralized controller.

One key challenge in using MARL for PoI coverage is the dynamic nature of the environment [3]. MARL algorithms can be developed to be responsive, meaning that they can adjust their behavior based on the current state of the environment. The requirement for effective coordination among the agents presents another challenge when employing MARL for PoI coverage. For the agents to be able to coordinate their actions in a way that is both effective and efficient, it is necessary to design sophisticated communication and coordination algorithms. Although there are some challenges, MARL can enable to design of effective approaches for solving the PoI coverage problem. By using methods based on MARL, a group of agents can learn to work together and coordinate in order to cover the maximum number of PoIs in an area, even when the environment is changing and unpredictable.

1.2. Literature Review

Area coverage-based studies have been carried out by deployment strategies in order to solve problems such as maximum PoI coverage and maximum area surveillance. Mozaf-fari et al. [21] have investigated how effectively multiple unmanned aerial vehicles (UAVs) could be deployed to serve as wireless base stations and cover ground users. In this context, the authors proposed a 3-dimensional (3-D) deployment strategy based on circle packing theory to maximize area coverage while increasing the lifetime of the network. In [33], an algorithm based on the Artificial Bee Colony (ABC) algorithm has been applied for increasing area coverage and quality of connectivity between sensor nodes. Experimental results were compared with random distribution and Genetic Algorithm. According to the results, the proposed algorithm has a longer lifetime of communication and a higher coverage rate than the others. Gupta et al. [10] studied sensor node placement using a genetic algorithm-based approach that all target points k-coverage and sensor nodes m-connected. Kalantari et al. [14] tried to find the number of UAV base stations needed to cover an area by considering the system requirements and constraints of the environment. The researchers proposed a heuristic algorithm based on Particle Swarm Optimization to locate base stations in a 3-D plane [25]. Njoya et al. [22] proposed a hybrid method that aims to spread sensor nodes effectively considering connectivity between sensor nodes to cover the target area. Jagtap and Gomathi [12] have studied the Euclidean Spanning Tree Model (ECST) and ECST-adaptive velocity added (VABC) (ECST-AVABC) methods for solving the mobile sensor deployment problem, which includes target coverage and network connectivity. Additionally, the researchers designed an AVABC optimization

method by acquiring the least amount of movement of mobile sensors through the network. In [26], a novel method for increasing the WSN's coverage and longevity while preserving network connectivity was presented. The proposed method GCVD, which is based on the Voronoi diagram (VD) and the geometric center (GC), allows mobile sensors to only move within a predefined distance. This movement strategy helps maximize area coverage with minimal energy consumption while guaranteeing the connectivity of nodes. In [15], a deployment strategy using IoT devices association policy has been investigated for fixed-wing UAVs while gathering data from IoT devices. It was assumed that each UAV follows a circular path over devices associated with it. The problem of device association was designed as a Multi-Knapsack Problem that takes into account the service capacities of UAVs as well as the load requests of devices. Ganganath et al. [7] have studied two different methods based on anti-flocking for dynamic coverage with mobile sensor networks in environments with and without obstacles.

The studies that designed with collaborative behavior strategies have been studied, where it was recommended an online coverage algorithm. In [31], the area coverage problem was handled using a target assignment with path planning approach, which has online execution process. Liu et al. [18] proposed a learning method called DRL-EC3 in order to perform the UAV positioning and orienting process with less energy consumption. This method was designed based on the deep deterministic policy gradient (DDPG) [17] algorithm that single actor-critic network was used in the training phase. An improved version of the DRL-EC3 model was presented to increase fault tolerance for dynamic changes in the environment [19]. In this improved model, each UAV had its own actor-critic network that helps find out convenient actions to maximize area coverage with minimum energy consumption. In the target area, it was also aimed to establish a network between the UAVs performing distributed manner. Nemer et. al. [23] designed an actor-critic method on the bases of a state-based potential game (SBG-AC) that guides UAVs in moving from the initial position to the final position to obtain maximum PoI coverage with minimum energy consumption. The method has a similar motion model with the improved DRL-EC3 both methods use the distributed version of DDPG. Cabreira et al. [2] proposed a coverage path planning algorithm based on the grid-based approach presented in [29]. The original cost function was replaced with an energy-cost function to minimize energy consumption while mapping an area with UAVs. Aydemir and Cetin [1] proposed a method to maximize the covered area in a dynamic environment using multi-UAVs. Agents modeled with deep reinforcement learning techniques produced a model-free policy using a central module in the learning phase, but the central module was not used during the execution phase. The proposed method built as distributed system aims to place the active agents at the most appropriate points in the target area by creating a graph.

In agent-based methods, mobile agents focus on improving the coverage process by spreading over the area. The main difference between them is how the motion model is designed. In systems managed by a central controller, the process is managed from a single point, and errors that may occur in the centralized controller directly affect the entire system. Heuristic methods can optimize the dynamic range coverage process, but agents do not include the actions of other agents in the optimization process. With this approach, it may take a long time to converge to the optimal result. In methods using anti-flocking, agents evaluate the instantaneous situation rather than the entire coverage process. For this reason, the most appropriate result of the instant situation is obtained.

1.3. Contribution of the Paper

In the dynamic range coverage process, UAVs equipped with sensors can be used to achieve low cost with high flexibility [31]. In this paper, a multi-agent deep RL-based method in which each UAV is represented as an agent is proposed to cover the maximum number of PoI in the target area. RL is a reward-driven machine learning approach that an agent learns in an interactive environment through its experiences. An RL agent interacts with the environment and receives positive or negative reward points for each action. Classical RL methods are individual learning processes in that an agent learns to change its behavior to maximize the reward PoInts it receives. In multi-agent RL (MARL), the actions of all agents are evaluated for the common goal of the group. Collective behavior refers to a system that provides maximum coverage when evaluated for the agent group. When this behavior is evaluated for an agent, it represents a subsystem that finds an appropriate location for itself with limited communication capabilities.

In the proposed method, the learning-execution process of the agent group is based on the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [20] algorithm, which is one of the central learning-decentralized execution methods. In the MADDPG algorithm, each agent is trained by the DDPG algorithm, where the actor has access to only local observations. The proposed method, which is a modified version of MADDPG, takes into account the state and actions in a collective intelligence manner using an actor-critic network. This approach helps the MAS cover area by avoiding collisions and orienting the agents to the most suitable locations as soon as possible. In addition, it enables the achievement of maximum PoI coverage in regularly-irregularly shaped areas with connected agents. It is assumed that the agents are homogeneous and have a circular coverage. Agents create an abstract rectangular plane using the $x_{\min}, y_{\min}, x_{\max}$ and y_{\max} values of the target area to be covered and then divide this area into grids. Then, they form path planning for positioning into the nearby grid which has the largest number of PoI. Path planning helps to achieve high fairness index by reducing the number of commonly covered PoIs. While minimizing the energy consumption is handled with path planning, the grid decomposition assists irregularly shaped areas to be covered.

1.4. Organization of the Paper

The remainder of the paper is organized as follows. In Section 2, a MAS model for dynamic area coverage problems is presented by giving detailed information about the proposed method. In Section 3, experimental studies and simulation results are reported. In Section 4, the results are analyzed, and a road-map is drawn for further work.

2. Material-Method

Each agent in the group is modeled as a UAV agent that has the same motion model as a real UAV. It is aimed to meet the following objectives by collaborating with agents:

- Building a distributed system equipped with learning capabilities.
- Covering the maximum number of PoIs.
- Minimizing energy consumption caused by agent movements.
- Maintaining the connection between agents.

- Moving without exceeding the borders of the target area.
- Optimizing agent movements and avoiding collisions between agents.
- Ensuring task continuity in dynamic environments.

With learning capabilities, agents can adapt to changes and achieve collective success. With the design in accordance with the distributed system architecture, it can be ensured that the system can make decisions independently from the central control. Thus, the proposed method designed with the MARL approach helps the agents produce collective success independent of central control.

2.1. Reinforcement Learning

In RL, an agent interacts with the environment and optimizes its behavior using the reward received. As seen in Fig. 1, the agent takes an action to change its state, and then receives a reward from the environment that indicates the quality of the action. One of the key advantages of reinforcement learning is that it enables the agent to learn from experience, without being explicitly programmed with a set of rules or procedures.

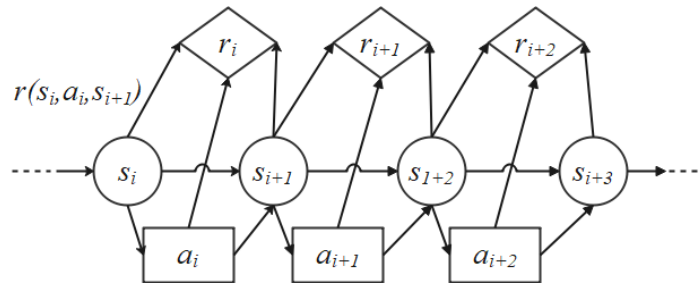


Fig. 1. RL agents

RL is modeled as Markov Decision Process (MDP), which are defined as fully observable environments. The assumption of full observability of MDPs allows the agent to access the current state of the system at each step [8]. Partially-observable Markov Decision Process (POMDP) is a type of MDP that connects unobservable system states to observations probabilistically and is used to model dynamic systems. Depending on the system state and the agent's future actions, the agent may take actions that affect the system in order to maximize expected future rewards. The goal is to find the most appropriate policy that guides the agent's actions. Unlike MDPs, an agent in POMDPs cannot directly observe the entire system state but makes circumstantial observations.

Multi-agent RL involves multiple agents learning to interact with each other and the environment. It focuses on examining the behavior of multiple learning agents coexisting in a shared environment. MARLs exist in which collaborative agents are modeled as nodes that leverage information shared through a communication network. A node means that agents can instantly communicate, exchange information with each other. Therefore, agents know their local and neighbor information. Information sharing helps to obtain more and more stable information about the environment in which the agents are located.

This multi-agent behavior approach is called connected agents. The proposed method in this paper is used connected agents approach to find more appropriate locations using reachable agents' observations.

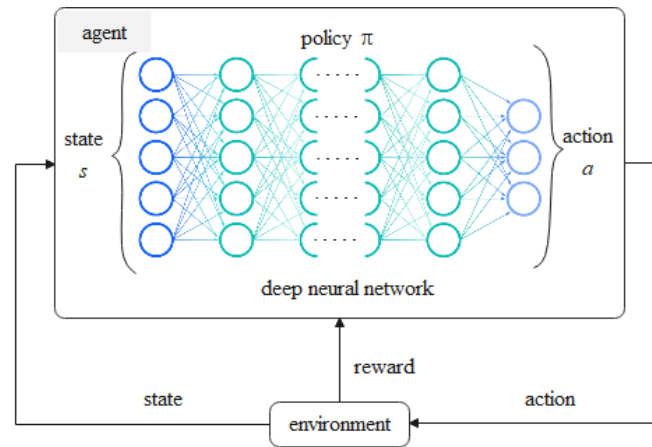


Fig. 2. Deep RL agent

According to [4], deep learning is a machine learning approach that learns multiple representation levels corresponding to different levels of abstraction. In a simple case, it contains two groups of neurons, one receiving the input signal and the other sending the output signal. When the input layer receives an input, it transmits the processed version of the input to the next layer. In a deep network, there are many layers between input and output, consisting of multiple linear and multiple processing layers [28].

As seen in Fig. 2 Deep RL (DRL) improves RL by using deep neural networks as the underlying model to represent the policy, value function or model of the environment. This approach allows the agent to handle more complex, high-dimensional observation spaces and make decisions based on more sophisticated representations of the state of the environment. In other words, DRL combines the power of deep learning with the problem-solving framework of reinforcement learning to create more advanced agents.

Multi-agent deep RL (MADRL) is a multi-agent system that consists of agents which compete or cooperate to solve complex tasks. Each agent has direct access to local observations only. These observations can be many things, a view of the environment, locations relative to landmarks, and even relative locations of other agents. Agents in MADRL are designed distributed manner, and during the learning phase, are guided by a central module or critic. While each agent has only local information and local policies, agents have a critic which reviews their information and advises them on how to update their policies. This approach is called centralized learning. During execution, the removal of the central module limits the state and action space effects. The central module is intended to provide sufficient information for the most appropriate local policy at execution time.

2.2. Multi-agent Deep Deterministic Policy Gradient

DDPG [27] is an actor-critic method that adapts the core achievement of deep Q-learning [6] to continuous action. The MADDPG algorithm has been proposed as an extension of the actor-critic policy gradient methods, in which agents can only access local information and share their policies with other agents. The main idea of the policy gradient (PG) approach is to set a policy parameter to maximize a given target by taking steps in the direction of the given gradient. Using a critic within the system is a common solution to handle the dynamic state of the environment. Therefore, this central critic can be used as a reliable guide to increase the flexibility of agents with local observations. In MADDPG, each agent has two networks: an actor-network and a critic network. The actor-network calculates the action to take based on the situation in which the agent is located, while the critic network evaluates the consequences of the action to improve the performance of the network of actors. Experience replay buffer used for critic network update helps to break correlations in training data and make training more stable. In the training phase, each agent is trained by a DDPG algorithm, where the actor has access to local observations. The centralized critic, on the other hand, combines all states and actions as input and uses the local reward function to obtain the corresponding Q-value. In the execution phase, the critic network is removed, and agents only use the actor-network. This means that the execution is decentralized. In fact, MADDPG can be thought of as a multi-agent version of DDPG. The main goal is to decentralize the execution. Q-learning and DDPG perform poorly in multi-agent environments as they do not use the knowledge of other agents in the group. The MADDPG approach overcomes this challenge by using the observations and actions of all agents.

2.3. Proposed Method

The main purpose of this paper is to construct a method in which a group of UAVs tasked with dynamic area coverage is modeled with a multi-agent deep reinforcement learning (MADRL) approach. In the MADRL approach, UAVs are modeled as mobile agents operating in a dynamic environment and interacting with each other for a common goal. In this way, an intelligent system can be achieved by generating strategies that provide high fairness index with low energy consumption while maximizing coverage.

To simplify agent-based coverage and energy consumption, the target area is divided into grids and the center of each grid is called the grid center (GC). Each agent is intended to be deployed to a GC in a reasonable amount of time.

Assumption 1. It is assumed that all agents in the team have the same properties and move in a 2-D plane. Each agent is represented by A_i where $A_i \in A | i = 1, \dots, N$.

Assumption 2. It is assumed that each agent knows its location. The proposed method uses a geospatial approach to reach the target area, discover other agents in the target area and interact with the agents in the communication range, $(x_i^A(t), y_i^A(t))$. Agents have information on how many agents are on the team. However, it does not have the location information of the agents that are not within communication distance. The positions of all agents in the environment at time t are indicated by:

$$A^N = \{(x_1^A(t), y_1^A(t)), (x_2^A(t), y_2^A(t)), \dots, (x_m^A(t), y_m^A(t))\} \quad (1)$$

Assumption 3. Each agent is assumed to have a circular shape (diameter: \emptyset_A) and a circular detection zone. The distance between A_i and A_j is expressed as $d_{ij} = |A_i A_j|$. For agents A_i and A_j , the following equation must be satisfied:

$$|(x_i^A(t), y_i^A(t)), (x_j^A(t), y_j^A(t))| \leq d_{ij} \quad (2)$$

Each agent has limitations such as communication/detection range. At each learning step, agents divide the target area into grids. When the communication distance is shorter than the grid separation distance, the connection between agents will be broken. However, any agent action to restore communication may cause other agents' positions to change. As a result, each action taken to find a suitable solution increases energy consumption.

In this section, the details of the proposed method are presented in order to achieve maximum coverage with minimum energy consumption under the frame of a high fair index of connected agents (which can exchange information) in the target area.

Grid Decomposition. Within the system, each UAV is represented by an agent. In a 2-D environment, agents learn to locate grid centers in the target area. The target area may not be a regular shape. An abstract area is created for the regular or irregular target area. The abstract area is the smallest regular rectangular area containing the target area. Agents are directed to abstract grid centers that are divided into grids. Agents try to find the fastest and most convenient solution by going to the grid closest to them and containing the maximum number of PoIs. It is also aimed to provide communication between agents in the target area. The smallest regular quadrilateral area containing the target area, with T being the target area and k^T the vertices of the target area are expressed as $\{(x_i^T, y_i^T), \dots, (x_n^T, y_n^T) \forall x^T, y^T \in k^T\}$ for $(x_{\min}^T, y_{\max}^T), (x_{\max}^T, y_{\max}^T), (x_{\max}^T, y_{\min}^T)$, and (x_{\min}^T, y_{\min}^T) , respectively. GCs are denoted by $j \in \{1, \dots, N\}$, while the set of GCs is given by $GC_j \subseteq T$. The set of all PoIs within the target area is defined as:

$$PoI_T = \{PoI_1, \dots, PoI_n\} \quad (3)$$

The grid decomposition algorithm is given in Algorithm 1.

Algorithm 1 Grid decomposition

- 1: Initialize: $L_I(i, k) \leftarrow$ (GC index, number of PoI)– empty list for grids
 - 2: Set: distance for grid decomposition $m, m \leq A_m |A_m$ communication distance of an agent
 - 3: Start: $x_{temp} = x_{\min} + (m/2), y_{temp} = y_{\min} + (m/2), PoI_T \leftarrow$ PoIs in the target area
 - 4: **for** from y_{temp} to y_{\max} **do**
 - 5: $j = 0$
 - 6: **for** from x_{temp} to x_{\max} **do**
 - 7: $GC_j = (x_{temp}, y_{temp})$
 - 8: $x_{temp} = x_{temp} + m$
 - 9: $PoI_{temp} = GC_j.buffer(m/2) \cap PoI_T$
 - 10: $L_I.insert(j, PoI_{temp}.size())$
 - 11: $j ++$
 - 12: $x_{temp} = x_{temp} + (m/2)$
 - 13: $y_{temp} = y_{temp} + m$
- return** L_1
-

First, as seen in Alg. 1, the coordinates of the smallest regular rectangular area containing the target area are found. Then, using the specified decomposition distance (usually the agent's communication/sensing distance), GCs are found where the agent will be located. As seen in Fig. 3, the number of PoIs in GCs is calculated.

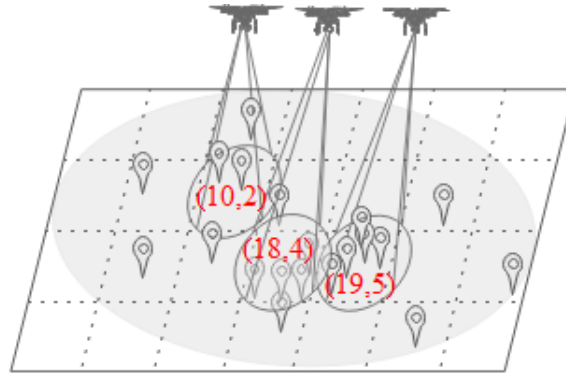


Fig. 3. Grid center and PoI pairs

The goal here is not to create a grid and put agents at its center, but to place agents at the calculated PoInts and to create a coverage area equal to the communication distance of the agents. The circular coverage areas are intersected with PoIs. Finally, it is thrown into a list containing the GC index and the number of PoIs it contains.

Reward Strategy. Minimizing travel distance has many benefits, such as saving time, energy, and equipment usage time. Inspired by [24], a reward structure has been designed to perform target assignments and minimize travel distance. As seen in Alg. 2, the reward is determined by the distance of each agent from each GC. At each step, the distances of the agents to the GCs are calculated, but maximizing the reward received would be an unsuccessful strategy here. Maximizing the reward achieved will distract agents from the target area and prevent areas from being covered. For this reason, the smallest of the distance values obtained is selected. Then, this distance value is multiplied by -1 to get a negative value. In this way, the reward of the agents for distributing the areas separated into the grids converges from negative to 0. In this way, reward-driven learning, which is the basis of RL, is realized.

Algorithm 2 Reward function 1

- 1: Start: $r_1 = 0$
 - 2: **for** from GC^0 to GC^N **do**
 - 3: $d \leftarrow$ calculate the distance between agents and GCs
 - 4: $r_1 = r_1 + (-\min(d))$
- return** r_1
-

Agents moving toward GCs are penalized if they collide with each other. Therefore, agents must learn to cover target areas while avoiding collisions. The distance between the agents is calculated at each step so that a collision can be determined. Considering that the agents are modeled as a circle if the distance is smaller than the sum of the radius of the agents, it means that the collision has occurred.

Algorithm 3 Reward function 2

```

1: Start:  $r_2 = 1, \emptyset_A$ 
2: for from  $agent = 0$  to  $E^N$  do
3:   if  $d_{A_i A_j} < \emptyset_A$  then
4:      $r_2 = \gamma_r * r_2$ 
   return  $r_2$ 

```

In the collision calculation method, whose pseudo-code is given in Alg. 3, γ_r represents the discount factor for collision avoidance. Collisions for A_i are represented by p_i where $w(ij)$ is used to stand for the calculation whether there is a collision between A_i and A_j .

$$p_i = \prod_j w(ij), j \in \{1, \dots, N\} \text{ all } A_j \in E_i^C \quad (4)$$

$$w(ij) = \begin{cases} 1, & (\emptyset_i + \emptyset_j)/2 \leq d_{ij} \\ \gamma_r, & (\emptyset_i + \emptyset_j)/2 > d_{ij} \end{cases} \quad (5)$$

It is calculated to find the percentage of PoIs that needs to be covered.

$$r_c = \sum_i^n c_i \forall i, n \{A_i, \dots, A_n\} \in A^R \quad (6)$$

$$r_3 = \left(1 - \frac{r_c}{PoI^N}\right) * 100 \forall i, n \{PoI_i, \dots, PoI_n\} \in PoI_T \quad (7)$$

The reward function for orienting agents towards GCs with many PoIs is given in Alg. 4:

Algorithm 4 Reward function 3

```

1: Start:  $r_3 = 0$ 
2: for from  $agent = 0$  to  $E^N$  do
3:   Get: reachable agents of  $A_i$ 
4:   Get: PoIs covered by the reachable agents ( $A^R$ )
5:   Sum:  $r_c \leftarrow$  number of the PoIs covered ( $c$ ),  $L_I(i, k)$  using Eq. 6:
6:   Calculate:  $r_3 \leftarrow$  the percentage of PoIs not covered by all PoIs using Eq. 7
   return  $r_3$ 

```

The proposed method has 3 reward functions to maximize covered area with minimum energy consumption. However, these reward functions designed for different purposes affect the coverage quality at different weights. Therefore, coefficient matrix represented

as $[k_1, k_2, k_3]$ is used to calculate the collaborative reward of the system. K is a matrix that contains the coefficients of all the reward functions in the system. The collaborative reward function obtained using reward functions designed for the shortest route, collision avoidance, and covered PoIs is as follows:

$$K = [k_1, k_2, k_3] \quad (8)$$

$$r_i = k_1 r_1 * k_2 r_2 * k_3 r_3 \quad (9)$$

In the collaborative reward function, the reward points for collision avoidance route planning are multiplied by the percentage of PoIs not covered. The path planning process is a negative value; the higher the number of PoIs not covered, the lower the cumulative score will be. This approach speeds up the learning process by growing the reward range. Thus, the RL agents in the proposed method try to converge to 0 with the reward-driven approach using Eq. 8.

Multi-agent System Model. A MAS model is designed to solve the complexity of collaboration between agents. The position of agents A_i moving on a 2-D plane at time t is expressed as $(x_i^A(t), y_i^A(t))$. The target area T , and the GCs are represented by $j \in \{1, \dots, N\}$ for all $GC_j \subseteq T$. In the MAS model, it is aimed to meet the following objectives:

- Agents should be distributed across GCs to cover the maximum number of PoIs.
- Each GC should only be covered by one agent.
i.e., $\forall i, j | GC_i' \neq GC_j' | i, j \in \{1, \dots, GC^T\}$
- Agents positioned on GCs within the target area should be at a distance to communicate with each other.
- Agents should avoid the collision, so two agents cannot be in the same position at that same time t .
i.e., $\forall i, j, (x_i^A(t), y_i^A(t)) \neq (x_j^A(t), y_j^A(t))$

The agent policy is represented by θ and the policy parameter by μ . In the N-agent MAS model, the policies for state transitions are expressed as $\mu = \{\mu_1, \mu_2, \dots, \mu_N\}$, and the parameters as $\theta = \{\theta_1, \theta_2, \dots, \theta_N\}$. Additionally, the model uses an experience replay buffer for model-free training. The agent stores information in the form of $(x, x', a_1, \dots, a_N, r_1, \dots, r_N)$, showing the joint status at each step, the next joint status, the joint action, and the rewards received by each of the agents. Then, it again does a sampling of its buffer to train the agent. The agent's critic is updated using the sampled information. Thus, the loss function using the sampled temporal difference error for the critic is defined as:

$$y = r_i + \gamma Q_i^{\mu'}(x', a'_1, \dots, a'_N) |_{a'_j = \mu'_j(o_j)} \quad (10)$$

$$L(\theta_i) = \frac{1}{S} \sum_j (y^j - Q_i^{\mu}(x^j, a_1^j, \dots, a_N^j))^2 \quad (11)$$

In the above equation, a' , γ , S , o and Q_i^μ represent the next joint action, the discount factor, the size of the randomly selected sample from the replay buffer, the partial observations of the environment, and the central action-value function, respectively. The sampled policy gradient for updating the actor is defined as follows:

$$\nabla_{\theta_i} J \approx \frac{1}{S} \sum_j \nabla_{\theta_i} \mu_i(o_j^i) \nabla_{a_i} Q_i^\mu(x^j, a_j^i, \dots, a_N^j) |_{a_i = \mu_i(o_j^i)} \quad (12)$$

Construction of the proposed method In this paper, a method is proposed that aims to cover the maximum number of PoIs in regular-irregular shaped areas with MAS. In the method designed in the context of the actor-critic, it is aimed to achieve high coverage of the connected agents with low energy consumption. According to the experimental results, the proposed method was able to successfully complete the coverage task in the dynamic environment. In the testing phase, the agents created distributed but collective actions for the common purpose of the group using their own critic and actor networks. In addition, the results show that policies with high fairness index were produced under communication restrictions by adapting to the changing number of agents in the dynamic environment. On the other hand, in the proposed method, the state space depends on local observations. Therefore, the growing action-state space is eliminated. This approach has helped produce an effective policy in less time while the designed reward structure enabled the generation of collective behavior without the need for reward sharing, as seen in Fig. 4. Moreover, the proposed model-free method, which is based on local observations independent of central control, is capable of guiding real applications with its reward strategy.

The behavior strategy of the proposed method is based on the commonly used combination of "attractiveness" and "avoidance". Inspired by [34], "attraction" is considered a positive reward when agents are attracted to GCs in the target area as expressed in Algorithm 2. If the agents go outside the communication distance, it is considered a negative reward. In addition, as presented in Algorithm 4, agents receive positive rewards for PoIs they cover with reachable agents. This positive reward prevents agents from exceeding the communication distance. Therefore, a positive reward is obtained in terms of "attractiveness". Inspired by [11], the collision distance between agents is defined, and based on this distance, a negative reward (avoidance) is given whenever any two agents are too close to each other. Thus, according to Algorithm 3, each agent receives a positive reward for being farther from the others than the specified distance and a negative reward for being too close to others.

The proposed method has a model-free learning structure and operates the learning process using the replay buffer. The centralized module guides agents on how to update their policies during the training period. The training process is divided into sections, and the location of the agents is randomly determined before each section is run. Agents store a tuple of action, status, and reward information in the replay buffer to rerun the scenario during the training process. At each time step, samples are taken from the replay buffer. Critics are updated with Eq. 11, while the policy gradient of agents is updated with Eq. 12. Each agent determines the most appropriate joint action to increase area coverage. In the execution process, the central module is removed, and the actor-network is used for local observations. The pseudo-code of the proposed method is given in Alg. 5.

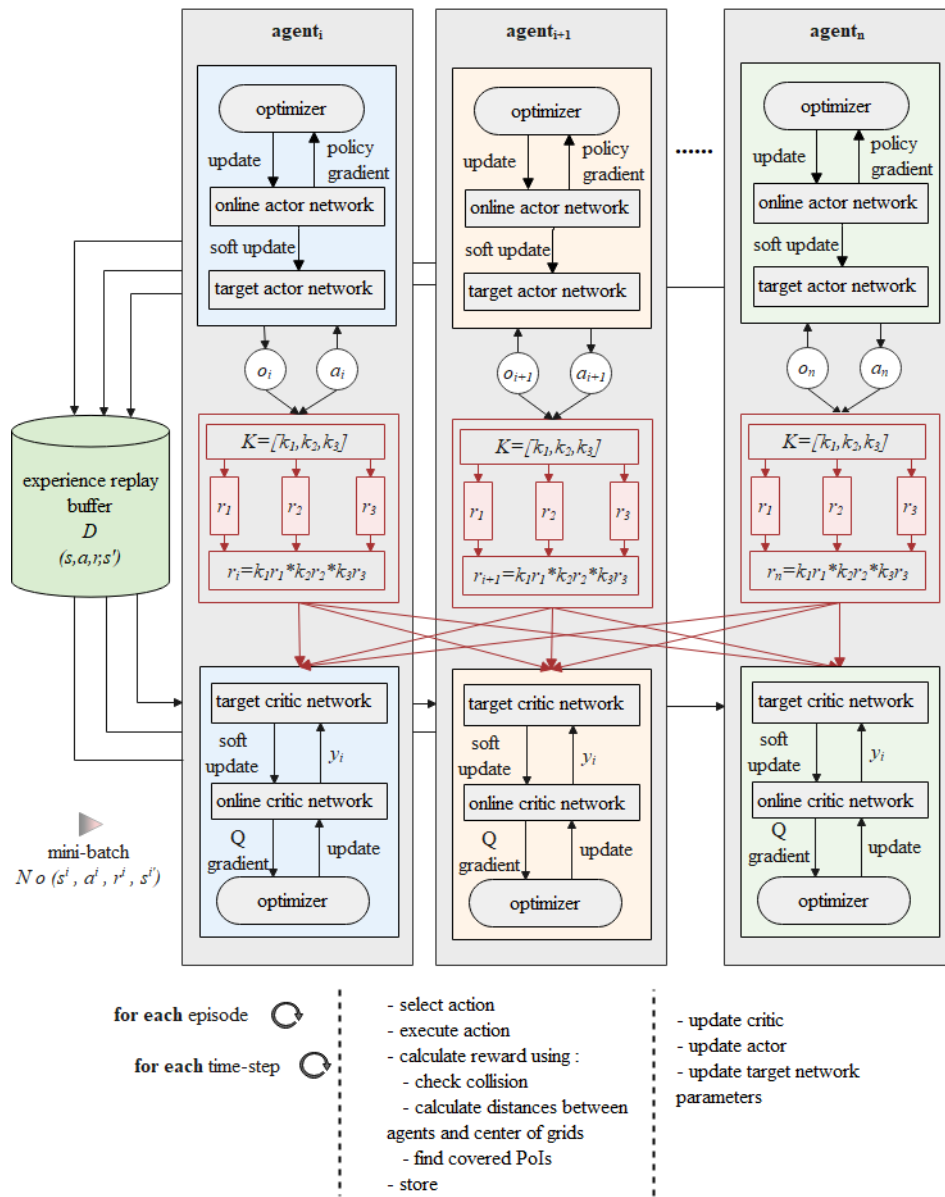


Fig. 4. The framework of the proposed multi-agent deep RL method

Algorithm 5 Proposed method

```

1: Initialize: learning and reward discount factor
2: for from  $episode = 0$  to  $M$  do
3:   Start:  $N$  and target area  $T$ 
4:   Find:  $GC^T$  using Algorithm 1
5:   Get: the initial state of  $x$ 
6:   for from  $stept = 1$  to  $max_s\,tep$  do
7:     Set: for each agent  $i$ , selected action with noise:  $N_t, a_i = \mu_{\theta_i}(o_i) + N_t$ , policy and
       observation:  $w.r.t$ 
8:     Execute: actions  $a = (a_1, \dots, a_N)$ 
9:     Calculate:  $r$  using Algorithm 4
10:    Store:  $(x.a.r.x')$  to replay buffer  $D$ 
11:     $x \leftarrow x'$ 
12:    for from  $agent_i = 0$  to  $N$  do
13:      Sample: mini-batch of  $S$  samples from replay buffer  $D$ 
14:      Update: critic network using Eq. 11
15:      Update: actor-network using the sampled policy gradient using Eq. 12
16:    Update: target network parameters for each agent

```

Training process Each agent has its own actor and critic network. As described in Sect. 2.3, the method learns from experiences (i.e. action, state, and reward) stored in the replay buffer. In other words, actors and critics of all agents are updated by using the mini-batch, which is randomly sampled at each time step during the learning process. In Algorithm 5, the pseudo-code of the learning approach in the training process is given. In the proposed method, the process from the starting position of the agents to positioning in the target area is expressed as an episode. Each episode for training consists of time steps represented t . In the training loop, the system gets the initial state s_1 , and then the initial conditions of the environment are created. Each agent i selects an action according to the actor μ_{θ_i} by observation Q_i . Noise is added to the selected action to prevent the agent from choosing the most appropriate local policy and performing further exploration. Agents performing the selected action obtain a reward value r_t and a new state s_{t+1} . If the selected action forces the agent to leave the target area or collide with other agents, it will be penalized by Eq. 8. Therefore, the agent learns to avoid this action and not to choose the state transition while the last values of (s_t, a_t, r_t, s_{t+1}) are stored in the replay buffer. At the end of the training period, each agent at time step t randomly selects the mini-batch S from the replay buffer D and then updates the critic using Eq. 11. After updating the critic, the actor is updated with Eq. 12. Finally, the target network is gradually updated with the loss function and learning rate.

3. Experimental Studies

A simulation environment is designed using the multi-agent actor-critic platform developed by the OpenAI team [20] to evaluate the proposed method.

3.1. Experimental Settings

A coordinate plane, which is formed of both a horizontal line and a vertical line as 1 unit, is used as a simulation environment where the geometric center is the origin. The agents and target area are located on the plane. A circular shape is used to express an agent. Also, the area coverage of an agent is represented by a circle around the agent. At the beginning of the training episode, the locations of the agents, the shape and the location of the target area, the number and location(s) of PoI(s), and the communication distance of agents are randomly generated. The number of episodes of training is determined as 5000 while the number of step size of the episode is set to 250. The rest of the parameters are specified as follows; the number of units in the multi-layer perceptron is 64, the learning rate is 0.001, the batch size is 1024. The learning rate γ_r to be used in the r_2 is chosen as 1. Due to the random generation of the initial positions of the agents, the simulation scenarios are repeated 50 times. The average of the metrics obtained are taken to evaluate the proposed method's performance.

Two analyses were carried out to determine discount factor and find out the effects of the reward functions to coverage quality. Firstly, it was fixed the number of agents to 5, and then discount factor was set the number of between 0.8 and 0.99 to try to find appropriate one. According to the results, in particular, as γ was increased, benchmark metrics went up until the peak value, then fluctuated it as seen in Fig. 5. According to these result, the best results were obtained when the discount factor was set to 0.88. Secondly, it was tried to find appropriate coefficient values to maximize collaborative reward. In this scenario, three experimental studies applied while only one coefficient function was changed at each experimental study. The coefficient matrices used for collaborative reward function are expressed as $[k_1, 1, 1]$, $[1, k_2, 1]$, $[1, 1, k_3]$ where $k_1, k_2, k_3 \in k = \{0.1, 0.5, 1, 1.5, 2\}$. As seen in Fig. 6, when the weights of the reward functions used for target assignment and area coverage increased, the number of covered PoIs increased while the average step size decreased. However, as the collision avoidance function weight was increased too much, agents focused on collision avoidance and the quality of coverage decreased. According to these results, when the coefficient matrix was set to $[k_1 = 2, k_2 = 0.5, k_3 = 2]$, coverage quality could increase satisfactorily.

The proposed model is compared with DRL-EC3, improved-DRL-EC3, and SBG-AC, using the same simulation settings. Three topics are used for the comparison and validation of the simulation results:

- The effect of increasing the number of agents on coverage: It is the average PoI score covered by the system. It is calculated using algorithm 4.
- The effect of increasing the number of agents on energy consumption: energy efficiency is expressed by the ratio of the number of agents to the PoIs covered. It is the normalized version of the PoIs covered, that is, the ratio of the result obtained by Algorithm 4 to the number of agents in the system.
- Fairness index for covered PoIs: The Jain [13] fairness index for PoI coverage scores. Where N represents the number of PoIs in the environment, $c_t(i)$ represents the number of agents containing the PoI at time t . The case of $J = 1$ indicates perfect fairness between agents.

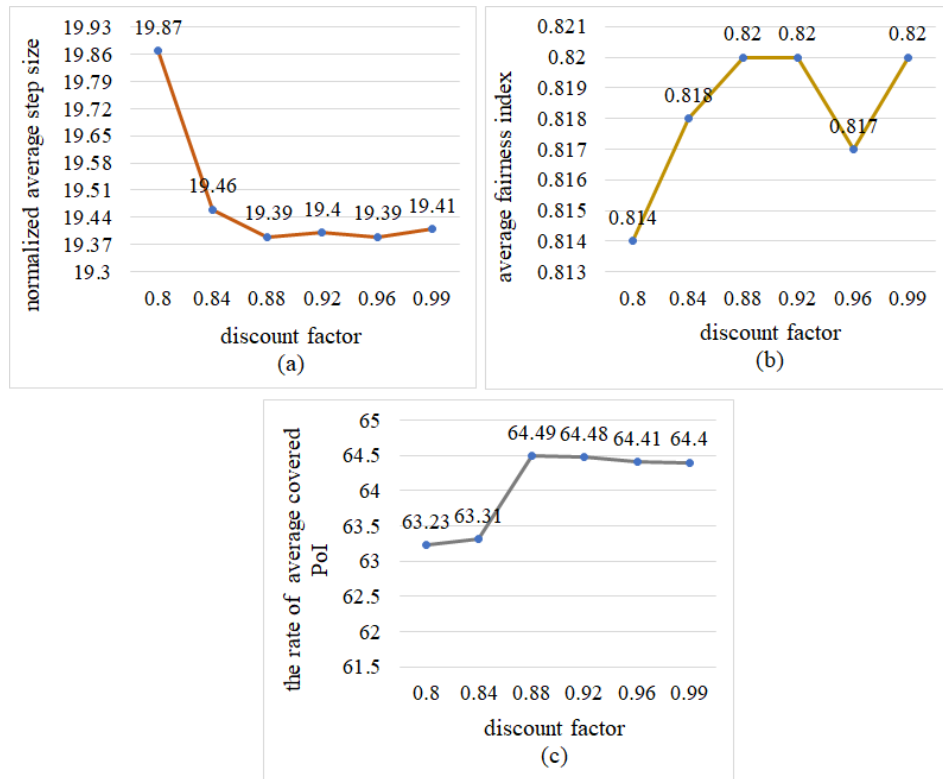


Fig. 5. The effects of the discount factor on coverage quality

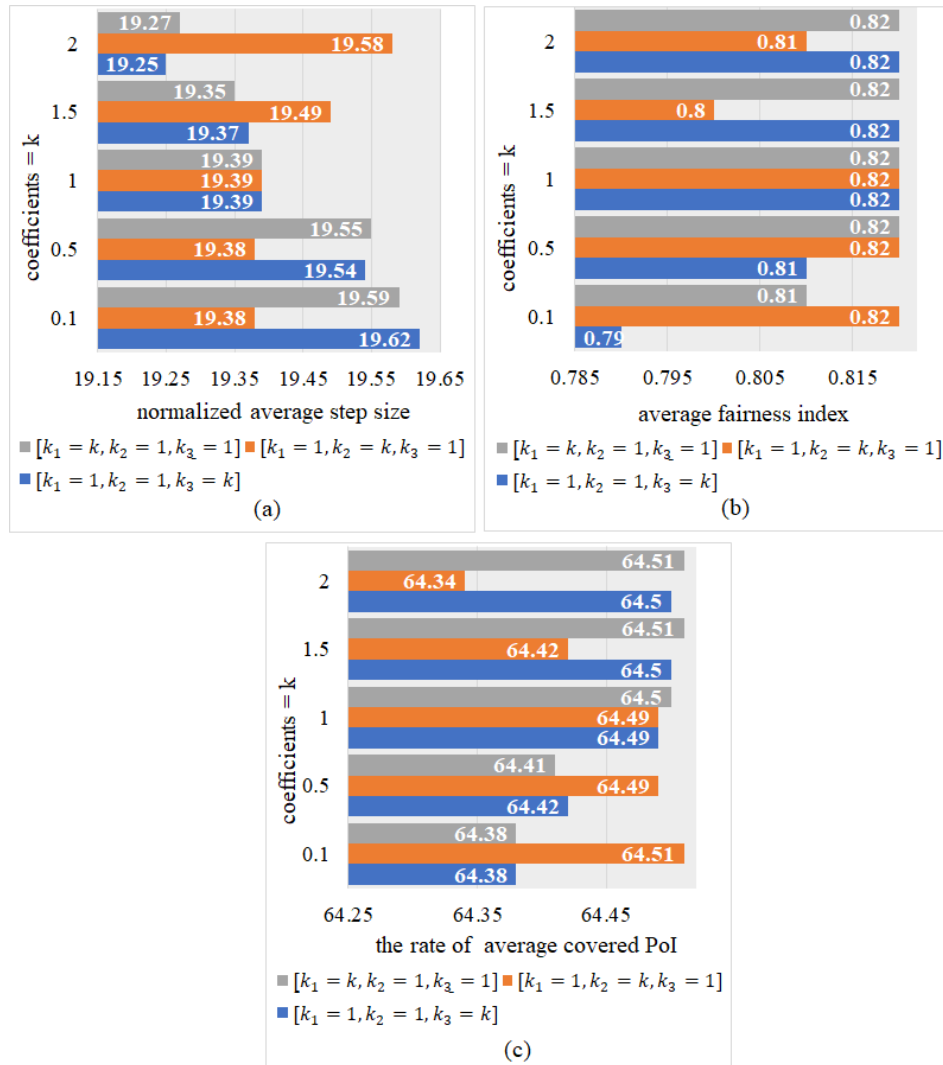


Fig. 6. The effects of the collaborative reward function coefficients on coverage quality

$$J = \frac{(\sum_{i=1}^N c_t(i))^2}{N \sum_{i=1}^N c_t(i)^2} \quad (13)$$

3.2. Experimental Results

In the first experiment, the effect of increasing the number of agents is examined and the performance of the 4 models was compared. The coverage ratios to be used for comparison were calculated according to the method given in Algorithm 4. As seen in Fig. 7, the proposed method achieved approximately 10.1% more coverage than DRL-EC3, 5.06% more coverage than improved-DRL-EC3, and 3.18% more coverage than SBG-AC. For example, when the number of agents was 4, the proposed method covered approximately 59.8%, while DRL-EC3 covered approximately 53.9%, improved-DRL-EC3 approximately 55.2%, and SBG-AC approximately 57.4%. In the case of 8 agents, the coverage obtained by the proposed method was approximately 90.9%, 80.2% obtained by DRL-EC3, 87.3% obtained by improved-DRL-EC3, while the coverage obtained by SBG-AC was 88.7%. There was a similar trend for other scenarios. Therefore, the average covered PoI rate achieved by the proposed method increased regularly. The increase in the number of agents has allowed agents to connect with different patterns in the PoI coverage process.

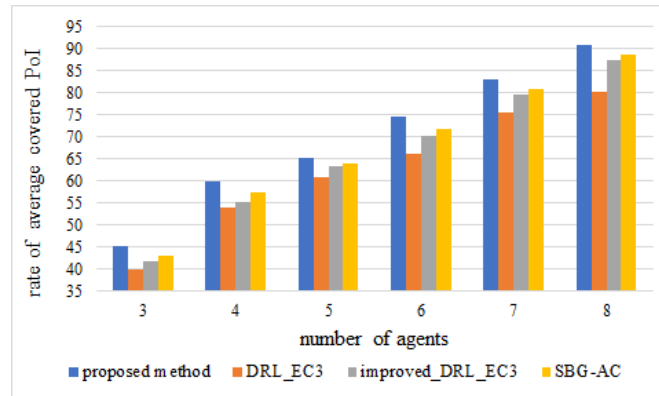


Fig. 7. Agent relationship with the covered PoIs

In the second experimental study, an energy consumption comparison of 4 models was made. In the calculation of energy consumption, the number of actions performed by the agents was used. In each episode, the coverage rate achieved by the system is recorded. At the end of the training, the coverage scores are summed and divided by the number of actions. The result obtained is divided by the number of agents in the system so that the number of PoIs covered per action is found. Finally, it is calculated how many steps an agent takes from the initial position to the final position to cover one PoI. This result represents the normalized average energy consumption. As seen in Fig. 8, DRL-EC3, improved-DRL-EC3, and SBG-AC models consumed approximately 9.35%, 4.44%,

and 2.8% more energy, respectively, than the proposed method. For example, when the number of agents was 4 and 8, the normalized average step size is given as follows; the proposed model was 16.85 and 22.1, DRL-EC3 was 18.55 and 24.94, improved-DRL-EC3 was 18.12 and 22.94, and SBG-AC was 17.77 and 22.53. According to these results, it was observed that there was no significant change in energy consumption values as the number of agents increased. Due to the competitive and cooperative nature of multi-agent systems, the number of agents does not have a large impact on policies. The proposed method has reached the maximum coverage rate by consuming less energy. It is thought that there are 2 reasons for this; (i) a specific reward strategy is designed for path planning; (ii) the reward is calculated by the status of each agent it can access, not just the agent's own status.

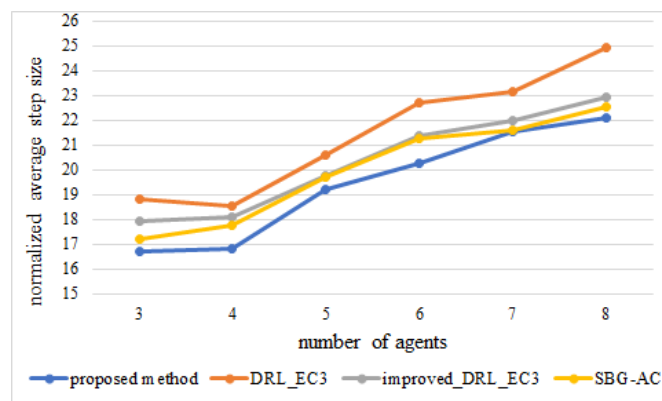


Fig. 8. Number of actions-agent relationships

Finally, four models were compared in terms of fairness index according to the number of agents. In Fig. 9, the fairness index results obtained by different numbers of agents for the 4 models are presented. The proposed model appeared to have a better fairness index than DRL-EC3, improved-DRL-EC3, and SBG-AC with an average increase of about 11.14%, 2.34%, and 1.76%, respectively. For example, when the number of agents is 3, the proposed model achieved 0.71 fairness index, DRL-EC3 obtained 0.588 fairness index, improved-DRL-EC3 obtained 0.698 fairness index and SBG-AC obtained 0.698 fairness index. When the number of agents was 6, the fairness indices of the proposed model, DRL-EC3, improved-DRL-EC3, and SBG-AC were 0.937, 0.801, 0.911, and 0.919, respectively. A similar trend is observed for other scenarios as well. The larger the number of agents, the greater the number of grids covered. This relationship leads to a direct improvement in the fairness index. In addition, with fewer steps in the proposed method, a similar fairness index rate was achieved with other methods due to purpose-based reward strategies.

In this section, the behavior of the proposed model against issues such as energy, coverage, and fairness index are examined. Then, using these three metrics, the proposed method is compared with DRL-EC3, improved-DRL-EC3, and SBG-AC models. The simulation results are summarized in Table 1.

Table 1. Simulation results

	proposed method	DRL-EC3	improved-DRL-EC3	SBG-AC
Approach	MARL	MARL	MARL	MARL
Algorithm	MADDPG	DDPG	MADDPG	MADDPG
Reward strategy	Agent&group-specific	Agent specific	Agent specific	Group specific
Evaluation Metric Average rate of PoIs covered	3 agents: 45.2	3 agents: 39.8	3 agents: 41.8	3 agents: 42.9
	4 agents: 59.8	4 agents: 53.9	4 agents: 55.2	4 agents: 57.4
	5 agents: 65.1	5 agents: 60.7	5 agents: 63.3	5 agents: 63.8
	6 agents: 74.6	6 agents: 66.2	6 agents: 70.2	6 agents: 71.7
	7 agents: 83.1	7 agents: 75.6	7 agents: 79.7	7 agents: 80.9
	8 agents: 90.9	8 agents: 80.2	8 agents: 87.3	8 agents: 88.7
Evaluation Metric Normalized average step size	3 agents: 16.72	3 agents: 18.83	3 agents: 17.95	3 agents: 17.2
	4 agents: 16.85	4 agents: 18.55	4 agents: 18.12	4 agents: 17.77
	5 agents: 19.21	5 agents: 20.59	5 agents: 19.76	5 agents: 19.69
	6 agents: 20.28	6 agents: 22.68	6 agents: 21.37	6 agents: 21.26
	7 agents: 21.54	7 agents: 23.15	7 agents: 21.98	7 agents: 21.61
	8 agents: 22.1	8 agents: 24.94	8 agents: 22.94	8 agents: 22.53
Evaluation Metric Average fairness index	3 agents: 0.71	3 agents: 0.588	3 agents: 0.698	3 agents: 0.698
	4 agents: 0.75	4 agents: 0.694	4 agents: 0.724	4 agents: 0.731
	5 agents: 0.82	5 agents: 0.762	5 agents: 0.812	5 agents: 0.814
	6 agents: 0.937	6 agents: 0.801	6 agents: 0.911	6 agents: 0.919
	7 agents: 0.946	7 agents: 0.844	7 agents: 0.915	7 agents: 0.923
	8 agents: 0.957	8 agents: 0.86	8 agents: 0.94	8 agents: 0.945

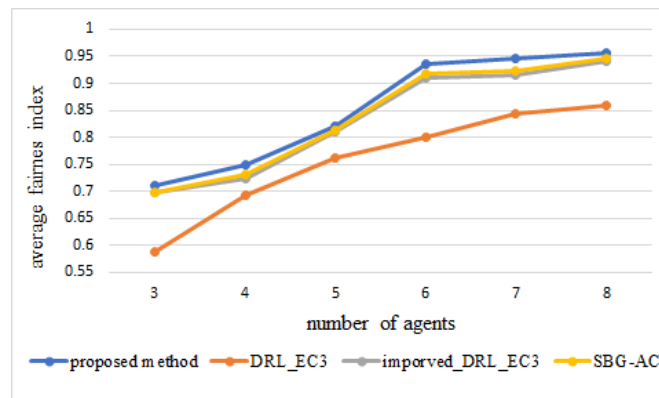


Fig. 9. Average fairness index with respect to number of agents

In this paper, 3 different reward strategies are designed as purpose-based while there is only one reward strategy in the DRL-EC3, improved-DRL-EC3 and SBG-AC methods. With the collaborative design, low energy consumption-high coverage ratio has been tried to be achieved. The use of 3 different reward strategies increases the space of the reward that can be obtained and clarifies the relationship between situation-action pairs. Therefore, while learning time decreases, learning quality increases. There are two main reasons for this result; (i) connected agents are positioned on grids without going out of communication range, (ii) agents receive high rewards as they distribute into grids and reduce intersections. DRL-EC3 and improved-DRL-EC3 use only agent-specific reward strategies that the reward received is only affected by agent-environment interaction. However, SBG-AC and the proposed method are interested in the current state of the environment, in which all agents share a common reward. Contrary to the proposed method, there is no reward strategy related to target assignment in DRL-EC3, improved-DRL-EC3, and SBG-AC methods. In the proposed method, agents try to maximize their reward points by using the information of connected agents. This approach forces agents to connect with each other and make collaborative decisions. In addition, successful results were obtained in terms of energy consumption, since the agents tried to locate the GC with the maximum number of PoIs in the closest distance to them. This approach represents the path planning algorithm with target assignment in MAS. With the help of this algorithm, the intersection of the agents in the coverage areas is minimized and a high fairness index result is obtained.

4. Conclusions

In this paper, a method is proposed that aims to cover the maximum number of PoIs in regular-irregular shaped areas with MAS. In the method designed in the context of the actor-critic, it is aimed to achieve high coverage of the connected agents with low energy consumption. According to the experimental results, the proposed method was able to successfully complete the coverage task in the dynamic environment. In the testing phase, the agents created distributed but collective actions for the common purpose of the

group using their own critic and actor networks. In addition, the results show that policies with high fairness index were produced under communication restrictions by adapting to the changing number of agents in the dynamic environment. On the other hand, in the proposed method, the state space depends on local observations. Therefore, the growing action-state space is eliminated. This approach has helped produce an effective policy in less time while the designed reward structure enabled the generation of collective behavior without the need for reward sharing. Moreover, the proposed model-free method, which is based on local observations independent of central control, is capable of guiding real applications with its reward strategy.

References

1. Aydemir, F., Cetin, A.: Multi-agent dynamic area coverage based on reinforcement learning with connected agents. *Computer Systems Science and Engineering* 45(1), 215–230 (2023)
2. Cabreira, T.M., Ferreira, P.R., Franco, C.D., Buttazzo, G.C.: Grid-based coverage path planning with minimum energy over irregular-shaped areas with uavs. In: 2019 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 758–767 (2019)
3. Canese, L., Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., Spanò, S.: Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences* 11(11) (2021)
4. Deng, L., Yu, D.: Deep learning: Methods and applications. *Found. Trends Signal Process.* 7(3–4), 197–387 (2014)
5. Dorri, A., Kanhere, S.S., Jurdak, R.: Multi-agent systems: A survey. *IEEE Access* 6, 28573–28593 (2018)
6. Fan, J., Wang, Z., Xie, Y., Yang, Z.: A theoretical analysis of deep q-learning. *arXiv* (2019), [Online]. Available: <https://arxiv.org/abs/1901.00137> (current December 2022)
7. Ganganath, N., Cheng, C.T., Tse, C.K.: Distributed antiflocking algorithms for dynamic coverage of mobile sensor networks. *IEEE Transactions on Industrial Informatics* 12(5), 1795–1805 (2016)
8. Ge, Y., Zhu, F., Huang, W., Zhao, P., Liu, Q.: Multi-agent cooperation q-learning algorithm based on constrained markov game. *Computer Science and Information Systems* 17(2), 647–664 (2020)
9. Gupta, H., Verma, O.P.: Monitoring and surveillance of urban road traffic using low altitude drone images: A deep learning approach. *Multimedia Tools Appl.* 81(14), 19683–19703 (2022)
10. Gupta, S.K., Kuila, P., Jana, P.K.: Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks. *Computers & Electrical Engineering* 56, 544–556 (2016)
11. Hüttenrauch, M., Susic, A., Neumann, G.: Local communication protocols for learning complex swarm behaviors with deep reinforcement learning. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A., Reina, A., Trianni, V. (eds.) *Swarm Intelligence, ANTS 2018. Lecture Notes in Computer Science*(), vol. 11172, pp. 71–83. Springer, Cham, Rome, Italy
12. Jagtap, A.M., Gomathi, N.: Minimizing movement for network connectivity in mobile sensor networks: An adaptive approach. *Cluster Computing* 22(1), 1373–1383 (2019)
13. Jain, R., Chiu, D.M., Wu, H.: A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *CoRR* (1998)
14. Kalantari, E., Yanikomeroglu, H., Yongacoglu, A.: On the number and 3d placement of drone base stations in wireless cellular networks. In: 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall). pp. 1–6 (2016)

15. Kuo, Y.C., Chiu, J.H., Sheu, J.P., Hong, Y.W.P.: Uav deployment and iot device association for energy-efficient data-gathering in fixed-wing multi-uav networks. *IEEE Transactions on Green Communications and Networking* 5(4), 1934–1946 (2021)
16. Lee, H.R., Lee, T.: Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response. *European Journal of Operational Research* 291(1), 296–308 (2021)
17. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. *arXiv* (2015), [Online]. Available: <https://arxiv.org/abs/1509.02971> (current December 2022)
18. Liu, C.H., Chen, Z., Tang, J., Xu, J., Piao, C.: Energy-efficient uav control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE Journal on Selected Areas in Communications* 36(9), 2059–2070 (2018)
19. Liu, C.H., Ma, X., Gao, X., Tang, J.: Distributed energy-efficient multi-uav navigation for long-term communication coverage by deep reinforcement learning. *IEEE Transactions on Mobile Computing* 19(6), 1274–1285 (2020)
20. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. pp. 6382—6393. Curran Associates Inc., Red Hook, NY, USA (2017)
21. Mozaffari, M., Saad, W., Bennis, M., Debbah, M.: Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters* 20(8), 1647–1650 (2016)
22. Ndam Njoya, A., Ari, A., Awa, M., Titouna, C., Labraoui, N., Effa, Y., Abdou, W., Gueroui, A.: Hybrid wireless sensors deployment scheme with connectivity and coverage maintaining in wireless sensor networks. *Wireless Personal Communications* 112, 1893—1917 (2020)
23. Nemer, I.A., Sheltami, T.R., Belhaiza, S., Mahmoud, A.: Energy-efficient uav movement control for fair communication coverage: A deep reinforcement learning approach. *Sensors* 22(5), 1–27 (2022)
24. Qie, H., Shi, D., Shen, T., Xu, X., Li, Y., Wang, L.: Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access* 7, 146264–146272 (2019)
25. Shi, W., Li, J., Xu, W., Zhou, H., Zhang, N., Zhang, S., Shen, X.: Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks. *IEEE Access* 6, 12518–12529 (2018)
26. Shu, T., Dsouza, K.B., Bhargava, V., Silva, C.: Using geometric centroid of voronoi diagram for coverage and lifetime optimization in mobile wireless sensor networks. In: *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. pp. 1–5 (2019)
27. Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: *31st International Conference on Machine Learning, ICML 2014, Proceedings of Machine Learning Research*. vol. 32, pp. 387–395 (2014)
28. Song, H., Lee, S.Y.: Hierarchical representation using nmf. In: Lee, M., Hirose, A. and Hou, Z., Kil, R. (eds.) *Neural Information Processing, ICONIP 2013. Lecture Notes in Computer Science*, vol. 8226, pp. 466–473. Springer
29. Valente, J., Sanz, D., Cerro, J., Barrientos, A., de Frutos, M.: Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. *Precision Agriculture* 14, 115—132 (2013)
30. Woolley, A.W., Aggarwal, I., Malone, W.T.: Collective intelligence and group performance. *Current Directions in Psychological Science* 24(6), 420–424 (2015)
31. Xiao, J., Wang, G., Zhang, Y., Cheng, L.: A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning. *IEEE Access* 8, 33511–33521 (2020)

32. Ye, Z., Wang, K., Chen, Y., Jiang, X., Song, G.: Multi-uav navigation for partially observable communication coverage by graph reinforcement learning. *IEEE Transactions on Mobile Computing* pp. 1–1 (2022)
33. Yue, Y., Cao, L., Luo, Z.: Hybrid artificial bee colony algorithm for improving the coverage and connectivity of wireless sensor networks 108, 1719—1732 (2019)
34. Zoss, B.M., Mateo, D., Kuan, Y.K., Tokić, G., Chamanbaz, M., Goh, L., Vallegra, F., Bouffanais, R., Yue, D.K.: Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies. *Auton. Robots* 42(8), 1669—1689 (2018)

Fatih Aydemir received the B.S. degree in computer engineering from 9 Eylul University, Turkey, and the M.S. degree in computer engineering from Gazi University, Turkey. He is currently pursuing the Ph.D. degree in computer engineering with the Gazi University. He is also Leader Software Engineer and working for STM Defence Technologies Engineering and Trade. Inc., Ankara, Turkey.

Aydin Cetin is currently with Gazi University Department of Computer Engineering. He received his MS degree from LSU in Electrical and Computer Engineering and PhD. Degree in Electrical Ed. from Gazi University. His current research areas of interest include AI, Smart and Autonomous things, Agent systems and optimization techniques. He is member of IEEE, IEEE Computational Intelligence and PE Societies, and IEEE Big Data, Cyber Security and IoT communities.

Received: December 22, 2022; Accepted: March 25, 2023.

The Effective Skyline Quantify-utility Patterns Mining Algorithm with Pruning Strategies

Jimmy Ming-Tai Wu¹, Ranran Li¹, Pi-Chung Hsu², and Mu-En Wu^{3,*}

¹ College of Computer Science and Engineering, Shandong University of Science and Technology
Shandong, China

wmt@wmt35.idv.tw
734181156@qq.com

² Department of Information Management, Shu-Te University

Kaohsiung, Taiwan
pichung@stu.edu.tw

³ Department of Information and Finance Management, National Taipei University of Technology
Taipei, Taiwan

mnsial@gmail.com

Abstract. Frequent itemset mining and high-utility itemset mining have been widely applied to the extraction of useful information from databases. However, with the proliferation of the Internet of Things, smart devices are generating vast amounts of data daily, and studies focusing on individual dimensions are increasingly unable to support decision-making. Hence, the concept of a skyline query considering frequency and utility (which returns a set of points that are not dominated by other points) was introduced. However, in most cases, firms are concerned about not only the frequency of purchases but also quantities. The skyline quantity-utility pattern (SQUP) considers both the quantity and utility of items. This paper proposes two algorithms, FSKYQUP-Miner and FSKYQUP, to efficiently mine SQUPs. The algorithms are based on the utility-quantity list structure and include an effective pruning strategy which calculates the minimum utility of SQUPs after one scan of the database and prunes undesired items in advance, which greatly reduces the number of concatenation operations. Furthermore, this paper proposes an array structure superior to utilmax for storing the maximum utility of quantities, which further improves the efficiency of pruning. Extensive comparison experiments on different datasets show that the proposed algorithms find all SQUPs accurately and efficiently.

Keywords: Internet of Things, skyline quantity-utility patterns (SQUPs), utility-quantity list, minimum utility of SQUPs (MUSQ), quantity maximum utility of the array (QMUA).

1. Introduction

The Internet of Things (IoT) has resulted in the daily generation of massive amounts of data, making the extraction of valuable information a significant challenge. Data mining techniques, also known as knowledge discovery from databases (KDD) [2,18,30,47], can be applied in this endeavor. Association rule mining (ARM) [3,4,5] and frequent item-set

* Corresponding author

mining (FIM) [16,17,27,48] are traditional methods for processing data. ARM typically finds not only frequent itemset (FI) patterns based on a user-defined minimum support threshold (*minsup*) but also correlations or causal structures between different item sets based on a minimum confidence threshold (*minconf*). ARM and FIM are widely applied in fields such as news recommendation, weather correlation analysis, precision marketing, and price prediction.

Both FIM and ARM count how many times a commodity appears in a transaction by measuring if a specific commodity or a combination of commodities is present. This means that other essential factors, such as the profit of the commodity or the number of purchases, are not considered. In practice, these factors are often more important to the user. In order to further satisfy the needs of users, a concept called high-utility itemset mining (HUIM) has been proposed, it is gradually becoming the focus of research in the field of big data [6,14,44,45]. In a large shopping mall, for example, the number of luxury bags sold in a single day is much lower than the number of daily necessities. However, the profits generated by luxury bags might be higher than those of daily necessities. Yao *et al.* [45] proposed finding high-utility item sets (HUIs) by considering the number of items and the profit per unit of items. In FIM, if an item set $\{AB\}$ is frequent, then any subset of this item set, such as $\{A\}$ or $\{B\}$ is frequent; however, in HUIM, if an item set $\{AB\}$ is an HUI, its subset $\{A\}$ or $\{B\}$ is not necessarily an HUI. Thus, HUIM does not satisfy the downward closure property. If there are n items, then $2^n - 1$ combinations are generated, which requires a large search space in order to determine whether this set of items is a conforming HUI. To solve this difficulty, Liu *et al.* [26] proposed a new model called TWU, in which the utility also satisfies the downward closure property, which greatly narrows the search space. Subsequently, several scholars have researched and successfully proposed new algorithms and effective pruning strategies [7,34,40,43].

To achieve information extraction, these algorithms require the user to set a threshold, which determines the final quality of the results. If the value is too high, much of the useful information will be ignored. If the value is too small, much of the extracted information will be redundant. Setting a suitable parameter is also time-consuming and inefficient for the user. To address this challenge, the concept of Top- k [12,37] was proposed, i.e., the user can extract the top k most essential pieces of information from the database by setting a parameter k . Although this approach significantly shortens the decision-making process, information is only extracted from a single aspect. FIM can help users to find goods that are frequently purchased, and HUIM can help users to find goods that can earn high profits; however, it is important to firms to know what goods are frequently purchased and generate high profits. Therefore, Goyal *et al.* [15] proposed an algorithm to find the frequent-utility skyline (SFU), which is a set of points measuring frequency and utility that are not dominated by each other. Considering that the quantity of items purchased by users is also a concern in real life, Wu *et al.* [42] subsequently designed the skyline quantity-utility pattern (SQUP) model to include the factor of quantity and proposed two algorithms based on UQL structure: SQU-Miner and SKYQUP. However, because these two algorithms generate numerous candidate sets, they create a vast search space.

With the widespread adoption of the IoT, intelligent decision support systems (IDSSs) have evolved into powerful tools for extracting useful information from large amounts of data. This paper proposes a smart supermarket model to demonstrate the application of

the proposed algorithm (see Fig. 1). Touchable smart electronic screens, gravity sensors, and image sensors are all included in the proposed smart shopping cart. The electronic screens summarize the list of products purchased and calculate the total number of items purchased. These electronic screens send information back to the supermarket's data center, and the supermarket can use the proposed algorithm to find non-dominated points and extract valuable patterns. Based on these, the supermarket can design effective marketing strategies.

The main contributions of this paper are as follows:

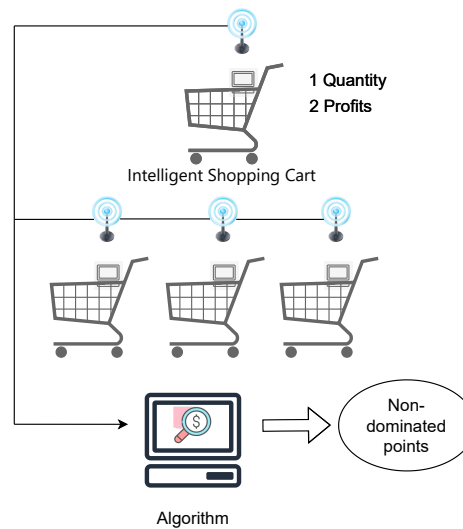


Fig. 1. Skyline model framework in smart supermarket

1. This paper presents two efficient UQL structure-based algorithms for mining SQUPs (FSKYQUP-Miner and FSKYQUP), both of which are depth-first search-based algorithms that do not require user-defined thresholds.
2. The maximum utility of the quantity is stored in a *QMUA* array, and based on this array, an efficient pruning strategy is proposed to prune undesired candidates and their extended sets.
3. The minimum utility of the SQUPs (*MUSQ*) is found in order to eliminate undesired individual items and all their extended sets in the initial stage of the algorithm using the TWU property. This dramatically narrows the search space of the algorithm.
4. Extensive experiments were conducted on real-world and synthetic datasets, the results of which demonstrate the efficacy of the proposed algorithm compared to existing approaches.

The remainder of this thesis is organized as follows. In Sect. 2, we review the current research on HUIM and skyline queries. Sect. 3 presents relevant formulas and definitions

while Sect. 4 details the proposed algorithm, including the proposed pruning strategy and the pseudo-code of the algorithm. Sect. 5 presents the experimental comparison results, and a summary and directions for future research are presented in Sect. 6.

2. Related Work

In this section, we briefly review research on high-utility itemset mining and skyline queries.

2.1. High-utility itemset mining

FIM algorithms in general include level-wise and pattern-growth algorithms. Apriori [5] was the first algorithm proposed for the former, and it satisfies the DC property. However, it does generate massive candidate sets and necessitates several database scans to compute these candidate sets. To address this issue, a new FP-Growth algorithm based on pattern growth [17] is proposed, which is based on the compact data structure FP-Tree. It only scans the database once and does not generate any candidate sets for recursively mining FIs from the database. While other algorithms for investigating FIs have been proposed in recent years, all are single-minded and can only compute the frequency of item sets while ignoring critical metrics such as quantity, weight, and utility.

With its focus on utility, HUIM has been widely studied as an important tool for data mining. HUIM computes the revenue generated by a commodity or combination of commodities and compares it to a minimum revenue parameter specified by the user; if it is greater than this parameter, this item is placed in an HUI. Because HUIM lacks DC like the Apriori algorithm, Liu *et al.* [26] proposed upper bound TWU in order to find more comprehensive HUIs. The TWU-based model, however, necessitates multiple database scans and a vast search space. Subsequently, Lin *et al.* [22] proposed a new structure called a high-utility pattern (HUP) tree based on the FP-Tree to improve the quality of mining performance. However, precisely because the algorithm is based on the FP-Tree, a large portion of memory is needed to store the generated intermediate nodes. Therefore, Tseng *et al.* proposed a new UP-Tree structure to maintain similarity with the FP-Tree structure and proposed two algorithms, UP-Growth [38] and UP-Growth+ [36], to efficiently mine HUIs by reducing the number of database scans. These tree-structure algorithms nevertheless generate a large number of candidate item sets. Liu *et al.* [25] created a new utility list (UL) structure based on the TWU model and proposed the HUI-Miner algorithm. This structure does not require multiple scans of the database and does not generate a large number of candidate sets. The list concatenation operation makes mining HUIs simple, efficient, and complete. Further HUIM extensions have subsequently been proposed [13,24], including the top- k algorithm [12,39], which mines the top k eligible item sets in the database to overcome the necessity of setting a threshold value. Modifications have also been proposed [20,41,49] to reduce the algorithm runtime by improving pruning strategies and designing better data structures.

2.2. The previous hybrid approach

The works reviewed above focus on a single factor, which is inconvenient for decision-making. Yeh *et al.* [46] thus combined utility and frequency in the FUP model; however,

in this approach, a threshold must still be set by the user. Podpecan *et al.* [31] proposed a novel algorithm to increase mining efficiency that also requires user-defined parameters. Goyal *et al.* [15] then proposed SKYMINE, which does not require the user to set any parameters. This algorithm is based on the well-known UP-Tree structure and returns a set of points for decision-making that is not dominated by any other points. However, due to the limitations of its data structure, the algorithm generates numerous candidate sets and is thus inefficient. Pan *et al.* [28] proposed an efficient utility list structure-based SFU-Miner algorithm to reduce the number of candidate sets. Lin *et al.* [23] proposed two algorithms based on the UL structure, called SKYFUP-D and SKYFUP-B algorithms, which are two typical algorithms based on DFS and BFS search. Although the application of list structure has dramatically improved mining efficiency, researchers continue to search for more effective pruning strategies. Song *et al.* [33] proposed SFUI-UF, which deletes undesired item sets from the database in the initial stages of the algorithm and thus considerably shortens runtime. Song *et al.* [32] also proposed cross-entropy-based mining algorithm SFU-CE to improve mining efficiency. These algorithms all consider the utility and frequency of items but neglect the fact that in practice, the quantity of items purchased is still the primary concern of users. Wu *et al.* [42] were the first to suggest considering utility and quantity, proposing the SQUP model and two new algorithms to mine SQUPs.

2.3. The skyline concept

Mining SFUPs from a database is, in general, a multi-objective optimization case that considers frequency and utility and returns a set of points as a solution. That is, subsets $\{a_1, a_2, \dots, a_m\}$ (holding information valuable to the user) are found within a large set of databases D . These subsets are not dominated by other points in at least one dimension. If, for example, there exists a point b_n which is better than a_n in all dimensions, then a_n is dominated by b_n and will eventually return to b_n as the decision point instead of a_n . This skyline result is highly relevant to real-world scenarios. For instance, parents may consider house price and distance from schools when choosing a suitable residence. Generally, house prices close to schools will be higher than those far from schools; therefore, parents look for distances and prices that are relatively suitable. In Fig. 2, the x -coordinate represents the distance to the school, with larger values representing longer distances; the y -coordinate represents house prices, with larger values representing higher prices; and the buildings in the figure represent houses available for rent. The houses $\{g, c, l\}$ in the figure are the skyline points because these points are not dominated by other points in the dimensions of distance and price; therefore, these houses represent the best choices.

Kung *et al.* [21] introduced the skyline concept in 2005, using a “partitioning” strategy to find skyline points. Borzsonyi *et al.* [8] were the first to combine skylines and databases, proposing an algorithm based on block nested loops, which gained wide attention. Chomicki *et al.* [10] improved this block nested loop algorithm using a specific tuple order in the window to improve the performance. Tan *et al.* [35] proposed two algorithms, Bitmap and Index, which output skyline points step by step, unlike the usual algorithms that need to traverse the dataset at least once to return the first point. Kossmann *et al.* [19] proposed an NN algorithm based on nearest-neighbor search and used a form of “partitioning” to compute skyline queries. Papadias *et al.* [29] proposed the branch-and-bound skyline (BBS) algorithm, which is also based on nearest-neighbor search and has the

characteristics of I/O so that it can be applied to various asymptotic operations. These explorations of skyline computation have been widely discussed [1,9].

Traditional algorithms FIM and HUIM consider only one factor, while skyline algorithms return non-dominated points based on multiple factors. This paper proposes a list-based FSKYQUP-Miner and FSKYQUP algorithm to mine SQUPs using the utility quantity list structure for the join operation. The preparatory knowledge and problem statement of skyline quantity utility pattern mining (SQUPM) are presented in the following section.

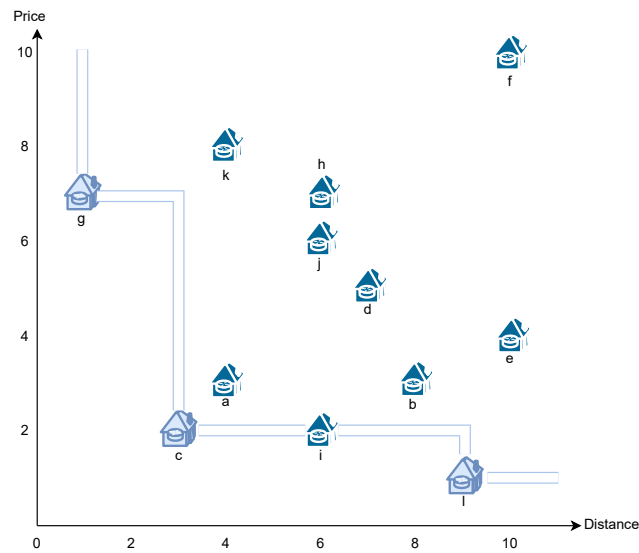


Fig. 2. Example of skyline points

3. Preliminary Knowledge and Problem Statement

3.1. Preliminaries

Assuming that $D = \{T_1, T_2, \dots, T_n\}$ is a transaction database with n transactions, $I = \{i_1, i_2, \dots, i_m\}$ is a set of m distinct items in the database. In D , each transaction $T_q \in D$ is a subset of I containing a number of items and their purchased quantities $q(i_j, T_q)$, along with a unique identifier called TID . Additionally, a profit table called $ptable = \{pr_1, pr_2, \dots, pr_m\}$, where pr_j is the per-unit profit (profit) generated by each item i_j (i.e., good). An itemset $X = \{i_1, i_2, \dots, i_k\}$ is a set of k distinct items, where k is the length of the k -itemset. If $X \subseteq T_q$, then the set of items X is said to occur in transaction T_q . In this paper, our running example is shown in Table 1, which is a database consists of 7 transactions, and in Table 2 the profit corresponding to each item in the running example is given.

Table 1. Original transaction database DB in the running instance

T_{ID}	Item and its quantity	Transaction utility
T_1	B:2,D:2,E:3	23
T_2	A:2,B:3	8
T_3	B:2,C:3,D:4,E:1	33
T_4	B:2,D:2	14
T_5	A:1,D:2,E:2	17
T_6	C:3,E:2	12
T_7	A:2,B:1,C:1,D:1,E:2	17

Table 2. Unit profit table of the item in the running example

Item	Profit
A	1
B	2
C	2
D	5
E	3

Definition 1. In the transaction T_q , the quantity of the itemset X to be purchased is denoted as $q(X, T_q)$, a mathematical definition of which is as follows:

$$q(X, T_q) = \min\{q(Y) | Y \subseteq X \wedge X \in T_q \wedge Y \in T_q\}. \tag{1}$$

It is obtained from T_2 in Table 1 that $q(A) = 2$, $q(B) = 3$, so the quantity of the itemset (AB) is the smallest one, which is 2.

Definition 2. The utility of an item i_j in a transaction T_q is called as $u(i_j, T_q)$, a mathematical definition of which is as follows:

$$u(i_j, T_q) = q(i_j, T_q) \times pr(i_j). \tag{2}$$

It is obtained from T_2 in Table 1 that the utility of the item $\{A\}$ can be computed as $u(A, T_2) = q(A, T_2) \times pr(A) = 2 \times 1 = 2$, the utility of the item $\{B\}$ can be computed as $u(B, T_2) = q(B, T_2) \times pr(B) = 3 \times 2 = 6$.

Definition 3. The utility of an itemset X in a transaction T_q is called as $u(X, T_q)$, a mathematical definition of which is as follows:

$$u(X, T_q) = \sum_{i_j \subseteq X \wedge X \in T_q} u(i_j, T_q). \tag{3}$$

It is obtained from T_2 in Table 1 that the utility of the itemset $\{AB\}$ can be computed as $u(AB, T_2) = u(A, T_2) + u(B, T_2) = 2 + 6 = 8$.

Definition 4. The utility of itemset X in a transaction database D is called as $u(X)$, a mathematical definition of which is as follows:

$$u(X) = \sum_{X \subseteq T_q \wedge T_q \in D} u(X, T_q). \tag{4}$$

It is obtained from Table 1 that the utility of itemset $\{B\}$ in database D can be computed as $u(B) = u(B, T_1) + u(B, T_2) + u(B, T_3) + u(B, T_4) + u(B, T_7) = 4 + 6 + 4 + 4 + 2 = 20$, $u(BD) = u(BD, T_1) + u(BD, T_3) + u(BD, T_4) + u(BD, T_7) = 14 + 24 + 14 + 7 = 59$.

Definition 5. The utility of a transaction in a transaction database D is called as $tu(T_q)$, a mathematical definition of which is as follows:

$$tu(T_q) = \sum_{i_j \in T_q} u(i_j, T_q). \quad (5)$$

It is obtained from Table 1 that there are 3 items in T_1 , which are B , D and E , so $tu(T_1) = u(B, T_1) + u(D, T_1) + u(E, T_1) = 4 + 10 + 9 = 23$. The transaction utility of other transactions in the running example is shown on the right side of Table 1, $tu(T_2) = 8$, $tu(T_3) = 33$, $tu(T_4) = 14$, $tu(T_5) = 17$, $tu(T_6) = 12$, $tu(T_7) = 17$.

Definition 6. The transaction-weighted utility of an itemset X in a transaction database D is called as $twu(X)$, a mathematical definition of which is as follows:

$$twu(X) = \sum_{X \subseteq T_q \wedge T_q \in D} tu(T_q). \quad (6)$$

It is obtained from Table 1 that item $\{A\}$ appears in T_2, T_5, T_7 , so the twu of the itemset $\{A\}$ is called as $twu(A) = tu(T_2) + tu(T_5) + tu(T_7) = 42$.

For the sake of taking both quantity and utility into account, the concept of skyline quantity-utility pattern mining (SQUPM) is listed below:

Definition 7. For itemset X and itemset Y , if $q(X) \geq q(Y)$ and $u(X) > u(Y)$ or $q(X) > q(Y)$ and $u(X) \geq u(Y)$, then the itemset X governs Y and it is represented as $X > Y$.

It is obtained from Table 1 that $q(A) = 5$, $q(B) = 10$, and $u(A) = 5$, $u(B) = 20$. It can be said that the item $\{B\} > \{A\}$ because $u(B) > u(A)$ and $q(B) > q(A)$.

Definition 8. When considering two-dimensional factor quantity and utility, an itemset is said to be SQUPM if it behaves as if it is not governed by other itemsets in the database.

3.2. Problem statement

Via the above definition, the problem of mining SQUPM can be formally defined as finding all sets of ungoverned points, namely SQUPs, from a quantitative database D .

For the running example in Table 1, the utility and quantity of $\{ED\}$ are computed as 69 and 6, the utility and quantity of $\{BD\}$ are computed as 59 and 7, and the utility and quantity of $\{D\}$ are computed as 55 and 11. Since any one of these three points cannot dominate the others, the sets $\{ED\}$, $\{BD\}$, and $\{D\}$ are eventually returned as skyline points.

4. Proposed Algorithms to Mine SQUPs

This paper proposes two depth-first search-based algorithms, FSKYQUP-Miner and FSKYQUP. This section consists of five subsections. In the first subsection, the utility-quantity-list structure is introduced, which is the basis of the algorithm proposed in this section. The proposed new structure is introduced in the second subsection. The third subsection introduces the pruning strategy used in the proposed algorithm. The fourth subsection describes the proposed algorithm in detail. The pseudo-code used will also be shown in this section, and the last part will be a step-by-step detailed mining process with the running example in Table 1.

4.1. Utility-quantity-list structure

In database D , calculate the TWU of each item and sort the TWU in ascending order using the \triangleright represent. Create a utility-quantity-list (UQL) [42] structure for each item, which is a quadruplet containing (tid , $quantity$, $utility$, $remaining\ utility$). Where tid represents the transaction ID containing this item, $quantity$ (abbreviated as $quan$) is to calculate the quantity of purchases of this item in the tid , $utility$ (abbreviated as $iutil$) is to calculate the utility of this item in this tid , and $remaining\ utility$ (abbreviated as $rutil$) is to calculate the sum of the utilities of the items appearing in this item in a tid after sorting by \triangleright .

Definition 9. *The mathematical definition of $rutil$ is as follows:*

$$rutil(X) = \sum_{i_j \subseteq T_q / X} iutil(i_j, T_q). \quad (7)$$

Assume in Table 1 that \triangleright indicates that the items are sorted in ascending order based on the transaction-weighted utility of each item; then the sorted items are $A \triangleright C \triangleright B \triangleright E \triangleright D$, and in transaction T_1 , the items that appear after item B after the sorting are item E and item D . As a result, in the transaction, $rutil = iutil(E, T_1) + iutil(D, T_1) = 9 + 10 = 19$.

4.2. Quantity maximum utility of the array (QMUA)

In this section, two efficient array structures for storing the maximum utility of quantities are proposed to record and update the maximum utility of itemsets, which largely reduces the search space for mining SQUPs.

Definition 10. (*Quantity Maximum Utility of the Array*) Define q_{max} to be the maximum quantity of all 1-itemsets in the database D .

If the quantity $q(X)$ of an itemset X ($1 \leq q(X) \leq q_{max}$) is equal to the original parameter i , then the QMUA structure will be defined as follows:

$$QMUA1(i) = \max\{u(X) \mid q(X) = i\}. \quad (8)$$

If the quantity of itemset X , $q(X)$ ($1 \leq q(X) \leq q_{max}$), is greater than or equal to the original parameter i , then the QMUA structure will be defined as follows:

$$QMUA2(i) = \max\{u(X) \mid q(X) \geq i\}. \quad (9)$$

Among them, unlike utilmax, the size of utilmax is set to $|D|$ and the size of *QMUA* is set to $q_{max} + 1$. *QMUA1* and *QMUA2* update in different ways; *QMUA1* only updates the utility of the set of items whose quantity is equal to i , whereas *QMUA2* updates the utility of all sets of items whose quantity is greater than or equal to i .

Definition 11. *Computing the quantity of an itemset X in the database D as $q(X) = q$. An itemset X is called a potential SQUP (PSQUP) if none of the other itemsets with quantity q has a utility greater than $u(X)$.*

Theorem 1. *If an itemset X is not a PSQUP, then it cannot be an SQUP. That is, $SQUP \subseteq PSQUP$.*

Proof. For $\forall X \notin PSQUPs$, there \exists an itemset Y that makes $q(Y) = q(X) \wedge u(Y) > u(X)$. According to Definition 7, it is known that Y dominates X . So $\forall X \notin SQUPs$.

This maximum utility array structure of quantity is used in the algorithms proposed in this paper to update the maximum utility of storing an equal quantity of itemsets using the *QMUA* structure, which can greatly reduce the space needed to search during the mining of SQUPs. In addition, the update method *QMUA1* corresponds to the FSKYQUP-Miner algorithm proposed in this paper, and the update method *QMUA2* corresponds to the FSKYQUP algorithm. For simplicity, the two update methods of *QMUA* are directly distinguished by the algorithm names in the following text.

4.3. Pruning strategies

In this portion, a pruning strategy for the initial phase of two algorithms and two pruning strategies in the mining phase will be presented.

Definition 12. *(minimum utility of SQUPs) In the original database D , the minimum utility of SQUPs is defined as the maximum utility of the largest quantity of the 1-itemset, a mathematical definition of which is as follows:*

$$MUSQ = \max\{u(X) | q(X) = q_{max}\}. \quad (10)$$

Where X is a 1-itemset in database D , and q_{max} is the maximum quantity of all 1-itemsets computed. Taking the running example, the quantity of item D in database D is $q(D) = 2 + 4 + 2 + 2 + 1 = 11$, so $q_{max} = 11$, and since $u(D) = 55$, $MUSQ = 55$.

Theorem 2. *An itemset X is a 1-itemset in the database. If the TWU of X is less than $MUSQ$, then this itemset X and all its extensions are not SQUPs.*

Proof. Assume Y is another 1-itemset in the database, and with $q(Y) = q_{max}$, $u(Y) = MUSQ$.

$\therefore u(X) \leq TWU(X) < MUSQ = u(Y) \wedge q(X) \leq q_{max} = q(Y)$.

Y dominates X .

$\therefore X \notin SQUPs$.

Assume that eX is an arbitrary extended set of items containing itemset X .

$\therefore u(eX) \leq TWU(X) < MUSQ = u(Y) \wedge q(eX) \leq q(X) \leq q_{max} = q(Y)$.

$\therefore eX$ is dominated by Y .

\therefore Any extension set of X is not SQUPs.

Therefore, according to Theorem 2, the set of items with a TWU smaller than $MUSQ$ can be directly pruned at the beginning of the algorithm, which greatly reduces the number of candidate sets. Furthermore, it is essential for the efficiency of the algorithm that the value of $MUSQ$ be assigned to the $QMUA$ array as the initial value.

Theorem 3. *An itemset X is not a SQUP if the sum of the iutil of the itemset X is less than the $QMUA$ value corresponding to $q(X)$.*

Proof. Suppose there exists an itemset Y with $q(Y) \geq q(X)$, $u(Y) = QMUA(q(X))$
 $\therefore X.sumiutil < QMUA(q(X)) = u(Y) \Rightarrow u(X) < u(Y)$
 since $q(X) \leq q(Y)$
 $\therefore Y$ dominates $X \Rightarrow X \notin SQUPs$.

According to the Theorem 3, it is possible to prune those terms whose sum of utilities of the itemset is less than $QMUA$, and these items are not SQUPs.

Theorem 4. *Any extension eX of X is not a SQUPs if the sum of iutil and rutil of the extension eX of the itemset X is less than the $QMUA$ value corresponding to $q(X)$.*

Proof. Assume that eX is an arbitrary extended set of items containing itemset X .

\therefore For \forall transaction T , it is possible to obtain:

$$\begin{aligned}
 eX \subseteq T &\Rightarrow (eX - X) = (eX/X) \Rightarrow (eX/X) \subseteq (T/X) \\
 \therefore u(eX, T) &= u(X, T) + u(eX - X, T) \\
 &= u(X, T) + u(eX/X, T) \\
 &= u(X, T) + \sum_{i_j \in eX/X} u(i_j, T) \\
 &\leq u(X, T) + \sum_{i_j \in T/X} u(i_j, T) \\
 &= u(X, T) + rutil(X, T) \\
 \therefore q(X) &\geq q(eX) \\
 \therefore eX.tids &\leq X.tids \\
 \therefore u(eX) &= \sum_{tid \in eX.tids} u(eX, T) \\
 &\leq \sum_{tid \in eX.tids} u(X, T) + rutil(X, T) \\
 &\leq \sum_{tid \in X.tids} u(X, T) + rutil(X, T) < QMUA(q(X)). \\
 \therefore \exists \text{ an itemset } Y &\text{ that makes } q(Y) \geq q(X) \geq q(eX), u(Y) = QMUA(q(X)) \geq u(eX) \\
 \therefore Y \text{ dominates } eX &\Rightarrow eX \notin SQUPs.
 \end{aligned}$$

According to the sum of *iutil* and *rutil* of the itemset in Theorem 4, it can be determined whether the extension of the itemset is PSQUPs or not. If the sum is less than $QMUA$, then the extension of this item is not SQUPs and the extension of this item can be cut to reduce the search space.

4.4. The proposed algorithm

This paper proposes two UQL structure-based algorithms, FSKYQUP-Miner and FSKYQUP, to find SQUPs quickly and efficiently. Both algorithms are based on depth-first search, and the itemsets are ordered among themselves. In addition, the difference between the two

algorithms is the different update methods, i.e., Algorithm 3 and Algorithm 4. The two algorithms and their related pseudo-code will be shown in the following.

Algorithm 1 is the pseudo-code of the proposed algorithms. Firstly, the database is scanned for the first time and the TWU of single items, the maximum quantity q_{max} and $MUSQ$ are calculated (line 1 of the algorithm). According to Theorem 2, if the TWU of the item is less than $MUSQ$, then this item i_j is deleted from the database and the database is pruned in the initial stage of this algorithm (lines 2–4 of the algorithm). Lines 5–6 sort the items in ascending order of TWU and reorganize the database. This loop creates a UQL structure for each item in the reorganized database (lines 7–11). Then the $QMUA$ is initialized to $MUSQ$, the maximum utility for the largest quantity of itemsets (lines 12–14 of the algorithm). It is worth noting that although the FSKYQUP-Miner algorithm and the FSKYQUP algorithm are updated in different ways, the initialization is the same. The **Search** function is then called to find all SQUPs (shown in detail in Algorithm 2). A set of SQUPs has finally been returned.

Algorithm 1 FSKYQUP-Miner/FSKYQUP algorithm

Require:

Original database D ; profit table.

Ensure:

A set of SQUPs.

```

1: Scan the database  $D$  and calculate the  $TWU$  of the item  $i_j$ ,  $q_{max}$ ,  $MUSQ$ ;
2: if  $TWU(i_j) < MUSQ$  then
3:   Delete  $i_j$  from original database  $D$ ;
4: end if
5: Sorting items  $i_j$  by  $TWU$  in ascending order;
6: Reorganization database;
7: for each  $T_q \in re-D$  do
8:   for each  $i_j \in T_q$  do
9:     Create  $i_j.UQLs$ ;
10:   end for
11: end for
12: for  $i = 1$  to  $q_{max}$  do
13:    $QMUA(i) = MUSQ$ ;
14: end for
15: set  $SQUPs = null$ ;
16: Search ( $null$ ,  $UQLs$ ,  $QMUA$ ,  $SQUPs$ );
17: return  $SQUPs$ ;

```

Algorithm 2 mines SQUPs based on depth-first search. For each itemset X belonging to the UQL (where UQL refers to the UQL corresponding to each extension of the prefix), if the sum of the utilities of the itemset X is greater than or equal to the $QMUA$ of $q(X)$, the itemset X may be SQUPs according to Theorem 3, and the **Judge** function is called to determine whether it is the final SQUP (lines 3–5). The subsequent lines 6–9 are to determine whether the extensions of the itemset X are psqups, and if the sum of $iutil$ and $rutil$ of X is greater than or equal to the $QMUA$ of $q(X)$, its extension eX is psqup.

According to Theorem 4, the extended UQL is established. Line 10 of the algorithm is a recursive call process until lines 7-8 no longer yield candidates.

Algorithm 2 Search

Require:

PUQL, UQL of the current prefix; *UQLs*, the UQL corresponding to each extension of the prefix; *QMUA*; *SQUPs*.

- 1: **for** $i = 0$ to *UQLs.size* **do**
- 2: $X = UQLs.get(i)$;
- 3: **if** $X.sumiutil \geq QMUA[q(X)]$ **then**
- 4: **Judge** ($X, QMUA, SQUPs$);
- 5: **end if**
- 6: **if** $X.sumiutil + X.sumrutil \geq QMUA[q(X)]$ **then**
- 7: **for each** $Y \triangleleft X$ **do**
- 8: $eXUQLs \leftarrow Create(PUQL, X, Y)$;
- 9: **end for**
- 10: **Search** ($X, eXUQLs, QMUA, SQUPs$);
- 11: **end if**
- 12: **end for**

Algorithm 3 and Algorithm 4 are pseudo-codes based on the FSKYQUP-Miner algorithm and FSKYQUP algorithm, respectively, to determine whether the itemset X is SQUPs. The difference between the two algorithms lies in the different update methods, which are explained in detail by Algorithm 3 as an example. If the *sumiutil* of an itemset X exceeds $QMUA[q(X)]$, it is necessary to investigate whether this itemset is an SQUP. In the first line of the algorithm, if Y is the first itemset in the SQUP set whose quantity is greater than X , i.e., $q(Y)$ is greater than $q(X)$, then the itemset X is an SQUP only when Y is equal to the empty set or when the utility of the itemset X is greater than the utility of the itemset Y . Then, insert X into the set of SQUPs. Otherwise, the itemset Y will dominate the itemset X and X must not be an SQUP. Then, update the value of $QMUA$ (line 4 of the algorithm). Next, determine whether, after inserting X , the set of SQUPs with a quantity less than X is an SQUP (lines 5-7 of the algorithm).

4.5. Illustrative example

Using the FSKYQUP-Miner algorithm as an example, the database used in the example is displayed in Table 1, and the profit table is displayed in Table 2. After the first scan of database D , it is calculated that $q(D) = q_{max} = 11$ and $MUSQ = 55$. The TWU of each item in the database is $\{A: 42, B: 95, C: 62, D: 104, E: 102\}$. Since $TWU(A) = 42 < MUSQ = 55$, according to Theorem 2, item A and all its extended itemsets are not SQUPs, and therefore, item A is removed from the database. The remaining items, after sorting in ascending order by TWU are $C \triangleright B \triangleright E \triangleright D$. According to this order, the original database will be reorganized, and the reorganized database is shown in Table 3.

Moreover, a UQL structure is created for each item as shown in Table 4. After initialization, $QMUA[1]$ to $QMUA[11]$ are assigned a value of 55.

Algorithm 3 Judge-FSKYQUP-Miner**Require:**

```

X, the PSQUP; QMUA; SQUPs.
1: find the first  $Y \in SQUPs$ , and  $q(Y) > q(X)$ ;
2: if  $Y == null$  or  $u(X) > u(Y)$  then
3:    $SQUPs \leftarrow X$ ;
4:    $QMUA[q(X)] = X.sumiutil$ ;
5:   for each itemset  $Y \in SQUPs$  do
6:     if  $q(X) = q(Y) \wedge u(X) > u(Y)$  or  $q(X) > q(Y) \wedge u(X) \geq u(Y)$  then
7:       delete  $Y$  from  $SQUPs$ ;
8:     end if
9:   end for
10: end if

```

Algorithm 4 Judge-FSKYQUP**Require:**

```

X, the PSQUP; QMUA; SQUPs.
1: find the first  $Y \in SQUPs$ , and  $q(Y) > q(X)$ ;
2: if  $Y == null$  or  $u(X) > u(Y)$  then
3:    $SQUPs \leftarrow X$ ;
4:   for  $n = q(X)$  down to 1 do
5:     if  $X.sumiutil > QMUA[n]$  then
6:        $QMUA[n] = X.sumiutil$ ;
7:     end if
8:   end for
9:   for each itemset  $Y \in SQUPs$  do
10:    if  $q(X) = q(Y) \wedge u(X) > u(Y)$  or  $q(X) > q(Y) \wedge u(X) \geq u(Y)$  then
11:      delete  $Y$  from  $SQUPs$ ;
12:    end if
13:  end for
14: end if

```

Firstly, starting from C , the UQL of C gives $q(C) = 7$, $iutil(C) = 14 < QMUA[7] = 55$, so C is not a SQUP, and since $iutil(C) + rutil(C) = 60 > QMUA[7]$, consider the extensions of C . The items that appear following C after sorting, and are connected to C at the beginning and end, form the extensions of C , which are CB , CE , and CD . Establish UQL for these items. Next, explore CB . Since $q(CB) = 3$, $iutil(CB) = 14 < QMUA[3]$, and $iutil(CB) + rutil(CB) = 48$, it is obvious that it is less than $QMUA[3]$, so CB and its extensions are not SQUPs. Since the algorithm is based on depth-first search, CE is checked next. According to Table 4, $q(CE) = 4$, $iutil(CE) = 29 < QMUA[4] = 55$, similarly, $iutil(CE) + rutil(CE) = 54 < QMUA[4]$, CE and its extensions are also not SQUPs. Next check CD , $q(CD) = 4$, $iutil(CD) = 33 < QMUA[4]$, and $iutil(CD) + rutil(CD) = 33 < QMUA[4]$, so CD and its extensions are not SQUPs. Follow the same steps to check B . Finally, the discovered candidate sets are $\{BED, BD, ED, D\}$, and all skyline quantity utility itemsets found are shown in Table 5. The final updated $QMUA$ of FSKYQUP-Miner algorithm is $\{55, 55, 55, 63, 55, 69, 59, 55, 55, 55, 55\}$ while the final updated $QMUA$ of FSKYQUP algorithm is $\{69, 69, 69, 69, 69, 69, 59, 55, 55, 55, 55\}$

Table 3. Reorganization database in the running instance

T_{ID}	Item and its quantity
T_1	B:2,E:3,D:2
T_2	B:3
T_3	C:3,B:2,E:1,D:4
T_4	B:2,D:2
T_5	E:2,D:2
T_6	C:3,E:2
T_7	C:1,B:1,E:2,D:1

respectively. The $QMUA$ array is obviously updated faster in the FSKYQUP algorithm than in the FSKYQUP-Miner algorithm. Coincidentally, within the example of this paper, the search spaces of proposed two algorithms are the same, as shown in Fig. 3.

Table 4. The utility-quantity-list structures of 1-items

(a) C				(b) B				(c) E				(d) D			
t_{id}	$quan$	$iutil$	$rutil$	t_{id}	$quan$	$iutil$	$rutil$	t_{id}	$quan$	$iutil$	$rutil$	t_{id}	$quan$	$iutil$	$rutil$
3	3	6	27	1	2	4	19	1	3	9	10	1	2	10	0
6	3	6	6	2	3	6	0	3	1	3	20	3	4	20	0
7	1	2	13	3	2	4	23	5	2	6	10	4	2	10	0
				4	2	4	10	6	2	6	0	5	2	10	0
				7	1	2	11	7	2	6	5	7	1	5	0

Table 5. Excavated SQUPs

SQUPs	quantity	utility
ED	6	69
BD	7	59
D	11	55

5. Experimental Evaluation

The FSKYQUP algorithm and the FSKYQUP-Miner algorithm proposed in this paper are compared with the SKYQUP algorithm and the SQU-Miner algorithm [42], two of the most advanced algorithms for mining SQUPs, in terms of runtime, memory consumption, the number of search itemsets, the resulting candidate sets, and the scalability of the algorithms. The experiments were conducted on a computer with an Intel (R) Core (TM) i3-8100 CPU @ 3.60 GHZ and 16 GB of RAM. The algorithms were written in Java and run on the idea compiler. To evaluate the algorithms' performance in many aspects,

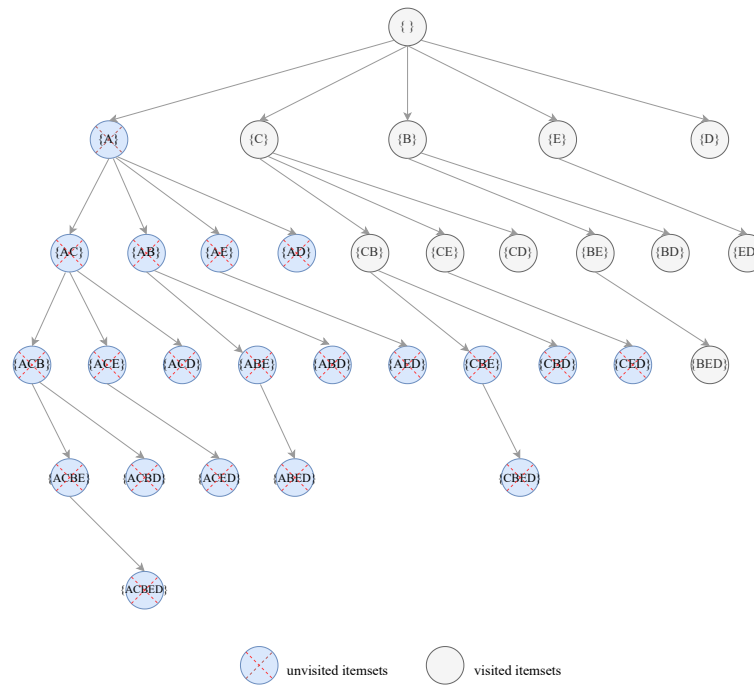


Fig. 3. Search space of proposed algorithms

the experiment was conducted on six different datasets, including five real-world datasets and one synthetic dataset. The following five real-world datasets were downloaded from SPMF [11]: namely Chess, Mushroom, Retail, Foodmart, and Ecommerce. A synthetic dataset T25I10D10K was generated using the utility quantity generator, obeying a Gaussian distribution. The parameters, such as the number of items, are shown in Table 6.

Table 6. Features of the datasets

Dataset	#Trans	#Items	Avg.Trans.Len	Max.Trans.Len	Type
Chess	3196	76	37	37	dense
Mushroom	8124	119	23	23	dense
Retail	88162	16470	10	76	sparse
Foodmart	4141	1559	4.42	14	sparse
Ecommerce	14975	3468	11.64	29	sparse
T25I10D10K	9976	929	24.77	63	dense

Table 6 details the following six characteristics of the six datasets: name of dataset, total number of transactions, number of items, average length of transactions, maximum length of transactions, and type of dataset (sparse or dense).

Runtime The runtimes of the proposed algorithms as well as the state-of-the-art SQU-Miner algorithm and SKYQUP algorithm for each of the datasets are shown in Fig. 4. The number of SQUPs mined for each dataset is shown in Table 7.

Table 7. The number of SQUPs

Dataset	#SQUPs
Chess	38
Mushroom	5
Retail	2
Foodmart	2
Ecommerce	1
T25I10D10K	1

In Fig. 4, generally speaking, the runtime of the FSKYQUP algorithm is shorter than that of the SKYQUP algorithm, and the runtime of the FSKYQUP-Miner algorithm is shorter than that of the SQU-Miner algorithm. In general, the runtimes of the proposed algorithms are shorter than that of SKYQUP and SQU-Miner. The FSKYQUP is better than the FSKYQUP-Miner, although the FSKYQUP-Miner is 0.02 seconds faster than the FSKYQUP on the Foodmart dataset. This is because the updating methods of $QMUA[q]$ differ. FSKYQUP updates based on the utilities of all item sets whose quantity is greater than or equal to q . Meanwhile, FSKYQUP-Miner only updates the utilities of item sets whose quantity is equal to q . Obviously, the FSKYQUP is more efficient at updating and produces fewer candidate item sets. For the dataset Chess, FSKYQUP is superior to the other three algorithms. The FSKYQUP-Miner runs for longer than the SKYQUP algorithm, but for shorter than the SQU-Miner due to the pruning strategy proposed in this paper. For the dataset Retail, the FSKYQUP and the FSKYQUP-Miner are 40 times faster than the SKYQUP and the SQU-Miner. This is because the Retail dataset is sparse, and in general, the items are not as closely related to each other as in a compact dataset. The $QMUA$ proposed in this paper is initialized based on the value of $MUSQ$, which means that the utility of most of the item sets does not reach the value for updating. As fewer candidates are generated, the runtime is shorter.

Memory We compared the memory usage of the proposed algorithms with that of the SQU-Miner algorithm and SKYQUP algorithm on each dataset. The experimental results are plotted in Fig. 5.

Fig. 5 shows that except for Mushroom and Foodmart, the proposed algorithms used less memory in mining SQUPs. In particular, the FSKYQUP and FSKYQUP-Miner on the Ecommerce and synthetic dataset T25I10D10K used roughly the same amount of memory, which is nearly 15 times less than the other two datasets. This is due to the efficient pruning strategy proposed in this paper, which narrows the search space. On the Foodmart dataset, the memory usage of the proposed algorithms is more than the existing algorithms, which is attributable to the creation of a list by the proposed algorithms for the storage of undesired candidates. The FSKYQUP-Miner uses the least memory on the Mushroom dataset. For the other two dense-type datasets, the FSKYQUP-Miner saves slightly more memory than the FSKYQUP.

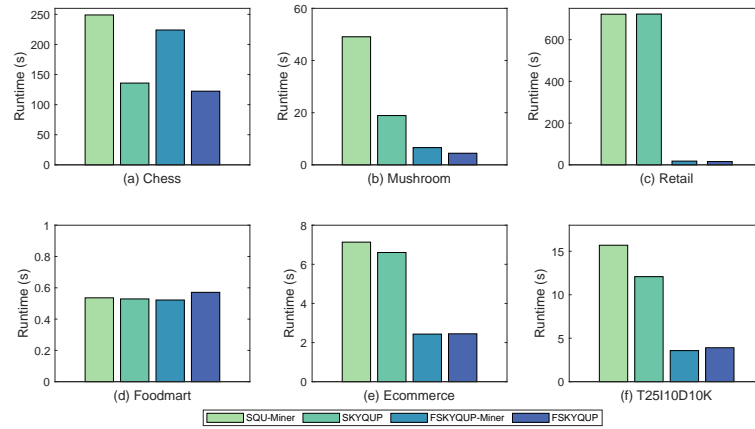


Fig. 4. Runtime on different datasets

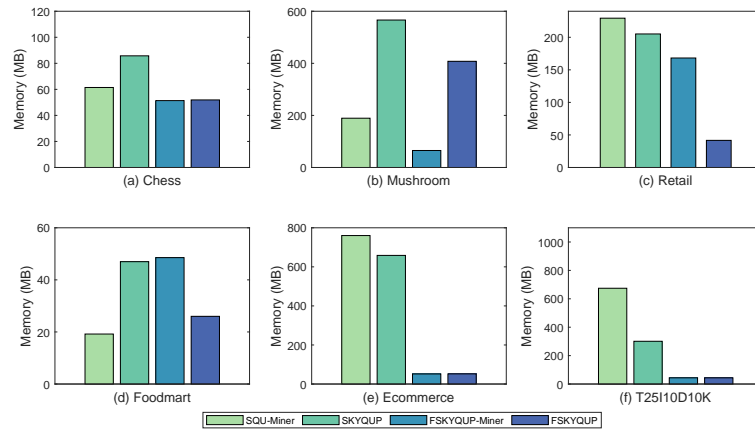


Fig. 5. Memory on different datasets

Search space We evaluated the size of the search space of the four algorithms for different datasets and plotted the results in Fig. 6.

Fig. 6 shows that FSKYQUP and FSKYQUP-Miner require less search space than the other two algorithms, which is due to the efficient pruning strategy proposed in this paper. The FSKYQUP requires the least search space, regardless of whether the dataset is sparse or dense. It is worth noting that on the Retail dataset, the number of search nodes of the SQU-Miner, SKYQUP, FSKYQUP-Miner, and FSKYQUP is respectively 26,803,198,632, 981,229,210, 71, and 71. The difference between the SQU-Miner and the proposed algorithms is eight orders of magnitude. On the Ecommerce dataset, the FSKYQUP and FSKYQUP-Miner are nearly 290 times worse than the other two algorithms in terms of search space. Similarly, on the T25I10D10K dataset, the FSKYQUP and FSKYQUP-Miner are nearly 230 times worse than the other two algorithms in terms of search space. On the datasets Retail, Ecommerce, and T25I10D10K, the FSKYQUP-

Miner requires the same search space as the FSKYQUP algorithm. These results indicate that the weaker the correlation between items in the dataset, the smaller the required search space.

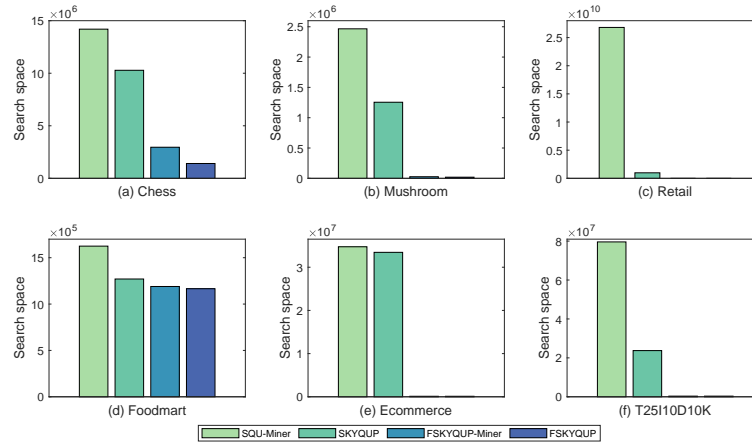


Fig. 6. Search space on different datasets

Candidate We evaluated the number of candidate item sets generated by the four algorithms for each dataset and plotted the results in Fig. 7.

Fig. 7 shows that the number of candidate sets generated by the proposed algorithms is smaller than the other two algorithms for all datasets except the Chess and Foodmart datasets. The FSKYQUP generated the least number of candidate sets for all datasets. For example, for the Retail dataset, the number of candidates generated by the four algorithms is respectively 1,472, 254, 7, and 4. The FSKYQUP generates 368 times fewer candidates than the SQU-Miner. For the reasons described in the first part of this section, the number of candidate sets for Chess and Foodmart generated by the FSKYQUP-Miner is larger than that of the SKYQUP but smaller than that of the SQU-Miner algorithm.

Scalability We conducted scalability experiments on the synthetic dataset, where the transactions of the dataset are set to 100k, 200k, 300k, 400k, and 500k. The performance is compared on each of these datasets in four aspects: runtime, memory usage, search space size, and the number of generated candidate sets. The experimental results are shown in Fig. 8.

The proposed algorithms compare favorably with the state-of-the-art SQU-Miner and SKYQUP algorithms in terms of runtime, memory usage, the size of the search space, and the number of candidate sets generated as the dataset increases. Fig. 8(a) compares the execution times of the four algorithms across the five synthetic data sets. Running the SQU-Miner algorithm takes a long time, and the runtime becomes longer when there are more datasets. The proposed algorithms have similar runtimes and good scalability,

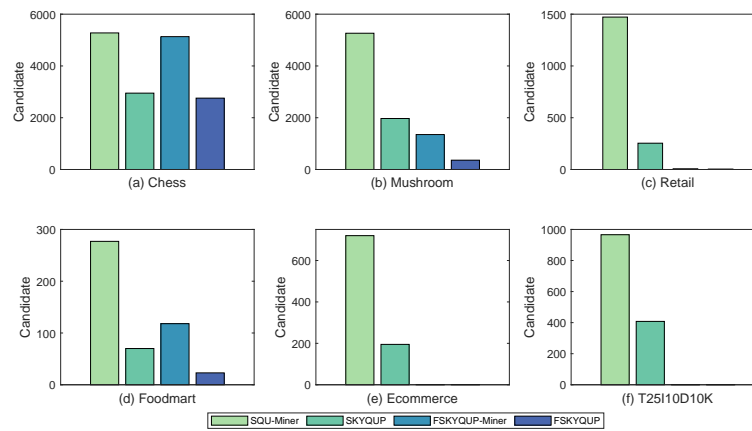


Fig. 7. Candidate on different datasets

as the runtime grows gradually as the dataset increases. Fig. 8(b) displays the memory usage of the four algorithms for the five synthetic datasets. The largest memory consumer is SQU-Miner, while FSKYQUP is the smallest. As the dataset increases, the proposed algorithms have good scalability in terms of memory usage. The search space required to run the four algorithms on different-sized datasets is depicted in Fig. 8(c). It is clear from the figure that the SQU-Miner requires a vast search space, the SKYQUP requires a smaller but still large search space, and the FSKYQUP and FSKYQUP-Miner require the smallest search spaces. Fig. 8(d) depicts the number of candidate sets generated for each dataset: the proposed algorithms generate the least candidate sets, followed by the SKYQUP, while the SQU-Miner generates the most candidate sets. These results indicate that the proposed algorithms offer good scalability in terms of runtime, memory usage, search space, and the number of candidate sets.

6. Conclusion

With the advent of the information age, relying solely on the support of FIs and HUIs is no longer good enough to support decision-making, so people prefer to take into account both the frequency and utility of the work. In contrast, quantity also plays a crucial role in the decision-making process. This paper proposes two methods that do not require a user-defined threshold: FSKYQUP-Miner and FSKYQUP. Both of these approaches are based on UQL and obtain a set of uncontrolled nodes. We also propose a more effective pruning method which eliminates undesired candidates in the initial stage of the algorithm, thus greatly narrowing the search scope. Extensive experiments on real-world and synthetic datasets verified that the proposed methods scale well in terms of runtime, memory usage, search space, and the number of candidate sets. These results indicate that the proposed algorithms are well-suited to supermarket applications. As big data continues to advance, in the future, it would be fruitful to explore SQUPs with other architectures, such as the MapReduce or Spark framework. The proposed algorithms would also benefit from

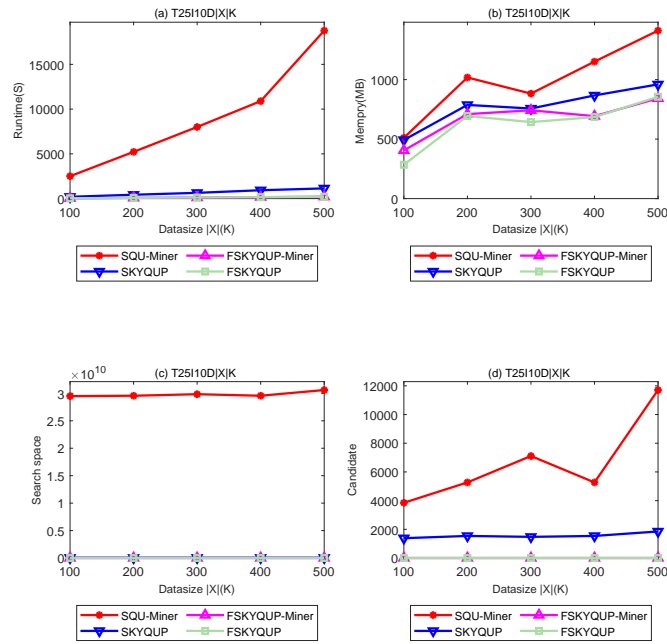


Fig. 8. Scalability on different datasets

additional pruning strategies to simplify the structure and thus mine SQUPs even more effectively.

Acknowledgments. This research is supported by Shandong Provincial Natural Science Foundation (ZR201911150391).

References

1. Afrati, F.N., Koutris, P., Suci, D., Ullman, J.D.: Parallel skyline queries. *Theory of Computing Systems* 57(4), 1008–1037 (2015)
2. Agrawal, R., Imielinski, T., Swami, A.: Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering* 5(6), 914–925 (1993)
3. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. pp. 207–216 (1993)
4. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I., et al.: Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining* 12(1), 307–328 (1996)
5. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: *Proceedings of the 20th International Conference on Very Large Databases*. vol. 1215, pp. 487–499. Citeseer (1994)
6. Ahmed, C.F., Tanbeer, S.K., Jeong, B.S., Lee, Y.K.: Efficient tree structures for high utility pattern mining in incremental databases. *IEEE Transactions on Knowledge and Data Engineering* 21(12), 1708–1721 (2009)

7. Ahmed, U., Lin, J.C.W., Srivastava, G., Yasin, R., Djenouri, Y.: An evolutionary model to mine high expected utility patterns from uncertain databases. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5(1), 19–28 (2020)
8. Borzsony, S., Kossmann, D., Stocker, K.: The skyline operator. In: *Proceedings 17th International Conference on Data Engineering*. pp. 421–430. IEEE (2001)
9. Chan, C.Y., Jagadish, H., Tan, K.L., Tung, A.K., Zhang, Z.: Finding k-dominant skylines in high dimensional space. In: *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*. pp. 503–514 (2006)
10. Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with presorting. In: *ICDE*. vol. 3, pp. 717–719 (2003)
11. Fournier-Viger, P., Lin, J.C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., Lam, H.T.: The spmf open-source data mining library version 2. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 36–40. Springer (2016)
12. Fournier-Viger, P., Wu, C.W., Tseng, V.S.: Mining top-k association rules. In: *Canadian Conference on Artificial Intelligence*. pp. 61–73. Springer (2012)
13. Fournier-Viger, P., Wu, C.W., Zida, S., Tseng, V.S.: Fhm: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In: *International Symposium on Methodologies for Intelligent Systems*. pp. 83–92. Springer (2014)
14. Gan, W., Lin, J.C.W., Fournier-Viger, P., Chao, H.C., Hong, T.P., Fujita, H.: A survey of incremental high-utility itemset mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8(2), e1242 (2018)
15. Goyal, V., Sureka, A., Patel, D.: Efficient skyline itemsets mining. In: *Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering*. pp. 119–124 (2015)
16. Grahne, G., Zhu, J.: Fast algorithms for frequent itemset mining using fp-trees. *IEEE Transactions on Knowledge and Data Engineering* 17(10), 1347–1362 (2005)
17. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. *ACM Sigmod Record* 29(2), 1–12 (2000)
18. Imielinski, T., Mannila, H.: A database perspective on knowledge discovery. *Communications of the ACM* 39(11), 58–64 (1996)
19. Kossmann, D., Ramsak, F., Rost, S.: Shooting stars in the sky: An online algorithm for skyline queries. In: *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. pp. 275–286. Elsevier (2002)
20. Krishnamoorthy, S.: Pruning strategies for mining high utility itemsets. *Expert Systems with Applications* 42(5), 2371–2381 (2015)
21. Kung, H.T., Luccio, F., Preparata, F.P.: On finding the maxima of a set of vectors. *Journal of the ACM (JACM)* 22(4), 469–476 (1975)
22. Lin, C.W., Hong, T.P., Lu, W.H.: An effective tree structure for mining high utility itemsets. *Expert Systems with Applications* 38(6), 7419–7424 (2011)
23. Lin, J.C.W., Yang, L., Fournier-Viger, P., Hong, T.P.: Mining of skyline patterns by considering both frequent and utility constraints. *Engineering Applications of Artificial Intelligence* 77, 229–238 (2019)
24. Liu, J., Wang, K., Fung, B.C.: Direct discovery of high utility itemsets without candidate generation. In: *2012 IEEE 12th International Conference on Data Mining*. pp. 984–989. IEEE (2012)
25. Liu, M., Qu, J.: Mining high utility itemsets without candidate generation. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. pp. 55–64 (2012)
26. Liu, Y., Liao, W.k., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. pp. 689–695. Springer (2005)
27. Luna, J.M., Fournier-Viger, P., Ventura, S.: Frequent itemset mining: A 25 years review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9(6), e1329 (2019)

28. Pan, J.S., Lin, J.C.W., Yang, L., Fournier-Viger, P., Hong, T.P.: Efficiently mining of skyline frequent-utility patterns. *Intelligent Data Analysis* 21(6), 1407–1423 (2017)
29. Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive skyline computation in database systems. *ACM Transactions on Database Systems (TODS)* 30(1), 41–82 (2005)
30. Park, J.S., Chen, M.S., Yu, P.S.: An effective hash-based algorithm for mining association rules. *Acm Sigmod Record* 24(2), 175–186 (1995)
31. Podpecan, V., Lavrac, N., Kononenko, I.: A fast algorithm for mining utility-frequent itemsets. *Constraint-Based Mining and Learning* p. 9 (2007)
32. Song, W., Zheng, C.: Sfu-ce: Skyline frequent-utility itemset discovery using the cross-entropy method. In: *Intelligent Data Engineering and Automated Learning–IDEAL 2021: 22nd International Conference, IDEAL 2021, Manchester, UK, November 25–27, 2021, Proceedings* 22. pp. 354–366. Springer (2021)
33. Song, W., Zheng, C., Fournier-Viger, P.: Mining skyline frequent-utility itemsets with utility filtering. In: *Pacific Rim International Conference on Artificial Intelligence*. pp. 411–424. Springer (2021)
34. Srivastava, G., Lin, J.C.W., Pirouz, M., Li, Y., Yun, U.: A pre-large weighted-fusion system of sensed high-utility patterns. *IEEE Sensors Journal* 21(14), 15626–15634 (2020)
35. Tan, K.L., Eng, P.K., Ooi, B.C., et al.: Efficient progressive skyline computation. In: *VLDB*. vol. 1, pp. 301–310 (2001)
36. Tseng, V.S., Shie, B.E., Wu, C.W., Philip, S.Y.: Efficient algorithms for mining high utility itemsets from transactional databases. *IEEE Transactions on Knowledge and Data Engineering* 25(8), 1772–1786 (2012)
37. Tseng, V.S., Wu, C.W., Fournier-Viger, P., Philip, S.Y.: Efficient algorithms for mining top-k high utility itemsets. *IEEE Transactions on Knowledge and Data Engineering* 28(1), 54–67 (2015)
38. Tseng, V.S., Wu, C.W., Shie, B.E., Yu, P.S.: Up-growth: an efficient algorithm for high utility itemset mining. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 253–262 (2010)
39. Wang, K., Wu, J.M.T., Cui, B., Lin, J.C.W.: Revealing top-k dominant individuals in incomplete data based on spark environment. In: *International Conference on Genetic and Evolutionary Computing*. pp. 471–480. Springer (2021)
40. Wu, J.M.T., Lin, J.C.W., Tamrakar, A.: High-utility itemset mining with effective pruning strategies. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13(6), 1–22 (2019)
41. Wu, J.M.T., Liu, S., Lin, J.C.W.: Efficient uncertain sequence pattern mining based on hadoop platform. *Journal of Circuits, Systems and Computers* (2022)
42. Wu, J.M.T., Teng, Q., Srivastava, G., Pirouz, M., Lin, J.C.W.: The efficient mining of skyline patterns from a volunteer computing network. *ACM Transactions on Internet Technology (TOIT)* 21(4), 1–20 (2021)
43. Wu, J.M.T., Zhan, J., Lin, J.C.W.: An aco-based approach to mine high-utility itemsets. *Knowledge-Based Systems* 116, 102–113 (2017)
44. Yao, H., Hamilton, H.J.: Mining itemset utilities from transaction databases. *Data & Knowledge Engineering* 59(3), 603–626 (2006)
45. Yao, H., Hamilton, H.J., Butz, C.J.: A foundational approach to mining itemset utilities from databases. In: *Proceedings of the 2004 SIAM International Conference on Data Mining*. pp. 482–486. SIAM (2004)
46. Yeh, J.S., Li, Y.C., Chang, C.C.: Two-phase algorithms for a novel utility-frequent mining model. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. pp. 433–444. Springer (2007)
47. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: Parallel algorithms for discovery of association rules. *Data Mining and Knowledge Discovery* 1(4), 343–373 (1997)
48. Zaki, M.J.: Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* 12(3), 372–390 (2000)

49. Zida, S., Fournier-Viger, P., Lin, J.C.W., Wu, C.W., Tseng, V.S.: Efim: a highly efficient algorithm for high-utility itemset mining. In: Mexican International Conference on Artificial Intelligence. pp. 530–546. Springer (2015)

Jimmy Ming-Tai Wu received the Ph.D. degree with major in computer science and engineering from National Sun Yat-sen University, Kaohsiung, Taiwan. He is currently an Associate Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology. He was an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He was with an IC design company in Taiwan as a Firmware Developer and Information Technology Manager for two years. He was also a Research Scholar with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan, and with the Department of Computer Science, College of Engineering, University of Nevada, Las Vegas. His current research interests include data mining, big data, cloud computing, artificial intelligence, evolutionary computation, machine learning, and deep learning.

Ranran Li is currently pursuing the M.S. degree in Shandong University of science and technology, Qingdao, China. Her current research interests include data mining and big data.

Pi-Chung Hsu received his doctor degree in information engineering at national Sun Yat-sen University Taiwan in 2003. Now he is the associate professor at Shu-Te University Taiwan. His research directions include computer graphics, implicit surfaces and solid modeling. Dr. HSU may be reached at pichung@stu.edu.tw

Mu-En Wu is an Associate Professor at Department of Information and Finance Management at National Taipei University of Technology, Taiwan. Dr. Wu received his Ph.D. degree with major in computer science from National Tsing Hua University, Taiwan, in 2009. After that, he joined Institute of Information Science, Academia Sinica at Taipei City, Taiwan as a postdoctoral fellow during 2009–2014. During February 2014 to July 2017, he served as an assistant professor of Department of Mathematics at Soochow University. He has a wide variety of research interests covering cryptography, information theory, prediction market, money management, and financial data analysis. He has published more than 100 research papers in referred journals and international conferences

Received: June 15, 2022; Accepted: March 27, 2023.

Probabilistic Reasoning for Diagnosis Prediction of Coronavirus Disease based on Probabilistic Ontology

Messaouda Fareh, Ishak Riali, Hafsa Kherbache, and Marwa Guemmouz

LRDSI Laboratory, Faculty of sciences

University Blida1

Soumaa, B.P 270, Blida, Algeria

{farehm, ishakriali, kherbachehaf, guemmouzm}@gmail.com

Abstract. The novel Coronavirus has been declared a pandemic by the World Health Organization (WHO). Predicting the diagnosis of COVID-19 is essential for disease cure and control. The paper's main aim is to predict the COVID-19 diagnosis using probabilistic ontologies to address the randomness and incompleteness of knowledge. Our approach begins with constructing the entities, attributes, and relationships of COVID-19 ontology, by extracting symptoms and risk factors. The probabilistic components of COVID-19 ontology are developed by creating a Multi-Entity Bayesian Network, then determining its components, with the different nodes, as probability distribution linked to various nodes. We use probabilistic inference for predicting COVID-19 diagnosis, using the Situation-Specific Bayesian Network (SSBN). To validate the solution, an experimental study is conducted on real cases, comparing the results of existing machine learning methods, our solution presents an encouraging result and, therefore enables fast medical assistance.

Keywords: COVID-19, Probabilistic Ontology, Multi-Entity Bayesian Networks, Uncertainty, Reasoning

1. Introduction

December 31, 2019, is the day of the appearance of COVID-19 for the first time in the Wuhan region, China notified the outbreak to the World Health Organization [49]. Since 2019, the world has been affected by coronavirus-19 (COVID-19), which has caused many deaths. The world does not realize the importance of this disease and its impact on future life, until March 11, 2020, the COVID-19 epidemic is declared a pandemic by the World Health Organization (WHO) [36].

The COVID-19 virus is a highly contagious respiratory disease that has spread rapidly around the world since it was first reported in China in late December 2019. The early detection of patients at risk to develop critical illness may aid in delivering proper care and reduce mortality [46]. Current literature has already identified several risk factors.

Widely reported statistics on COVID-19 over the globe need to consider the uncertainty of the data and possible explanations for this uncertainty. The professional environment of medical practice is characterized by great complexity, many grey areas, and uncertainty as an essential component of all levels of patient care.

The term "uncertainty" is intended to encompass a variety of aspects of imperfect knowledge, including incompleteness, vagueness, ambiguity, and others¹. Medical infor-

¹ <https://www.w3.org/2005/Incubator/urw3/group/draftReport.html>

mation can be imprecise, partial, vague, or imperfect as some lab results may be missing from the feature set or due to the patient's inability to answer properly [3].

We present an example of uncertainty in medical applications cited in [41]. In a simple scenario, we could try to model the knowledge that patients showing a running nose and complaining of feeling light-headed have a common cold; we can use an axiom like ($\{RunNose, LightHead\}, Cold$). This rule would be correct most of the time but ignores the fact that there exist many different maladies—some of them potentially serious—that share these same symptoms and require additional interventions.

Uncertainty pervades every activity related to health care, it prompts patients to seek care and stimulates medical intervention. The inability to abolish uncertainty, furthermore, creates difficult challenges for clinicians and patients [24]. Therefore the diagnosis is particularly doubtful, which can result in inaccurate or completely false diagnoses [23]. This is why the medical field gives great importance to the factor of uncertainty. Hence, it seems very interesting to design and implement automated healthcare systems that consider these challenges and ease the diagnosis task for the doctors [45].

Many research papers on COVID-19 used Machine Learning (ML) methods and ontologies as a knowledge base. Some contributions focus on applying ML in detecting the diagnosis of COVID-19, whereas others exploit data for predicting diagnosis.

Upon analyzing previous works, it has been observed that they use classical ontologies. ([50], [37], [47], [25], [6], [29] and [32]), these studies cannot deal with uncertainty, and they only have precise concepts and relationships. A commonly mentioned limitation of classical ontology languages, especially within the context of knowledge representation, is their inability to model or handle uncertainty [41]. Indeed, notice that we use the axioms in an ontology as absolute information, and the consequences follow (or not) from these axioms. This leaves no space for statements which are not completely certain [41].

Despite all the advances made in the field of the semantic web, problems associated with data uncertainty and ambiguity still need to be solved in the knowledge management of a real domain. So uncertainty is inevitable when we model most application domains, like in the medical field, the symptoms are subjective and therefore imprecise and incomparable. In addition, concepts and relationships may not be described by the description logic language. One of the main flaws of classical ontology is the inability to represent and reason under uncertainty. This uncertainty can manifest during the prediction of a patient with COVID-19.

Bayesian Networks (BNs) are proposed for decision support, and they allow probabilistic reasoning. They are a probability-based inference model, increasingly used in the medical domain as a method of knowledge representation for reasoning under uncertainty for a wide range of applications, including disease diagnosis [16].

Different from other techniques, BNs have been used to interpret and explain COVID-19 data, BNs integrate multiple sources of data in a single model that provides a statistical estimates model. This last presents the uncertainty concerning mechanisms that generate the data [34].

Many papers research on COVID-19 used Bayesian networks for handling uncertainty related to COVID-19 diagnosis, we cite [18], [7], [18], [43], [34], [51], these studies propose a Bayesian network model to predict the diagnosis of COVID-19 according to different purposes.

In this paper, we have combined Bayesian networks with classical ontologies to harness the power of Bayesian networks, based on probability theory and the inference and probabilistic reasoning mechanisms proposed by BNs, on the knowledge base represented by a classical ontology, which gives us a probabilistic ontology.

Probabilistic ontologies based on Multi-Entity Bayesian Networks (MEBN) [8] allow us to describe uncertain knowledge in a reasoned and structured way and to model uncertainty using probability factors and causal links. These methods have been used in the medical field through several languages such as PR-OWL, thus giving way to the modeling of probabilistic knowledge through ontologies. These are used to describe domain knowledge, and the uncertainty associated with this knowledge in a structured and shareable way, in a format that can be read and processed by a computer.

Bayesian methods allow the system, on the one hand, to integrate expert knowledge with machine learning, to provide understandable models, and on the other hand, to provide, in a natural manner, probabilistic predictions making it possible to keep account for uncertainty when making decisions.

Problems: The confronted problems are as follows:

- How to predict the diagnosis of COVID-19 in an uncertain environment; and how to use Bayesian networks and ontologies to manage associated uncertainty for decision support.
- How to integrate the knowledge of an expert in a probabilistic model to achieve machine learning to use probabilistic inference for predicting diagnosis.
- How to model the uncertain knowledge of the COVID-19 diagnosis, using probabilistic ontologies, which combine the two models of Bayesian networks and classical ontologies.

Contribution: our main contributions in this paper are summarized as follows:

- Development of a probabilistic ontology for COVID-19, containing important concepts, such as symptoms and risk factors related to COVID-19, and probabilistic knowledge.
- Predicting COVID-19 diagnosis based on probabilistic inference, using Situation-Specific Bayesian Network.
- Collecting real anonymous data of patients for constructing dataset used for learning.
- Our proposed system's results are promising and can be exploited in real environments.

Motivation: Bayesian networks are a very powerful formalism; on the one hand, it deals with random uncertainty through probability theory which represents a very efficient framework for the management of degrees of influence of an attribute in another and its propagation throughout the Bayesian network and on the other hand, its ability to make inferences even with missing data, for example for a patient without doing the COVID test, and with a small number of symptoms or risk factors, we can reason about his observations to help for diagnosis. In addition to these advantages, Multi-Entity Bayesian Networks are more expressive than classical BNs because they integrate first-order logic with Bayesian probability and are more flexible in terms of inferences.

Among the limits of classical ontologies is the inability to reason under uncertainty, which leads us to build a probabilistic ontology to represent the knowledge base and the treatment of uncertainty provided by MEBN.

The remainder of this paper is organized as follows: Section 2 explores the background knowledge. Section 3 describes the related works, it presents a comprehensive review of related studies on COVID-19. Section 4 is devoted to the proposal of probabilistic ontology. An experiment and an evaluation were conducted to validate the proposed approach presented in Section 5. We draw conclusions and discuss future work in Section 6.

2. Background

2.1. Uncertain ontological knowledge

The Uncertainty Reasoning for the World Wide Web Incubator Group (URW3-XG)² of World Wide Web Consortium (W3C)³ in his last report declares that Uncertainty is an intrinsic feature of many of the required tasks, and a full realization of the World Wide Web as a source of processable data and services demands formalisms capable of representing and reasoning under uncertainty. Although it is possible to use semantic markup languages such as OWL to represent qualitative and quantitative information about uncertainty, there is no established foundation for doing so. Therefore, each developer must come up with his/her own set of constructs for representing uncertainty. This is a major liability in an environment so dependent on interoperability among systems and applications.

The URW3-XG propose a high-level ontology to present various types of uncertainty: **ambiguity, empirical, randomness, vagueness, inconsistency and incompleteness.**

The new pandemic typically poses a challenge to data analytics, considering its limited information and the geographical and temporal evolution of the recent epidemic. Therefore, an accurate model for predicting the future behavior of a pandemic becomes challenging due to uncertainty [20]. In this context, we are interested in creating a probabilistic ontology to deal with this uncertainty.

2.2. Bayesian Networks

One of the most promising approaches to deal with uncertainty is Bayesian Networks (BN) [40]. A Bayesian Network N is a triplet (V, A, P) , where:

- V is a set of variables,
- A is a set of arcs, which together with V constitutes a direct acyclic graph $G=(V, A)$,
- P is a set of conditional probabilities of all variables given their respective parents.

The joint distribution for a BN is equal to the product of $P(\text{node}/\text{parents}(\text{node}))$ for all nodes. Bayesian networks are powerful models that compactly represent the joint probability distribution defined by the set of variables under study [12].

² <https://www.w3.org/2005/Incubator/urw3/group/draftReport.html>

³ <https://www.w3.org>

2.3. Multi-Entity Bayesian Networks (MEBN)

Multi-Entity Bayesian Networks [31] integrate first-order logic with Bayesian probability. MEBN logic expresses probabilistic knowledge as a collection of MEBN fragments (MFrag) organized into MEBN Theories (MTheories). Formally, an MFrag F is defined as :

$$F = (C, I, R, G, D) \text{ Where}$$

- C is a finite set of values a context can take form as a value.
- I is a set of input random variables
- R is a finite set of resident random variables
- G is a directed acyclic graph representing the dependency between input random variables and resident random variables conditional on context random variables in one-to-one correspondence.
- D is a set of local conditional probability distributions where each member of R has its own conditional probability distribution in set D .
- Sets C , I , and R are pairwise disjoint

2.4. Probabilistic Ontology PR-OWL

PR-OWL was developed as an extension enabling OWL ontologies to represent complex Bayesian probabilistic models. From [39], a probabilistic ontology is an explicit, formal knowledge representation that expresses knowledge about a domain of application. This includes:

- Types of entities that exist in the domain;
- Properties of those entities;
- Relationships among entities;
- Processes and events that happen with those entities;
- Statistical regularities that characterize the domain;
- Inconclusive, ambiguous, incomplete, unreliable, and dissonant knowledge related to entities of the domain; and
- Uncertainty about all the above forms of knowledge; where the term entity refers to any concept (real or fictitious, concrete or abstract) that can be described and reasoned about within the domain of application.

2.5. Reasoning under uncertainty

The term "uncertainty reasoning" is meant to denote the full range of methods designed for representing and reasoning with knowledge when Boolean truth values are unknown, unknowable, or inapplicable. Commonly applied approaches to uncertainty reasoning include probability theory, fuzzy logic, subjective logic, Dempster-Shafer theory, and numerous other methodologies [13].

In the context of probabilistic reasoning using Bayesian networks, these networks are primarily used for performing probabilistic inference, which involves generating probabilistic statements regarding the variables depicted within the network.

Two types of inference can be performed on a Bayesian network: exact and approximate. Exact inference leverages the conditional independences within the network to calculate an exact posterior probability for each inference. Examples of algorithms that fall under exact inference include Bucket Elimination [4], Message Passing [40], and Junction Tree [28]. On the other hand, approximate methods such as Markov Chain Monte Carlo and variational methods are used for the second category. We refer to algorithms such as Likelihood Weighting [21], Backward Sampling [22], and Self Importance [48], which estimate probabilities by drawing from the set of possible combinations of network variables' states multiple times.

In MEBN, the process of inference is done by constructing Situation Specific Bayesian Network (SSBN) [14], which is a Bayesian network constructed by creating and combining instances of the MFrag in the MTheory. When each MFrag is instantiated, instances of its random variables are created to represent known background information, observed evidence, and queries of interest to the decision maker. The process of inference starts with a generative MTheory, adds a set of finding MFrag representing problem-specific information, and specifies the target nodes for our query. The process of MEBN inference [14] consists of:

1. Construct an SSBN, which is constructed by creating and combining instances of the MFrag in the generative MTheory.
2. When each MFrag is instantiated, instances of its random variables are created to represent known background information, observed evidence, and queries.

3. Related works

After studying several related works that deal with uncertainty about knowledge in the medical field, we have classified them into several categories: medical ontology-based approaches, Bayesian network-based approaches, and probabilistic ontologies-based approaches.

Various predictive models based on machine learning have been proposed in the literature to help reduce the load of COVID-19 on healthcare systems, we cite for example: [35], [52], [27], [38], [42], [1], [2], [33] and [5], these studies do not deal with uncertainty, but a few studies use a Bayesian Network model for handling uncertainty and predicting a diagnosis of COVID-19.

3.1. Bayesian network-based approaches

Bayesian Networks can be useful for decision-makers to make sense of complex information using a probabilistic approach.

The authors [18] present a Bayesian network that provides the basis for a practical CTA (Contact Tracing Apps) solution that does not compromise privacy. Users of the model can provide personal information about relevant risk factors, symptoms, and recent social interactions. The model then provides them feedback about the likelihood of the presence of asymptomatic, mild or severe COVID-19.

In the paper [7] the authors propose a Bayesian network to predict the probability of COVID-19 infection, based on a patient's profile, the structure and prior probabilities have

been amalgamated from knowledge. This paper constructs the solution of the Bayesian network created by [18] and the work by [43] for predicting COVID-19 status coupled with eventual prognoses. The network takes an input of observable symptoms and risk factors to produce a personalized probability score for disease status.

In [34], the authors use a Bayesian Network model to estimate the COVID-19 infection prevalence rate (IPR) and infection fatality rate (IFR) for different countries and regions, where relevant data are available. This combines multiple sources of data in a single model.

Terwangne et al. [51] developed a model named COVID-19 EPI-SCORE to predict the severity classification of patients hospitalized with COVID-19. The purpose of the study is to assess the COVID-19 severity classification. In this approach, Bayesian network analysis was used to build the model for predicting the accuracy of severity classification.

3.2. Ontologies-based approaches

This section provides a review of recent articles that have discussed the significant contributions using ontologies-based approaches.

In [37], the author proposed an approach adopted in their study that employs the use of an improved Case-Based Reasoning (CBR) model for reasoning tasks in the classification of suspected cases of COVID-19. Knowledge representation in the proposed framework was achieved using an ontology-based knowledge formalization technique.

In [47] the authors developed an ontology representing major novel coronavirus (SARS-CoV-2) entities, Ontology has a strong scope on chemical entities suited for drug repurposing. COVID-19 Ontology is used to attain the semantic interoperability and mapping between various entities and relationships in the COVID Knowledge SuperGraph.

The paper [6] discusses the advantages of using ontologies for describing and modeling psychological research questions. The authors use and apply CCOnto as a theoretical and formal description system to categorize psychological factors that influence student behavior during the COVID-19 situation.

In [29] the proposed solution is to design a COVID-19 ontology model, as well as an alert system combining vital sign parameters and symptom parameters, to identify suspected early cases of chronic obstructive pulmonary disease patients with COVID-19.

Furthermore, [25] presented the Coronavirus Infectious Disease Ontology (CIDO) which is a community-based ontology for coronavirus disease knowledge and data integration, sharing, and analysis within the field of informatics.

In [32] the authors proposed a project nominated “Ontological and bio-informatics analysis of anti-coronavirus drugs and their implication for three drug re-purposing against COVID-19”. The authors fixed that their ontology-based bio-informatics strategy lead to drug prediction for COVID-19.

3.3. Probabilistic Ontology-based approaches

Various approaches have been made to represent uncertainty in ontology; one of them is the probabilistic ontology based on the Bayesian network.

One of the reasons that make the research in ontology languages focusing on deterministic approaches has limited expressiveness of knowledge uncertainty [44]. There is

current research based on extending OWL and combined with Bayesian networks, so, it can represent uncertain knowledge, and especially, probabilistic information [17].

Until today, there is no standard language for representing probabilistic ontologies, the W3C consortium recommends using BNs for semantic web uncertainty processing⁴. There are some contributions that have been proposed in the literature to deal with uncertain knowledge in ontologies, we cite some examples:

- BayesOWL [15]: the authors propose a framework to model uncertainty in semantic web ontologies based on Bayesian networks, it is a probabilistic generalization of OWL language. It provides a set of rules for the direct translation of OWL ontology into a Bayesian network, it also provides a method for incorporating probability constraints while constructing a Bayesian network.
- OntoBayes: Yang and Calmet [53] present an integration of the web ontology language OWL with Bayesian networks, called OntoBayes. This model makes use of probability and dependency-annotated OWL to represent uncertain information in BN structures
- In [19], the author proposed an ontology-based approach for constructing Bayesian networks, the proposed approach supports the construction of the structure of Bayesian networks and the conditional probabilistic tables of nodes of BN. This method enables the modification of Bayesian networks based on existing ontologies.
- In [26] a new methodology has been presented for using causal knowledge to extend and improve a standard hierarchical medical ontology. The structure of the variables and the symptoms of patients is obtained based on the medical dictionary of the terminology of regulatory activities (Medical Dictionary for Regulatory Activities Terminology).

Among the most important contribution of probabilistic ontologies language is a PR-OWL. PR-OWL is an upper ontology written in the Web Ontology Language (OWL) that provides constructs for representing probabilistic ontologies based on Multi-Entity Bayesian Networks [31]. Improvements in OWL compatibility in the second release of PR-OWL enable ontology designers to express uncertainty associated with an existing OWL ontology [10].

PR-OWL 2 has also been adopted by other domains such as maritime [30] and fraud detection in Brazil [9].

3.4. Analysis

- The medical field, in general, is full of uncertain knowledge, especially the COVID-19 diagnosis, despite the success of ontologies, classical ontologies have been widely used to model and represent data and reasoning with the knowledge of COVID-19, like the approaches [6], [25] and [32], however, classical ontologies do not provide adequate support to deal with uncertain knowledge. This is because classical ontologies are based on Boolean logic which does not allow representing uncertain data. So, After studying existing works, we find that works using classical ontologies ([50], [37], [47], [25], [6], can't deal with uncertainty and they only have precise concepts and relationships.

⁴ <https://www.w3.org/2005/Incubator/urw3/group/draftReport.html>

- In the work [29], the reasoning process used by the adaptation COVID-19 is a rule-based technique. For example, a buccal temperature greater than 38.5 °C and the patient having a headache is a rule for detection of COVID-19, the problem is that this rule is deterministic, i.e. all patients with observed temperature and headache values, must have a positive diagnosis of COVID-19, which is incorrect. The limit of this reasoning process is not in all of the suspected cases with the observation cited above, we can deduce the diagnosis. This is caused by the uncertainty of medical knowledge related to COVID-19 diagnosis, so, we need a formalism for representing uncertainty like the probabilistic theory using the stochastic rule, for example, if a patient has a 38.5 °C and he has a headache, we can detect COVID-19 with the probability of 0.8.
- Several techniques have been proposed to confront the problem of reasoning under uncertainty using Bayesian Networks [18], [7], [18], [43], [34], [51]. BNs are a well-established technique for dealing with uncertainty, they exploit probabilistic reasoning to provide information about causal relationships for a set of variables modeling a given domain. They are credible for decision support.
- PR-OWL is the most expressive language for representing uncertain knowledge, and probabilistic ontology compared to other existing approaches. The meta-model of PR-OWL is based on multi-entity Bayesian networks (MEBNs) that combine the expressivity of first-order logic with Bayesian probability theory, for this, we opt for PR-OWL for modeling COVID-19 probabilistic ontology.
- From our study on ontologies dealing with medical diagnosis, it was found that there are no probabilistic ontologies proposed for medical diagnosis, and also there is no probabilistic ontology for COVID-19 diagnosis.

4. Proposed modeling

In this paper, we have constructed a probabilistic ontology, which combines the notion of classical ontology for representing the knowledge base, and Bayesian Multi-Entity Networks for treatment and reasoning under uncertainty. Probabilistic ontologies based on Bayesian Networks make it possible to describe uncertain knowledge in a reasoned and structured way and to model uncertainty by probability distributions and causal links.

In fact, a classic ontology can't express uncertain knowledge because it is based on deterministic logic. Moreover, a probabilistic ontology is equipped with a mechanism of probabilistic reasoning based on the Multi-Entity Bayesian Network (MEBN) inference engine.

A MEBN has several advantages over standard Bayesian networks:

1. MEBN offers a very high level of expressiveness based on first-order logic to better represent the real world and perceived reality.
2. MEBN represents a simple formalism for quick and effective modeling especially when it comes to a problem that contains a lot of repetitive knowledge structures.
3. MEBN offers highly flexible inference mechanisms based on the generation of Situation Specific Bayesian Networks (SSBNs)
4. Modularity, MFragments can be easily added or removed from the modeled system without any loss in the structural coherence of the network.

We first discuss the various forms of uncertainty addressed in this paper before introducing our ontology.

4.1. The uncertainty types

The uncertainty types tackled in this paper are incompleteness and randomness, defined in the high-level ontology proposed by the Uncertainty Reasoning for the World Wide Web Incubator Group of the World Wide Web Consortium. They are defined as follows:

1. **Randomness** - the award is an instance of a class for which a statistical law governs whether instances are satisfied.

We give an example of this uncertainty type in relation to our case study: For the diagnosis of COVID-19, it is difficult to predict it because there isn't an exact law that determines in a certain way the diagnosis from a set of symptoms, so we can't say for example, if the patient has a cough, fever, and headache, then, sure and certain that he is affected with the diagnosis, but there is uncertainty in the influence of these symptoms on the diagnosis, which induces us to use probability theory to model this randomness.

2. **Incompleteness** - information about the world is incomplete, it can take the form of missing information [23]. So, in our case study, with only a subset of symptoms and factors, the system can reason on the diagnosis, using probability distributions.

In the rest of this section, we propose a probabilistic ontology for COVID-19 diagnosis. Figure 1 shows our proposed process for predicting COVID-19 diagnosis, this process starts with constructing the probabilistic ontology, which is divided into three steps, constructing entities, rules, and the probabilistic components, then the reasoning of the diagnosis using the proposed ontology, based on probabilistic inference offered by Multi-Entity Bayesian Network. The details of these steps are presented in the following sections.

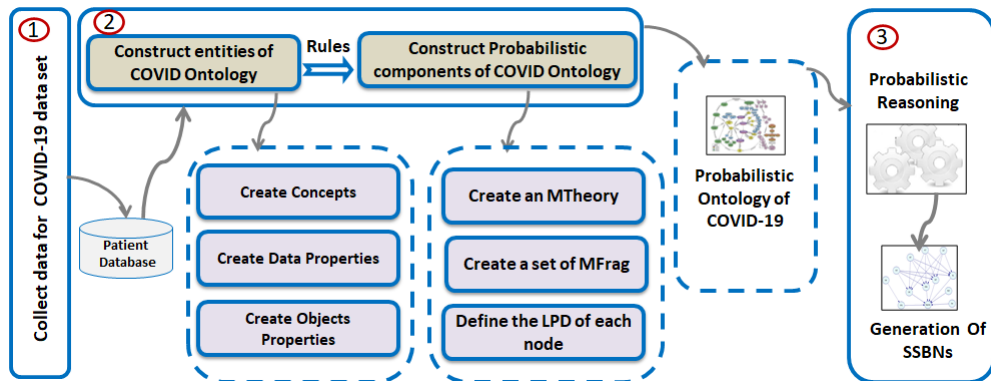


Fig. 1. Proposed modeling for aided diagnosis of COVID-19 based probabilistic ontology

The primary concern with the research on COVID-19 is the barrier prompted by the lack of adequate COVID-19 clinical data [11]. So, the first step to developing a probabilistic ontology is to collect anonymous data to use in the learning step.

The construction process of probabilistic ontology is described in the following major steps:

4.2. Collecting data for constructing COVID-19 dataset

It is worth emphasizing the significant problems facing any researcher with objectives like ours. There is a severe lack of transparency and important datasets are inaccessible. Diagnosing the COVID-19 pandemic is a multidisciplinary effort that requires significant data.

Data and their analytics are crucial components of such decision-making activities, one of the most difficult aspects is collected of accurate and detailed clinical data, even if these data have an imperfection.

We collected anonymous clinical data, and we analyzed it with expert domain, to facilitate better clinical decisions and treatment. Our dataset is made up of 300 real cases from patients in a hospital. This dataset contains anonymized medical data from patients that were tested for COVID-19. The important features of the data consist of symptoms and clinical signs of patients, risk factors, a set of radiological test results, and blood tests. The lack of such a dataset prompted us to build it ourselves.

The dataset includes patients that have been diagnosed positively with COVID-19 and those that have a negative diagnosis. Therefore both positive and negative diagnostic cases are present in the dataset. This latter is divided into two parts, one part is used for learning Bayesian Network parameters, which represents 80% of the dataset. The the remaining part is used for model testing.

Following discussions with the domain expert, we have identified the key **clinical symptoms** that are important for detecting COVID-19. These symptoms include Fever, Asthenia, Shortness of breath, Diarrhea, Dry cough, and Headache.

– **Biological symptoms** are:

- Blood urea,
- Creatinine,
- Blood Sugar,
- D.Dimers: is a fibrin degradation product, a small protein fragment present in the blood after a blood clot is degraded by fibrinolysis.
- WBC: The normal level of lymphocytes on the complete blood count
- HGB: Blood Hemoglobin (HB) is the amount of hemoglobin in 100 ml of blood
- SpO2: Blood saturation, it estimates a patient's condition.
- IGM/IGG: Are immunoglobulins produced by the immune system to provide protection against SARS-CoV-2. Anti-SARS-CoV-2 IgM and IgG can therefore be detected in samples from affected patients.
- CRP: The dosage of protein C.
- I.N.R/TP: I.N.R is an indicator of blood coagulation. TP is the prothrombin level is a biological test that evaluates the effectiveness of blood clotting in the body.

– **Radiological symptom** is a TDM, is an imaging test that scans an area of the body, such as the lungs and takes cross-sectional images of the area using a beam.

– **Risk factors** are Renal failure, Cardiovascular history, Diabetes, High blood pressure, Morbid obesity, and Chronic lung disease.

4.3. Constructing the probabilistic ontology of COVID-19

In this section, we present the development of the COVID-19 probabilistic ontology. While there is robust literature on ontology engineering and knowledge engineering for Bayesian Networks, the literature contains little guidance on how to model a probabilistic ontology. To fill the gap, Carvalho [10] proposed the Uncertainty modeling Process for Semantic Technologies, which describes the main tasks involved in creating probabilistic ontologies.

The First step in this model is to define the **entities, attributes, and relationships** by looking at the set of goals defined. After establishing the entities, their attributes, and relationships, we can proceed to specify the **rules** governing our Probabilistic Ontology (PO). Subsequently, we can create the **probabilistic components** by defining the groups and their elements. These components will aid in the implementation of the PO. In this step we define the Multi-Entity Bayesian Network, this network is composed of an MTheory and several MFrag, each MFrag contains the nodes and the corresponding probability distributions.

1. Entities and properties of COVID Ontology: After the data collection process, and after several discussions with the domain expert, we determined the domain entities with their properties, to model the entities of COVID-19 ontology, figure 2 shows the entities and attributes of COVID Ontology, it contains five classes.

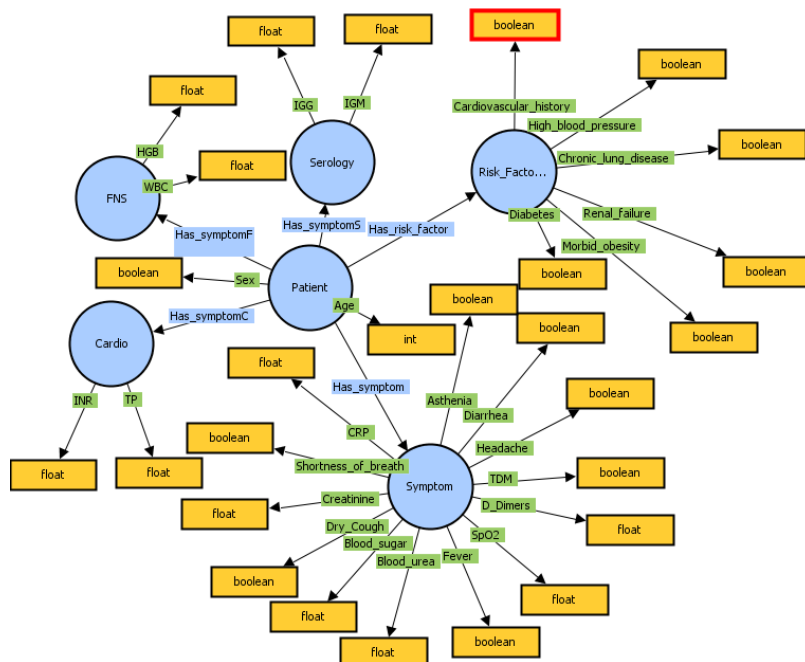


Fig. 2. Entities, attributes and relationships of COVID Ontology

The entities are presented in figure 2, which contains the following five classes.

- **Patient class** with the properties age and sex.
- **Symptom class**: with the properties Fever, Asthenia, Diarrhea, Blood urea, Headache, Creatinine, Blood Sugar, TDM, Shortness of breath, D_Dimers, SpO2, Dry cough, and CRP.
- **Risk factors class**: with the properties: Renal failure, Cardiovascular history, diabetes, High blood pressure, Morbid obesity, and Chronic lung disease.
- **Cardio class**: with the properties TP and INR.
- **Serology class**: with the properties IGG, and IGM.
- **FNS class**: with the properties HGB and WBC.

2. Rules: Generally, probabilistic rules are initially described using qualitative probability statements. The implementation of a probabilistic ontology requires specifying numerical probabilities and stochastic rules. Probability values can be obtained from domain experts or calculated from observation.

To determine stochastic rules, we propose to create a classical Bayesian Network structure, where all variables and dependencies must be linked, according to the causal relationship between the nodes, to extract the probability values. Then, we applied Expectation Maximisation (EM) for automatic learning parameters.

The stochastic rules are defined according to causal links between the nodes of the Bayesian network, taking into consideration the values of the probability distribution.

We present an example of stochastic rule between two nodes: TP which can take two values low and normal and Cardiovascular_history, which can take two values true and false, on a patient P.

Declaration of a rule on TP node:

```

if any P have ( Cardiovascular_History=false ) [
  low = 0.05 ,
  normal = 0.95
]else [
  if any P have ( Cardiovascular_History=true ) [
    low = 0.82 ,
    normal = 0.18
  ]else [
    low =0.5 ,
    normal = 0.5
  ]
]

```

3. Probabilistic components of COVID Ontology: The probabilistic COVID ontology aims to predict whether a patient is touched by COVID-19 or not, based on a set of evidence which represents symptoms and risk factors. Its construction pass through two steps.

In the **first step**, we design the different components of probabilistic COVID ontology PR-OWL2, which is based on a Multi-Entity Bayesian Network.

The MEBN language represents knowledge as a collection of MEBN Fragments (MFrag), which are organized into MEBN Theories (MTheories). An MFrag represents a repeatable pattern of knowledge that may apply to multiple domain entities. An MFrag consists of

Random Variables (RVs), a fragment graph whose nodes represent RVs, and Local Probability Distributions (LPDs) of some of these RVs. The repeated structure is represented by allowing RVs to have arguments that can be filled in with domain entities. These arguments are called ordinary variables to distinguish them from random variables. MFragS may have three kinds of nodes.

For constructing the MEBN, we define all components of MTheory and the set of MFragS of this MTheory. Creating the MFragS is done by defining the nodes, resident nodes, context nodes, and input nodes. Figure 3 shows the MFragS of COVID Ontology.

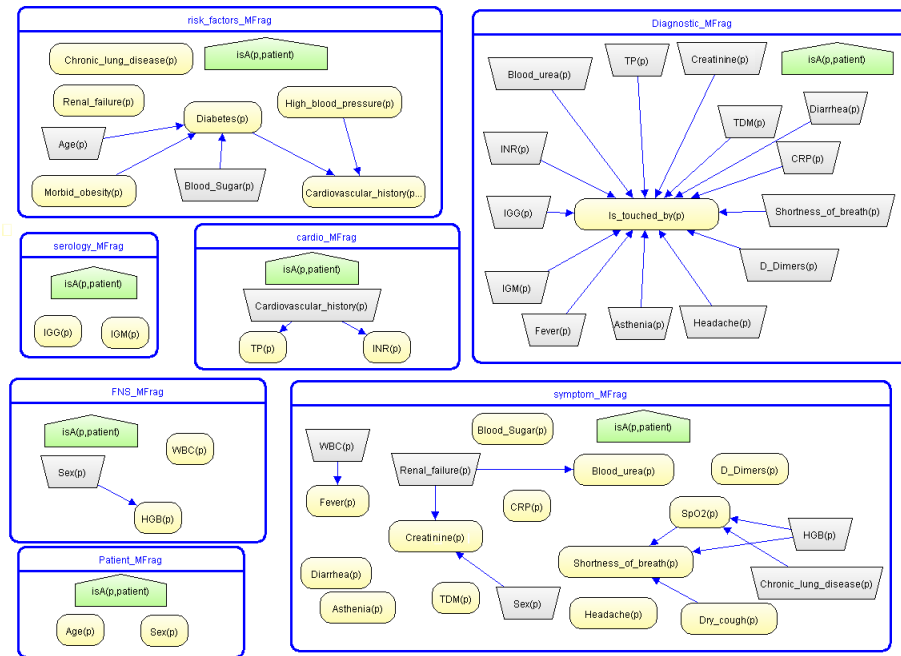


Fig. 3. MFragS of COVID Ontology

COVID probabilistic ontology presented in figure 3 contains 7 MFragS defined as follows:

- **Patient MFrag** contains the different attributes of the patient: age and sex, which are defined as resident nodes.
- **Risk-factors MFrag** contains the different risk factors for COVID-19 disease, it is defined by the resident nodes Chronic lung disease, Renal failure, High blood pressure and Morbid obesity which influence Diabetes node, this node is influenced also by input nodes Blood sugar and Age.
- **Serology MFrag** contains the different attributes of Serology: IGG and IGM, which are defined as resident nodes.
- **Cardio MFrag** contains the different attributes of Cardio: TP and INR, which are defined as resident nodes, and influenced by input node Cardiovascular history.

- **The FNS MFrag** contains the different attributes of FNS: WBC and HGB, which are defined as resident nodes. HGB is influenced by input node Sex.
- **Symptom MFrag** contains the different symptoms of Diarrhea, Asthenia, TDM, Blood sugar, CRP, Headache, and Shortness of breath which are influenced par Dy cough and SpO2. This later is influenced by input nodes HGB and Chronic lung disease. In this MFrag we define the Creatinine node influenced by input node renal failure and sex.
- **Diagnosis MFrag** with is represented by the input nodes IGG, IGM, INR, Blood urea, TP, Creatinine, TDM, Diarrhea, CRP, Shortness of breath, D_Dimers, Headache, Asthenia and Fever.

In the **second step**, we define the LPD of each node based on statistical study applied to collected data of COVID-19 anonymous patients. So, to complete the probabilistic ontology we add the stochastic rules between the random variables which be used in the reasoning process, but before this task, we compute the conditional probabilities between the variables, so that we can define a pseudo (called LPD) code for each variable.

All the components of the Multi-Entity Bayesian Network presented by the different MFraags with its nodes showing in Figure 3 are represented in the PR-OWL probabilistic ontology, we present a fragment that represents the PR-OWL code of the Asthenia node using the PR-OWL language.

```
<NamedIndividual
rdf:about="file:/unbbayes-4.22.18/COVID.owl#Asthenia_1">
  <rdf:type rdf:resource="pr-owl2:OrdinaryVariableArgument"/>
  <pr-owl2:hasArgumentNumber rdf:datatype="xsd:integer">
    1
  </pr-owl2:hasArgumentNumber>
  <pr-owl2:isArgumentOf
rdf:resource="file:/unbbayes-4.22.18/COVID.owl#MEXPRESSION_Asthenia"/>
  <pr-owl2:typeOfArgument
rdf:resource="file:/unbbayes-4.22.18/COVID.owl#symptom_MFrag.p"/>
</NamedIndividual>

<!-- file:/unbbayes-4.22.18/COVID.owl#Asthenia_Table -->
<NamedIndividual rdf:about=
"file:/unbbayes-4.22.18/COVID.owl#Asthenia_Table">
  <rdf:type rdf:resource="pr-owl2:DeclarativeDistribution"/>
  <pr-owl2:hasDeclaration rdf:datatype="xsd:string">
    [ false=0.89,
      true=0.11
    ]
  </pr-owl2:hasDeclaration>
</NamedIndividual>
```

4.4. Probabilistic inference using a probabilistic ontology of COVID-19

When a query is submitted, we use the proposed reasoning in MEBN, consisting construct a BN to answer the query, this process is called SSBN construction. Reasoning in MEBN consists of the generation of a Situation-Specific Bayesian Network (SSBN), a minimal Bayesian network sufficient to solve a set of target nodes for which it is necessary to calculate the probability.

The reasoning process for predicting COVID-19 diagnosis is realized by generating an SSBN for each patient, the set of nodes in SSBN represents a set of evidence of a patient, which represents the different values of symptoms and risk factors.

To construct the SSBN for each query of a patient, the process of reasoning is an iterative bottom-up process on the MFragments having a prior knowledge base of prior probability distribution across the resident random variables. This algorithm involves d-separation and inferring an intermediate Bayesian Network obtained from the set of Resident random variables for every iteration until the iterations are terminated.

The output of this process is an SSBN, which is a minimal Bayesian Network sufficient to obtain the posterior distribution for a set of instances of target random variables, given a set of occurrences of random variables. A standard Bayesian network inference algorithm is applied to SSBN.

The algorithm used for Bayesian inference is the junction tree algorithm; it propagates evidence through the whole structure of BN. Junction tree inference is used to perform efficient probabilistic inference in Bayesian networks. It involves transforming the network into a junction tree, where each node represents a cluster of related variables and joint probability distributions over those variables. The tree is constructed as a set of maximal cliques, and messages containing information about the joint probability distributions are passed between nodes to perform inference.

Finally, the answer to the query is obtained by inspecting the posterior probabilities of the target nodes.

In order to illustrate the mechanism of the inference in our system, let's suppose that we have a patient named "p2", with a set of evidences.

The SSBN generated for the query IsTouchedBy (p2) is shown in figure 4. For "p2" with a set of its evidences, we can see in the SSBN that there is a chance of 56,25% that p2 is not suffering from COVID-19.

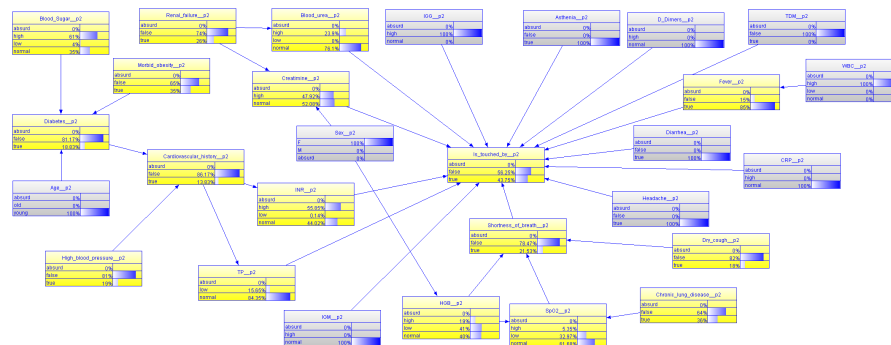


Fig. 4. The generated SSBN from evidences of p2

5. Experimentation

We have constructed our COVID-19 probabilistic ontology, to apply probabilistic inference for the diagnosis task, it is represented by PR-OWL language using the tool

UnbBayes¹. UnBBayes is a probabilistic network framework written in Java, it has a graphical interface.

In this section, we present the performance of our proposed modeling, tested on 20% of the dataset dedicated to model testing. We have used metrics are accuracy, precision, recall, and F1-score.

- **Accuracy** is the probability that the test will provide correct results, that is, be positive in sick patients and negative in healthy patients, it is the probability of the true positives and true negatives.
- **Precision** is the rate of the positive predictions that are actually positive.
- **Recall** it is the rate of true positives and indicates the classifier's ability to detect people with COVID-19 correctly.
- **F1-score** can be used as a single measure of performance of the test for the positive class. The F1-score is the harmonic mean of precision and recall.

This metric mainly is very applicable to cases of diagnosis in medicine since it allows for increased confidence and acceptability of results, an illustration of these parameters on our probabilistic ontology inference can be found in Table 1.

Table 1. Evaluation measures of our COVID-19 diagnosis model using ML metrics

Measures	Values
Precision	0.95
Recall	1
F1-score	0.97
Accuracy	0.97

The diagnosis accuracy metric was used to evaluate the ability of a diagnostic COVID to correctly identify by the probabilistic inference. In the medical world, the use of sensitivity, specificity, Positive predictive value (PPV), and Negative Predictive Value (NPV) metrics are more frequent, than other metrics.

- **Specificity** metric is the capacity of classifying healthy patients as negatives. It is the rate of true negatives.
- **Sensitivity** metric can be calculated in the same way as recall. it measures the ability to give a positive result when a hypothesis is verified, it refers to the ability to detect a maximum number of patients.
- **Positive Predictive Value (PPV)** It is the ratio of patients truly diagnosed as positive to all those who had positive test results (including healthy subjects who were incorrectly diagnosed as patients).
- **Negative Predictive Value (NPV)** It is the ratio of subjects truly diagnosed as negative to all those who had negative test results.

This metric mainly is very applicable to cases of diagnosis in medicine since it allows for increased confidence and acceptability of results, an illustration of these parameters on our probabilistic ontology inference can be found in Table 2.

¹ <https://sourceforge.net/projects/unbbayes>

Table 2. Evaluation measures of our COVID-19 diagnosis model using Diagnostic test evaluation

Measures	Values
Specificity	94.12%
Sensitivity	100%
Positive Predictive Value	95.92%
Negative Predictive Value	100.00%

Discussion: According to the presented results in Table 1 and Table 2:

- We note a high-specificity value in our system, which means it will correctly rule out almost everyone who doesn't have the disease and won't generate many false-positive results. The specificity value with 94% will correctly return a negative result for 94% of people who don't have the disease but will return a positive result (a false-positive for 6% of the people who don't have the disease and should have tested negative).
- Sensitivity measures how often a test correctly generates a positive result for people who have the condition that's being tested for (also known as the "true positive" rate). The Sensitivity value in our system is 100%, so, it will flag everyone who has the disease and not generate false-negative results. Therefore, it will correctly return a positive result for 100% of people who have the disease, but it won't return a negative result for the people who have the disease (the false Negative value is 0%).
- The PPV is the probability that the disease is present when the test is positive, in our system, we have a 95.92% value of PPV, which is a high value that indicates that a positive test result is likely correct, in 5% of healthy subjects who were incorrectly diagnosed as patients.
- NPV is the probability that the disease is not present when the test is negative, in our system we have a 100% value of NPV, so all subjects who were diagnosed as healthy are healthy subjects, and there are no patients who were incorrectly diagnosed as healthy.
- We can conclude that using the reasoning-based probabilistic ontology increases the prediction quality in terms of precision, recall, accuracy and F1-Score and handle the uncertainty related to the diagnosis of COVID-19. So, our system gives very good performance values, applied to the dataset instances used for the test.

A comparative analysis of the performance of our proposed approach was carried out with the machine learning methods: Logistic regression (LG), Support Vector Classifier(SVC), Decision tree (DT), Random Forest (RF), Gaussian Naive Bayes (GNB), and our model of Probabilistic Ontology (PO). The results of the performed evaluation are summarized in Table 3.

From Table 3, we can note that the experimental results for various machine learning models reveal that they provide less values of evaluation parameters regarding the probabilistic approach.

The best value of precision (1), is presented using SVC and FR methods, but they haven't the best value of recall which is 0.75, which resulted in a 0.83 of F1-Score, however, GNB and DT presented the best value of recall (1), but, it presented respectively 0.5

Table 3. Evaluation results using machine learning methods

Measures	LR	SVC	DT	RF	GNB	PO
Precision	1	1	0.75	1	0.5	0.95
Recall	0.75	0.75	1	0.75	1	1
F1-score	0.83	0.83	0.83	0.83	0.66	0.97
Accuracy	0.87	0.87	0.75	0.87	0.5	0.97

and 0.75 for precision value, which decreases its F1-Score measure to 0.5 for GNB and 0.75 for DT. LR presented intermediary values of precision and recall.

The proposed probabilistic model has exhibited promising predictive ability. It can be seen from Table 3 that the prediction for the probabilistic inference have the highest precision, recall, F1-Score, and accuracy and are, respectively, about **0.95**, **1**, **0.97**, and **0.97**, which signifies that the prediction basing on probabilistic inference is near to the prediction of the expert domain.

Histogram in Figure 5 outline the evaluation results using the standard evaluation measures defined in Table 3.

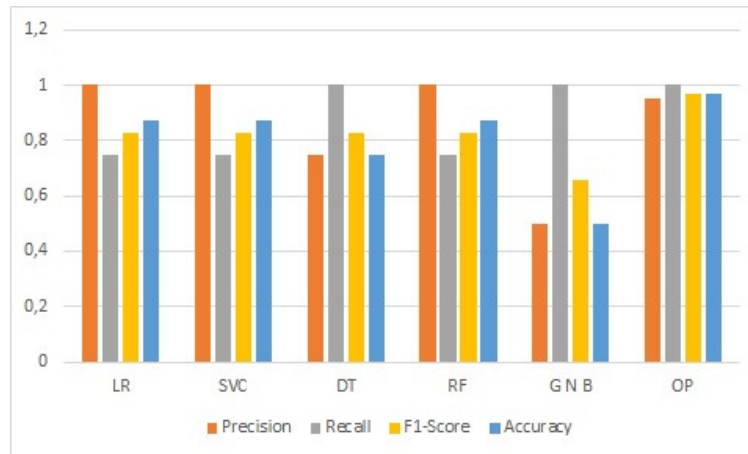


Fig. 5. Histogram of evaluation results

Discussion: The validity of a diagnostic test is evaluated in terms of its ability to detect subjects with COVID-19 as well as its capacity to exclude subjects without COVID-19. We can see from Figure 5 that:

- **Recall:** Our probabilistic ontology presents the best value of recall score (1), it is among the best learning machine methods which have the same value of recall, the two methods are: GNB and DT.
- **Precision:** Regarding precision value, OP achieved a very good value (0.95). It presents the second-best value after the methods of RH, SVC, and RD.

- **F1-Score:** For F1-Score, which combines precision and recall, the proposed modeling presents a high value (0.97), greater than the learning machine methods.
- **Accuracy:** For accuracy value, Our modeling based on probabilistic ontology presents the best value of accuracy (0.97) compared with machine learning methods.

The conducted evaluation of our modeling demonstrates that the probabilistic model based on Multi-Entity Bayesian Networks using ontology structure has improved the performance of the probabilistic ontology reasoning task, applied to COVID-19 diagnosis, so, probabilistic knowledge helps in improving tasks like disease prediction and risk propagation.

The current paper demonstrates the great potential of the probabilistic model as Multi-Entity Bayesian Network in tackling the uncertainty in knowledge for diagnosing COVID-19 by facilitating complex decision-making and fact interrogation.

6. Conclusion

In the medical field, doctors are regularly called upon to make several decisions; some of these decisions are taken easily when the diagnosis is easy, the treatment chosen is effective, and the risks are zero, while in other cases, such as COVID-19, the right decision to make is not obvious, because it is a new disease, it appeared recently in December 2019. The information necessary for its diagnosis is filled with uncertainty.

We aim to handle the uncertainty associated with the diagnosis of the new COVID-19 pandemic. Unlike the commonly known influenza, it came with limited information and very high uncertainty. Therefore, knowledge regarding the new epidemic needs to be treated due to the absence of a prior case similar to the recent pandemic. In this work, we are interested in tackling two types of uncertainty associated with the COVID-19 process: randomness and incompleteness.

Many studies applied ontology to represent knowledge, Ontologies are used for structuring and sharing knowledge because the common good practice of ontology engineering hinges on reusing and integrating existing ontologies. With the emergence of the immense quantity of data from various contexts, the need for shareable integration of domain knowledge increases. Ontologies are the key element for interoperability. However, classical ontologies do not have built-in mechanisms for representing or inferring with uncertain knowledge.

Network is a well-established technique for handling uncertainty within the artificial intelligence (AI) community. They exploit Bayes's probabilistic reasoning to provide insights into the causal relationships between the contributors and outcomes of an event.

The objective of this proposal is to predict the diagnosis of COVID-19 using the uncertainty management techniques in the field of AI, particularly the theory of probability, and the associated techniques, such as Bayesian networks and probabilistic ontologies.

The aim was to develop a probabilistic ontology for the aided diagnosis of COVID-19 infection in an individual, to take the necessary measures before reaching danger.

This model combines the advantages of classical ontologies for representing the knowledge base, the Multi-Entity Bayesian network for the uncertainty treatment associated with the prediction process, and the provided probabilistic inference mechanism.

We tested our modeling system on the collected dataset cases, and it gave good results in terms of precision and recall. The advantage of a Bayesian network is that it can predict

the diagnosis even in the case of missing values, without requiring some biological tests. With only a few symptom values, the system can respond to the given observations.

To treat the various symptoms and risk factors associated with COVID-19 variants, we envisage following the development of these variants, and testing our method on big data.

References

1. Al-Qaness, M.A., Ewees, A.A., Fan, H., Abd El Aziz, M.: Optimization method for forecasting confirmed cases of covid-19 in china. *Journal of Clinical Medicine* 9(3), 674 (2020)
2. Albahri, A.S., Hamid, R.A., Alwan, J.K., Al-Qays, Z., Zaidan, A., Zaidan, B., Albahri, A., AlAmoodi, A.H., Khlaf, J.M., Almahdi, E., et al.: Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (covid-19): a systematic review. *Journal of medical systems* 44, 1–11 (2020)
3. Ansari, A.Q., Biswas, R., Aggarwal, S.: Proposal for applicability of neutrosophic set theory in medical ai. *International Journal of Computer Applications* 27(5), 5–11 (2011)
4. Becker, A., Naim, P.: *Les réseaux bayésiens: modèles graphiques de connaissance*. Eyrolles (1999)
5. Bhuvana, J., Mirmalinee, T., Bharathi, B., Sneha, I.: Efficient generative transfer learning framework for the detection of covid-19. *Computer Science and Information Systems* (00), 33–33 (2022)
6. Bolock, A.E., Abdennadher, S., Herbert, C.: An ontology-based framework for psychological monitoring in education during the covid-19 pandemic. *Frontiers in Psychology* 12, 673586 (2021)
7. Butcher, R., Fenton, N.: Extending the range of symptoms in a bayesian network for the predictive diagnosis of covid-19. *medRxiv* pp. 2020–10 (2020)
8. Carvalho, R.N., Laskey, K.B., Costa, P.C.: Pr-owl—a language for defining probabilistic ontologies. *International Journal of Approximate Reasoning* 91, 56–79 (2017)
9. Carvalho, R.N., Matsumoto, S., Laskey, K.B., da Costa, P.C.G., Ladeira, M., Santos, L.L.: Probabilistic ontology and knowledge fusion for procurement fraud detection in brazil. In: *URSW (LNCS Vol.)*. pp. 19–40. Springer (2013)
10. Carvalho, R.N.: *Probabilistic ontology: representation and modeling methodology*. George Mason University (2011)
11. Chiroma, H., Ezugwu, A.E., Jauro, F., Al-Garadi, M.A., Abdullahi, I.N., Shuib, L.: Early survey with bibliometric analysis on machine learning approaches in controlling covid-19 outbreaks. *PeerJ Computer Science* 6, e313 (2020)
12. Conrady, S., Jouffe, L.: *Bayesian networks and BayesiaLab: a practical introduction for researchers*, vol. 9. Bayesia USA Franklin (2015)
13. Costa, P.C., Laskey, K.B., Blasch, E., Joussemme, A.L.: Towards unbiased evaluation of uncertainty reasoning: The urref ontology. In: *2012 15th International Conference on Information Fusion*. pp. 2301–2308. IEEE (2012)
14. Da Costa, P.C.G., Laskey, K.B., Laskey, K.J.: Pr-owl: A bayesian ontology language for the semantic web. In: *Uncertainty reasoning for the semantic web I: ISWC international workshops, URSW 2005-2007, revised selected and invited papers*. pp. 88–107. Springer (2008)
15. Ding, Z., Peng, Y.: A probabilistic extension to ontology language owl. In: *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*. pp. 10–pp. IEEE (2004)
16. Eom, J.H., Kim, S.C., Zhang, B.T.: Aptacdss-e: A classifier ensemble-based clinical decision support system for cardiovascular disease level prediction. *Expert Systems with Applications* 34(4), 2465–2479 (2008)

17. Fareh, M.: Modeling incomplete knowledge of semantic web using bayesian networks. *Applied Artificial Intelligence* 33(11), 1022–1034 (2019)
18. Fenton, N.E., McLachlan, S., Lucas, P., Dube, K., Hitman, G.A., Osman, M., Kyrimi, E., Neil, M.: A privacy-preserving bayesian network model for personalised covid19 risk assessment and contact tracing. *MedRxiv* pp. 2020–07 (2020)
19. Fenz, S.: An ontology-based approach for constructing bayesian networks. *Data & Knowledge Engineering* 73, 73–88 (2012)
20. Fong, S.J., Li, G., Dey, N., Crespo, R.G., Herrera-Viedma, E.: Composite monte carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction. *Applied soft computing* 93, 106282 (2020)
21. Fung, R., Chang, K.: Weighting and integrating evidence for stochastic simulation in bayesian networks, uncertainty in artificial intelligence 5 (1989)
22. Fung, R., Del Favero, B.: Backward simulation in bayesian networks. In: *Uncertainty Proceedings 1994*, pp. 227–234. Elsevier (1994)
23. Hamel, O., Fareh, M.: Missing types prediction in linked data using deep neural network with attention mechanism: Case study on dbpedia and uniprot datasets. In: *Information Technology for Management: Approaches to Improving Business and Society: AIST 2022 Track and 17th Conference, ISM 2022, Held as Part of FedCSIS 2022, Sofia, Bulgaria, September 4–7, 2022, Extended and Revised Selected Papers*. pp. 212–231. Springer (2023)
24. Han, P.K., Klein, W.M., Arora, N.K.: Varieties of uncertainty in health care: a conceptual taxonomy. *Medical Decision Making* 31(6), 828–838 (2011)
25. He, Y., Yu, H., Ong, E., Wang, Y., Liu, Y., Huffman, A., Huang, H.h., Beverley, J., Hur, J., Yang, X., et al.: Cido, a community-based ontology for coronavirus disease knowledge and data integration, sharing, and analysis. *Scientific data* 7(1), 181 (2020)
26. Hu, H., Kerschberg, L.: Evolving medical ontologies based on causal inference. In: *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. pp. 954–957. IEEE (2018)
27. Jangam, E., Barreto, A.A.D., Annavarapu, C.S.R.: Automatic detection of covid-19 from chest ct scan and chest x-rays images using deep learning, transfer learning and stacking. *Applied Intelligence* pp. 1–17 (2022)
28. Jensen, F.: Bayesian updating in recursive graphical models by local computations. *Computational Statistics and Data Analysis* 4, 269–282 (1990)
29. Kouamé, K.M., Mcheick, H.: An ontological approach for early detection of suspected covid-19 among copd patients. *Applied System Innovation* 4(1), 21 (2021)
30. Laskey, K.B., Haberman, R., Carvalho, R.N., da Costa, P.C.G.: Pr-owl 2 case study: A maritime domain probabilistic ontology. In: *STIDS*. pp. 76–83 (2011)
31. Laskey, K.B., Costa, P.C., Janssen, T.: Probabilistic ontologies for knowledge fusion. In: *2008 11th international conference on information fusion*. pp. 1–8. IEEE (2008)
32. Liu, Y., Chan, W., Wang, Z., Hur, J., Xie, J., Yu, H., He, Y.: Ontological and bioinformatic analysis of anti-coronavirus drugs and their implication for drug repurposing against covid-19 (2020)
33. Mondal, M.R.H., Bharati, S., Podder, P., Podder, P.: Data analytics for novel coronavirus disease. *informatics in medicine unlocked* 20, 100374 (2020)
34. Neil, M., Fenton, N., Osman, M., McLachlan, S.: Bayesian network analysis of covid-19 data reveals higher infection prevalence rates and lower fatality rates than widely reported. *Journal of Risk Research* 23(7-8), 866–879 (2020)
35. Nour, M., Cömert, Z., Polat, K.: A novel medical diagnosis model for covid-19 infection detection based on deep features and bayesian optimization. *Applied Soft Computing* 97, 106580 (2020)
36. Organization, W.H., et al.: *Coronavirus disease 2019 (covid-19) situation*. Geneva: Report-121 (2020)

37. Oyelade, O.N., Ezugwu, A.E.: A case-based reasoning framework for early detection and diagnosis of novel coronavirus. *Informatics in Medicine Unlocked* 20, 100395 (2020)
38. Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O., Acharya, U.R.: Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in biology and medicine* 121, 103792 (2020)
39. Paulo Cesar, G.: *Da Costa, Bayesian semantics for the semantic web*. Ph.D. thesis, Ph. D. Thesis, Fairfax, VA, USA (2005)
40. Pearl, J.: *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann (1988)
41. Peñaloza, R.: Introduction to probabilistic ontologies. *Reasoning Web. Declarative Artificial Intelligence: 16th International Summer School 2020, Oslo, Norway, June 24–26, 2020, Tutorial Lectures* 16 pp. 1–35 (2020)
42. Pirouz, B., Shaffiee Haghshenas, S., Shaffiee Haghshenas, S., Piro, P.: Investigating a serious challenge in the sustainable development process: analysis of confirmed cases of covid-19 (new type of coronavirus) through a binary classification using artificial intelligence and regression analysis. *Sustainability* 12(6), 2427 (2020)
43. Proadhan, G., Fenton, N.: Extending the range of covid-19 risk factors in a bayesian network model for personalised risk assessment. *medRxiv* pp. 2020–10 (2020)
44. Riali, I., Fareh, M., Bouarfa, H.: Fuzzy probabilistic ontology approach: A hybrid model for handling uncertain knowledge in ontologies. *International Journal on Semantic Web and Information Systems (IJSWIS)* 15(4), 1–20 (2019)
45. Riali, I., Fareh, M., Ibnaissa, M.C., Bellil, M.: A semantic-based approach for hepatitis c virus prediction and diagnosis using a fuzzy ontology and a fuzzy bayesian network. *Journal of Intelligent & Fuzzy Systems* 44(2), 2381–2395 (2023)
46. Rodriguez-Morales, A.J., Cardona-Ospina, J.A., Gutiérrez-Ocampo, E., Villamizar-Peña, R., Holguin-Rivera, Y., Escalera-Antezana, J.P., Alvarado-Arnez, L.E., Bonilla-Aldana, D.K., Franco-Paredes, C., Henao-Martinez, A.F., et al.: Clinical, laboratory and imaging features of covid-19: A systematic review and meta-analysis. *Travel medicine and infectious disease* 34, 101623 (2020)
47. Sargsyan, A., Kodamullil, A.T., Baksi, S., Darms, J., Madan, S., Gebel, S., Keminer, O., Jose, G.M., Balabin, H., DeLong, L.N., et al.: The covid-19 ontology. *Bioinformatics* 36(24), 5703–5705 (2020)
48. Shachter, R.D., Peot, M.A.: Simulation approaches to general probabilistic inference on belief networks. In: *Machine intelligence and pattern recognition*, vol. 10, pp. 221–231. Elsevier (1990)
49. Singhal, T.: A review of coronavirus disease-2019 (covid-19). *The indian journal of pediatrics* 87(4), 281–286 (2020)
50. Stancin, K., Poscic, P., Jaksic, D.: Ontologies in education—state of the art. *Education and Information Technologies* 25(6), 5301–5320 (2020)
51. de Terwangne, C., Laouni, J., Jouffe, L., Lechien, J.R., Bouillon, V., Place, S., Capulzini, L., Machayekhi, S., Ceccarelli, A., Saussez, S., et al.: Predictive accuracy of covid-19 world health organization (who) severity classification and comparison with a bayesian-method-based severity score (epi-score). *Pathogens* 9(11), 880 (2020)
52. Vrbačić, G., Pečnik, Š., Podgorelec, V.: Hyper-parameter optimization of convolutional neural networks for classifying covid-19 x-ray images. *Computer Science and Information Systems* 19(1), 327–352 (2022)
53. Yang, Y., Calmet, J.: Ontobayes: An ontology-driven uncertainty model. In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. vol. 1, pp. 457–463. IEEE (2005)

Messaouda Fareh is a doctor in computer sciences, and a lecturer at the computer sciences Department, University of Blida 1, Blida, Algeria. She is a member of the Laboratory LRDSI. Her research interests include ontology engineering and knowledge, information heterogeneity, uncertain knowledge of semantic and linked data, and data mining.

Ishak Riali is a Ph.D. in data science, in the computer science department at University of Blida1, Blida, Algeria. His primary research interests are artificial Intelligence, ontology engineering, uncertain Knowledge, Knowledge-Based Systems, semantic interoperability, and machine learning for Linked Data.

Hafsa Kherbache received her BSc (2019) and MSc (2021) degrees in Software Engineering from the University of Blida1. Her interests include knowledge and ontological engineering, semantic web, and probabilistic knowledge.

Marwa Guemmouz obtained her BSc degree in 2019 and her MSc in Software Engineering in 2021 from the University of Blida1. Her interests include knowledge and ontological engineering, semantic web, and probabilistic knowledge.

Received: August 29, 2022; Accepted: April 04, 2023.

The Proposal of New Ethereum Request for Comments for Supporting Fractional Ownership of Non-Fungible Tokens *

Miroslav Stefanović¹, Đorđe Pržulj¹, Darko Stefanović¹, Sonja Ristić¹,
and Darko Čapko^{1,2}

¹ University of Novi Sad, Faculty of Technical Sciences
Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
{mstef, przulj, darko.stefanovic, sdristic}@uns.ac.rs

² Eternal, Nikolajevska 2, 21000 Novi Sad, Serbia
darko@eternal.tech

Abstract. During the last couple of years, non-fungible tokens became the most prominent implementation of blockchain technology apart from cryptocurrencies. This is mainly due to their recent association with digital art, but the application of non-fungible tokens has been in the focus of researchers since the appearance of Blockchain 2.0. It was usually tightly coupled with the research on possible applications of blockchain technology in some real-life applications, such as land administration, healthcare, or supply chain management. Since the initial release of the Ethereum blockchain in 2015, until 2022, more than 44 million smart contracts have been created, and out of those that are still active, more than 70% are based on some prominent templates. In the Ethereum blockchain, the creation of non-fungible tokens is usually based on Ethereum Request for Comments 721. In this paper, the authors are proposing the creation of a new standard that would support fractional ownership of non-fungible tokens. Fractional ownership is necessary so non-fungible tokens and blockchain technology could be applied to an even wider number of use cases. This paper also presents an example of a possible implementation of the newly proposed standard in the Solidity programming language.

Keywords: blockchain, smart contract, non-fungible tokens, NFT, Ethereum, ERC

1. Introduction

In 2009, a white paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" was published and even though it was nowhere directly mentioned by that name, blockchain technology (BT) was born. BT represents the first implementation of distributed ledger technology (DLT). DLT is a solution that, instead of a centralized registry, has a unique registry that is distributed among multiple nodes with decentralized control. These nodes record, share and synchronize data across the network, making the data secure by reaching a consensus on the content of the registry [39]. The first concrete implementation of BT is the Bitcoin blockchain, but over the years various DLT platforms have been implemented [12].

* Based on extended abstract titled "Ethereum Request for Comments for Fractional Ownership of Non-Fungible Tokens" that was presented at XVIII International Symposium - SymOrg2022, Belgrade, Serbia

In BT, transactions are stored in the chain of blocks. Blocks are added in chronological order making the possibility of manipulation or forgery highly unlikely [25]. The system is secured by making data falsification almost impossible because the data is distributed among a large number of interconnected nodes, the so-called blockchain network. Distributed data and the fact that there is no single point of failure make the system resilient, and the fact that the use of the system is public makes it transparent [30]. It is very important to eliminate or at least minimize the possibility of manipulation or forgery within the transactions stored in a blockchain. To achieve this, BT relies on a cryptographic hash function, asymmetric cryptography, and distributed consensus mechanism [55]. The role of the consensus mechanism in the blockchain network is to make it possible for nodes to reach an agreement on the single state of the network [2]. The new block is added to the chain only once nodes perform the same computation, achieving the same result, and reaching a consensus on that result [21]. Some of the advantages of BT are efficiency, security, resilience, and transparency. The fact that it is possible to easily monitor and manage complex data logs with the help of BT makes this solution very effective.

One of the most common classifications divides blockchains into three categories: public permissionless blockchains, consortium/hybrid/public permissioned blockchains, and private blockchains [46]. Public blockchains are blockchains where all transactions are public and anyone can join the network as a node and participate in the process of confirmation of transactions [55][35]. In consortium/hybrid/public permissioned blockchains, transactions are also public, but only a set of predetermined nodes participate in confirmations of transactions. Private blockchains are blockchains where only selected nodes can see and participate in process of confirmations of transactions [28] [45].

The main characteristics of BT are decentralization, persistence, anonymity, and auditability [13][21][35][49][55].

In BT, decentralization is achieved by the possibility for every node to manage and store transactions. Information about transactions is exchanged between all the nodes on the network, thus eliminating the need for a trusted third party [49].

Blocks are added onto a chain by having the content of the previous block, through its hash value, participating in the content of the next block. In that way, each block in the chain participates in the content of its successor block. This implies that in case the content of a previous block, which is already a part of a blockchain, is changed, it would invalidate all the following blocks [13]. In this way, persistency of the transactions recorded within the blockchain is achieved. The new transactions can be added to the blockchain while the possibility of deleting or updating previously registered transactions is highly unlikely [35].

Anonymity is a characteristic of public permissionless and partly of consortium/hybrid/public permissioned blockchains. In these blockchains interested parties could participate as clients by exchanging assets, or even as a node in a public permissionless blockchain, without the need to expose their identity, thus preserving their anonymity.

In [31], Bitcoin is described as a peer-to-peer distributed timestamp server, meaning that all transactions that happen on the blockchain are timestamped. As each transaction is timestamped and since forgery is highly unlikely, as previously mentioned, an interested party can search the blockchain for any previous transactions, thus making the blockchain auditable [13][55].

All mentioned advantages and characteristics related to BT are applicable to smart contracts, too. The term smart contract was mentioned for the first time in 1996 and it was defined as a "set of promises, specified in digital form, including protocols within which the parties perform on these promises" [43]. The idea was based on the possibility for contract clauses to be implemented either through hardware or software in a way that any breach of contract would cause significant expense for a breaching party [43]. In BT, a smart contract is a computer program that is deployed on the blockchain network and that is governed by the same rules that govern transactions [42]. They could be used to automatically verify and execute contract clauses once predetermined conditions have been met [17].

The most common way of representing real-life assets in smart contracts is through tokens, and in the Ethereum blockchain, tokens are usually created in accordance with ERC-20 and ERC-721 standards. ERC-20 token standard represents an interface that will allow the creation of fungible tokens that can be used by other applications, such as wallets and exchanges. Fungible tokens are used for representing interchangeable assets, for example for the creation of new cryptocurrencies. ERC-721 non-fungible token standard represents an interface that will allow the creation of non-fungible tokens (NFT) that can be used in the same manner as ERC-20 tokens but are used to represent unique assets, that can not be interchanged, such as artwork or real-estate. Although these two standards can cover a wide range of use cases, there is a significant number of use cases that can not be fully supported by either of these tokens and those are mainly related to fractional ownership of non-fungible tokens. For example, in previously mentioned applications of land administration and supply chain management, it would surely be necessary for smart contracts to support fractional ownership and use cases in which:

- multiple entities could share ownership of an item,
- entities might have different shares in the ownership, and
- entities having a share of ownership could transfer less than their share of ownership to another entity.

The lack of an adequate standard for supporting fractional ownership of NFTs leads to the creation of smart contracts with different sets of application programming interfaces (APIs). This means that any application that needs to communicate with such a smart contract would need to be tailored to that specific set of APIs. This could especially be problematic if an application needs to communicate with a large number of non-standard smart contracts because it would need to be tailored to each of those smart contracts' specific set of APIs. This represents a major issue and this paper proposes a solution for it.

The solution for this issue is proposed in this paper in a form of a new ERC that will be built upon existing ERC-721 standards and that will propose a solution for the problem of fractional asset ownership and its sharing. Furthermore, the proposed solution defines a standard set of APIs that would make smart contracts implementing this new standard easily interoperable with other applications. This ERC will be proposed in a form of a Unified Modeling Language (UML) class diagram and as a programming interface written in Solidity programming language. The main contribution of the proposed solution is the definition of a novel ERC that could be introduced as a new standard to the existing body of ERC.

Apart from the Introduction and Conclusion, this paper is organized as follows, in Section 2 a short literature review is presented. In Section 3 ERC-20 and ERC-721 standards are presented together with the proposal of the new standard in section 4 and an example of implementation of the proposed standard in section 5.

2. Literature review

Bitcoin blockchain, now often referred to as Blockchain 1.0 [10][53], was introduced with the intention of creating of peer-to-peer electronic cash system that would not suffer from the problem of double spending and will not need a trusted third party to execute transactions [31]. Bitcoin blockchain had some possibilities of application in fields other than cryptocurrency, but true advances came with Blockchain 2.0. Blockchain 2.0 is a term used to represent blockchain that supports smart contracts [34][48]. Smart contracts are programs executed on the blockchain network and their correctness is enforced by the consensus mechanism [4][26]. Smart contracts first appeared on blockchain in 2015 with the Ethereum blockchain network. Ethereum blockchain network was first mentioned in 2014, in a white paper by Vitalik Buterin, where it was announced as a platform for developing Decentralized Applications (DApps) based on smart contracts [8][27].

Since then, the application of BT in fields other than cryptocurrency has come into the focus of scientific research. A literature review conducted in 2017, examining Web of Science, IEEE Xplore, the AIS Electronic Library, ScienceDirect, and SSRN for scholarly journal articles and conference proceedings, looking for conceptual papers or empirical analyses on possible application, use, or implications that BT could have on humans, organizations, and markets, has discovered only 69 papers on these subjects [36]. In 2019, another literature review paper presented the results of research conducted at the beginning of 2018, has shown a significant increase in the number of published papers on this subject. In this research, the main source of scientific papers was Scopus, and 245 papers were identified in fields other than cryptocurrency and finance. Out of those 245 papers, most were in the field of business and industry with 56 papers in total [11]. Research conducted in 2021, which included only journal articles on the subject of security, application, and challenges in BT, that were indexed in Scopus, IEEE Xplore, Google scholar, ScienceDirect, SpringerLink, and Web of Science, showed that this trend continues with 335 analyzed papers [24].

Some of the more prominent examples of possible applications of BT in fields other than cryptocurrency are healthcare [22][29][50], land administration [7][40][42], government [19][23][32], IoT [1][3][52] and supply chain management [5][18][37][38].

According to [51], since the genesis block, the first block of the Ethereum blockchain, over 44 million smart contracts have been deployed on this network. Half of that number has been destroyed, but from the remaining 22 million, around 70% are created based on only 15 templates. These data emphasize the importance of these templates, and in the case of the Ethereum blockchain, templates are usually built in accordance with Ethereum Request for Comments (ERC). In the Ethereum blockchain ERCs represent one of the Ethereum Improvement Proposal (EIP) types that are intended for defining application-level standards and conventions, such as token standards, URI schemes, library/package formats, and name registries.

The introduction of smart contracts into BT has opened the possibility for the development of DApps and Decentralized Autonomous Organizations (DAOs). As the blockchain network represents a distributed system, with no central authority, where decisions are made based on a decentralized consensus mechanism, in the same way, applications that are executed on blockchain networks are also decentralized and represent a special kind of software whose execution is not controlled by a single entity [47]. DAO represents a long-term smart contract for managing certain digital properties that holds all the business rules for one organization and functions without any human intervention [8][9].

Compared to traditional contracts, the following advantages of smart contracts have been recognized.

- Reducing risks – due to the manner in which persistence is achieved in BT, once smart contracts are deployed on a blockchain network, their implementation can not be changed, furthermore, since all transactions are public and since they are being saved on all full nodes they can be audited, thus reducing the risk of malicious behavior.
- Reducing administration and service costs – unlike centralized systems, where there are costs associated with operating trusted third parties, in blockchain networks it is a consensus mechanism that is tasked with confirming transactions, thus reducing the associated costs.
- Improving the efficiency of business processes – the possibility to execute contract clauses automatically, as soon as preconditions are met, can have significant time reduction compared to that required for the process to be executed by a trusted third party [54].

According to [45][54] life cycle of a smart contract consist of the following stages:

- Creation – involved parties, in some cases with help of a solicitor or other legal counsel, draft the initial contract. Software engineers convert this contract into a smart contract. The process of development of smart contracts passes the usual stages of software development, such as design, implementation, and validation.
- Deployment – in this stage, a smart contract is being deployed on the blockchain network. As previously mentioned, once deployed, a smart contract can not be changed and if any change is necessary, then a new smart contract must be deployed.
- Execution – once a smart contract has been deployed, contracted clauses are being monitored. When conditions are met, required functions are executed.
- Completion – once a smart contract has been executed, the state related to all parties in the contract has been updated and the new state has been saved onto the blockchain network.

The first application of smart contracts that achieved public prominence during the last couple of years was NFTs [14]. This prominence came as a result of hype related to digital collectibles. NFTs represent a tokenized item of value where each token owns a unique set of characteristics [33]. In some cases, those tokens can be a part of the same "universe" and still have different values, such as is the case with virtual collectibles Bored Ape Yacht Club, CryptoPunks, and Mutant Ape Yacht Club, or could represent unique digital artwork such as The Merge, The First 5000 Days, and Clock that were sold for almost 92m, 70m, and 53m dollars respectively [20]. Apart from these more prominent examples, significant efforts have been put into research related to the application of NFTs in other

fields, and those fields mainly coincide with the previously mentioned field of interest for the general application of BT. NFTs are often classified into six different categories: collectibles, art, metaverse, utility, and others [6]. In some of those categories, having a standard interface that would enable fractional ownership of NFTs would surely be beneficial, while in others, especially those related to real-life goods, it's not just that it would be beneficial, but in fact, it would be necessary.

This paper is based on extended abstract that was presented at SymOrg 2022 - XVIII International Symposium [41]. Presented extended abstract gave a short introduction to the need for a standard that will support fractional ownership of NFTs and provided a draft version of the necessary function that was used as a foundation of this research. The draft UML class diagram presented in this extended abstract was extended, elaborated, and refined for this paper. Based on this new UML class diagram, in this paper, a programming interface written in Solidity programming language is presented. Additionally, the proposed programming interface is accompanied by constraints that an implementation of this interface must satisfy. An example of such implementation of a smart contract, that satisfies all the required constraints, also represents an addition to the previously published extended abstract.

3. Existing ERC standards

In the Ethereum blockchain network, changes related to core protocol, smart contracts, and client APIs are made based on the Ethereum Improvement Proposal (EIP). EIP represents a standard for specifying potential new capabilities and processes on the Ethereum network. EIPs are divided into several categories, and those categories are:

- Standard track – used for changes that affect almost all segments of the Ethereum network, such as network protocol changes. At the end of 2022, there were 531 EIPs in this category.
- Core – improvements that consensus fork in the consensus mechanism. At the end of 2022, there were 197 EIPs in this category.
- Networking – improvements related to the implementation of devp2p Wire Protocol, RLPx Discovery Protocol and RLPx TCP Transport Protocol. At the end of 2022, there were 14 EIPs in this category.
- Interface – improvements related to API/remote procedure calls (RPC), standards related to method naming, and application binary interface (ABI) of smart contracts. At the end of 2022, there were 46 EIPs in this category.
- ERC – improvements related to standards and conventions at the application level, such as standards for tokens, name registries, uniform resource identifier (URI) schema, and library and package formats. At the end of 2022, there were 274 EIPs in this category, and out of those 274, 46 were in status final, 9 were in the last call, 24 were in review, 73 were in status draft, 117 were stagnant, and 5 were withdrawn.
- Meta – improvements related to the processes surrounding the Ethereum network, but unlike the Standard track, they do not refer to the Ethereum protocol itself. At the end of 2022, there were 20 EIPs in this category.
- Informational – do not represent improvement suggestions, but provide instructions, guidelines, or information to the Ethereum community. At the end of 2022, there were 6 EIPs of this type [16].

Among the 46 ERC standards that are in final status, 7 represent standards related to tokens:

- ERC-20 Token Standard – defines a standard interface that enables the creation of new tokens, which will be used by other applications,
- ERC-721 Non-Fungible Token Standard – defines a standard interface for creating unique (non-fungible) tokens,
- ERC-777 Token Standard – defines the improvement of the ERC-20 Token Standard,
- ERC-1155 Multi Token Standard – defines a standard interface for smart contracts managed by several different tokens,
- ERC-1363 Payable Token – defines the improvement of the ERC-20 Token Standard,
- ERC-3525 Semi-Fungible Token – defines a standard interface for creating tokens that will have part of the features described in ERC-20, and part of the features described in ERC-721, and
- ERC-4626 Tokenized Vaults – defines an enhancement of the ERC-20 Token Standard to provide support for the implementation of tokenized Vaults.

Out of those 7 standards, there are only two basic standards for creating tokens on the Ethereum network, namely ERC-20 and ERC-721, while the remaining five represent improvements of these standards. ERC-20 and ERC-721 are defined in the form of programming interfaces written in the Solidity programming language. In both cases, a set of APIs is defined that should allow tokens created in accordance with these standards to be used by various applications, cryptocurrency wallets, and decentralized exchanges. By implementing either of these two standards, functionality will be implemented that will enable the transfer of tokens by the owner or another authorized entity [15][44]. Both standards will be presented in the form of UML class diagrams and their function calls will be explained. UML class diagram representing the ERC-20 standard is shown in Fig. 1.

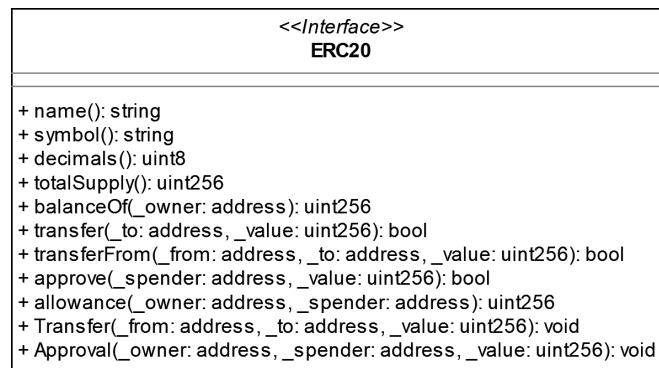


Fig. 1. UML class diagram of ERC-20 standard

Functions *name()*, *symbol()*, *decimals()*, and *totalSupply()* return values representing a name, symbol, decimal value, and total supply of a created token. These values are optional, and while they may improve application usability, other interfaces and smart contracts cannot expect that token name value will exist in every ERC-20 implementation.

Function *balanceOf()* returns a number representing the amount of tokens owned by that address passed as the argument in the function call. Address type in Solidity programming language represents a 20-byte value of Ethereum address. Depending on the function the address type is used to either represent the current or future owner of a token.

Functions *transfer()* and *transferFrom()* transfer the amount of tokens specified in the function call from either function caller or from the address passed as an argument.

Functions *approve()* and *allowance()* make it possible for an address to be approved to transfer a certain amount of tokens on the behalf of the owner.

Transfer() and *Approval()* events are emitted when corresponding functions are successfully executed [44]. Events are abstractions of the Ethereum logging protocol. In the case of their call, the passed arguments are stored in the transaction log, which is a special data structure on the blockchain. These logs are linked to the smart contract address and are permanently stored on the blockchain.

UML diagram representing the ERC-721 standard is shown in Fig. 2.

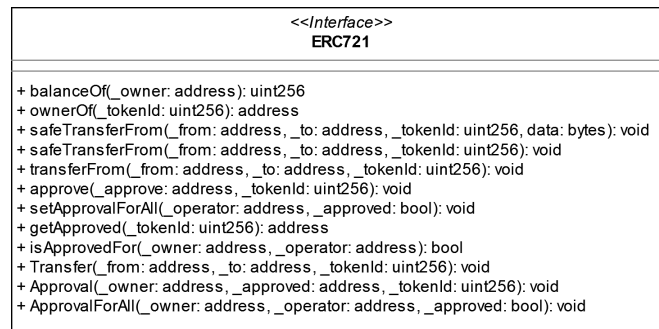


Fig. 2. UML class diagram of ERC-721 standard

Functions *balanceOf()* and *ownerOf()* return the amount of NFTs that an address owns or the address of a token owner, respectively.

Functions *safeTransferFrom()* and *transferFrom()* transfer ownership of a specific token from a previous owner to a new owner. Declarations of two *safeTransferFrom()* functions differ in *data* parameter which could be used to store additional information. Function *transferFrom()* does not perform validity checks related to the new owner.

Functions *approve()*, *setApprovalForAll()*, *getApproved()*, and *isApprovedForAll()* are providing a possibility for an entity other than the owner to be approved to transfer the ownership on behalf of the owner and to query information related to those possibilities.

Events *Transfer()*, *Approval()*, and *ApprovalForAll()* are emitted once corresponding functions are successfully executed.

Based on the described characteristics of ERC-20 and ERC-721, it is clear that neither of these two standards meets the needs for managing fractional ownership of non-fungible tokens. Mainly in ERC-20, there is no support for NFTs, while in ERC-721, there are no APIs that will enable storing data about fractional ownership or transferring less than full ownership of a token. To achieve this possibility, different solutions have been used over the years, most commonly creating a combination of two existing standards, but

having different implementations makes it hard for smart contracts to communicate with each other because there are no common APIs. Therefore, defining a new standard, with the intention to provide a common set of APIs for managing fractional ownership of non-fungible tokens would be beneficiary. The proposed standard is built upon existing ERC-20 and ERC-721 standards.

4. New ERC standards proposal

UML class diagram representing the proposed standard is shown in Fig. 3.

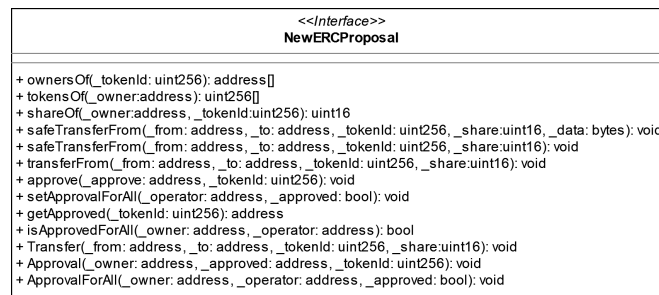


Fig. 3. UML class diagram of new ERC standard

Function *ownersOf(_tokenId: uint256): address[]* – unlike the function *ownerOf()* from ERC-721 where function call would return the only owner of a token, in case of the proposed function *ownersOf()*, for a token identifier, passed as a *_tokenId* argument of type *uint256*, the function returns the array of data type *address* representing the addresses of the owners of the token.

Function *tokensOf(_owner: address): uint256[]* – this function is based on function *balanceOf()* from ERC-721, but instead of providing a count of owned tokens, a call to proposed *tokensOf()* function for the owner who is identified by argument passed as a *_owner* argument of type *address*, the function returns the array of data type *uint256* representing identifiers of all the tokens that that specific owner owns. In case the function is called with an argument representing a zero address, an exception should be thrown.

Function *shareOf(_owner: address, _tokenId: uint256): uint16* – for the owner that is identified as *_owner* argument of data type *address*, the function will return the share of ownership as data type *uint16*, representing the share that that specific owner has in the token that identified as *_tokenId* argument of data type *uint256* that is passed in the function call. In case the function is called with an argument representing a zero address, an exception should be thrown.

Function *safeTransferFrom(_from: address, _to: address, _tokenId: uint256, _share: uint16, _data: bytes): void* – Similarly to *safeTransferFrom()* function ERC-721 this function for the address of the current owner, which is passed as the *_from* argument, to the address that is passed as *_to* argument, the ownership of the token whose identifier is passed as the *_tokenId* argument, is being transferred, with additional argument *_share*,

representing the share of ownership that is being transferred. The function keeps the additional parameter `_data` for the same purpose as is the case in ERC-721. The function does not return data. An exception should be thrown if the address that calls the function is not the owner of the token, if the address is not approved for the transfer of ownership of a specific token, if `_tokenId` is not a valid identifier if the `_to` argument is not a valid address or if transfer share specified in `_share` argument is bigger than the share the owner has in the specific token.

Function `safeTransfer(_from: address, _to: address, _tokenId: uint256, _share: uint16): void` – the function works as in the previous case with the difference that the function does not have an input parameter `data`, and the value `data` is set to an empty string (“”).

Function `transferFrom(_from: address, _to: address, _tokenId: uint256, _share: uint16): void` – from the address of the current owner, passed as the `_from` argument, to the address passed as the `_to` argument, ownership of the token whose identifier is passed as `_tokenId` is transferred in a share that is specified with argument `_share`. In this case, the function is not expected to check whether the address passed as the `_to` argument is valid, but the check should be done by the client calling the function. An exception will occur in the function if the address calling the function is not the owner of the token, if it is not approved for the transfer of ownership of the specific token, if `_tokenId` is not a valid identifier or if transfer share specified in `_share` argument is bigger than the share the owner has in the specific token.

Functions `approve()`, `setApprovalForAll()`, `getApproved()`, and `isApprovedForAll()` declare the same behavior as already presented in ERC-721, so they will not be presented again.

Event `Transfer(_from: address, _to: address, _tokenId: uint256, _share: uint16)` – an event that must be triggered in the case of a transaction and that will broadcast that ownership has been transferred from the address passed as the `_from` argument to the address passed as the `_to` argument over the token with the identifier passed in the `_tokenId` argument in a share passed as `_share` argument. The requirements set for `Transfer()` event in ERC-721 are valid in the case of this newly proposed `Transfer()` event.

Events `Approval()` and `ApprovalForAll()` declare the same behavior as already presented in ERC-721, so they will not be presented again.

In the following section, a simple implementation of the newly proposed ERC standard in the Solidity programming language will be presented and discussed.

5. Example of implementation of the proposed ERC standard

In this section, one implementation of the proposed ERC standard, written in the Solidity programming language, is presented. In Listing 1, the code representing the declaration of the new programming interface is shown, to be followed by examples of implementations of declared functions in Listings 2 through 6. Helper functions are presented in Listings 7 through 13, while error definitions are shown in Listing 14. The code is divided into several listings to make it easier to comment. In the presented listings, three dots replace the part of the smart contract code that is not relevant to the implementation currently being presented.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.17;
```

```

3
4 interface NewERCProposal {
5     function ownersOf(uint256 _tokenId) external view
6         returns (address[] memory);
7     function tokensOf(address _owner) external view
8         returns (uint256[] memory);
9     function shareOf(address _owner, uint256 _tokenId) external
10        view
11        returns (uint16);
12     function safeTransferFrom(address _from, address _to,
13        uint256 _tokenId, uint16 _share,
14        bytes memory _data) external;
15     function safeTransferFrom(address _from, address _to,
16        uint256 _tokenId, uint16 _share) external;
17     function transferFrom(address _from, address _to,
18        uint256 _tokenId, uint16 _share) external;
19     function approve(address _approve, uint256 _tokenId) external;
20     function setApprovalForAll(address _owner, address _operator)
21        external;
22     function getApproved(uint256 _tokenId) external view
23        returns (address);
24     function isApprovedForAll(address _owner, address _operator)
25        external view returns (bool);
26
27     event Transfer(address indexed _from, address indexed _to,
28        uint256 indexed _tokenId, uint16 _share);
29     event Approval(address indexed _owner,
30        address indexed _approved,
31        uint256 indexed tokenId);
32     event ApprovalForAll(address indexed _owner,
33        address indexed _operator,
34        bool indexed _approved);
35 }
36 ...

```

Listing 1. Interface declaration

In Listing 1 the functions are declared using the *function* reserved word, while events are declared using the *event* reserved word. In addition to these reserved words, the following reserved words are also used in the declaration of functions and methods: *external*, which represents one of the four function visibility specifiers in Solidity, *view*, which indicates that the function is not allowed to change the state of the blockchain, *memory*, which indicates that the argument passed to the function call will be saved only temporarily, during the execution of the function, and it will be deleted afterward, and *indexed*, which is used in events and indicates that arguments marked as *indexed* will be saved as a so-called topic and that events can be searched for by these values. In Listing 2, smart contract and state variables definitions are presented.

```

35 ...
36 contract FractionalOwnership is NewERCProposal {
37     address creator;
38     uint16 maximumShare;
39     mapping(uint256 => address[]) owners;
40     mapping(address => uint256[]) tokens;
41     mapping(uint256 => mapping(address => uint16)) share;
42     mapping(uint256 => address) approved;
43     mapping(address => address) approvedForAll;

```

```

44
45     constructor(uint16 _maximumShare) {
46         creator = msg.sender;
47         maximumShare = _maximumShare;
48     }
49     ...

```

Listing 2. Smart contract and state variables definitions

Listing 2 begins with the reserved word *contract*, followed by the name of the smart contract *FractionalOwnership*, and in line 36 it is declared that the smart contract implements the *NewERCProposal* interface. In listing 2, the following state variables are declared:

- *creator* – variable representing the address that was used to deploy the smart contract, and that will be used in the *mint()* function that will be presented in Listing 7;
- *maximumShare* – variable of type *uint16* that represents the maximum share in the ownership of a token. The value should be large enough to cover all the necessary use cases in the specific application;
- *owners* – variable of type *mapping*, mappings in the Solidity programming language represent the so-called key/value structure and in this case, it represents the relationship between the *uint256* value, which represents a token, and an array of *address* data type, which represent the owners of the token, is mapped;
- *tokens* — maps the relationship between the *address*, which represents the owner, and a series of *uint256* values, which represent tokens over which the address has ownership shares;
- *share* — maps the relationship between the *uint256* value, which represents the token, and the mapping that maps the relationship between the *address*, representing the owner, and the *uint16* value, representing the ownership share;
- *approved* - maps the relationship between the *address*, which represents the approved address for managing a token identified by *uint256* value;
- *approvedForAll* - maps the relationship in which key *address* grants the rights to value *address* to manage all of key address's tokens.

In addition to state variables, a constructor is declared and implemented in Listing 2. In smart contracts, the constructor is a function that is called only once, when the smart contract is placed on the blockchain network. In this particular case, the implementation of the smart contract is such that in line 46 the smart contract first queries which address sent the request for its creation and sets that value in the *creator* state variable and then in line 47, it sets the passed *uint16* argument in the *maximumShare* state variable. In Listing 3 implementation of *ownersOf()* function will be presented.

```

58     ...
59     function ownersOf(uint256 _tokenId) override external view
60         returns (address[] memory) {
61         if(isOwnerZeroAddress(_tokenId)) {
62             revert zeroAddress({
63                 _owner: address(0),
64                 _message:
65                     bytes("Zero address can not be queried.");
66             });

```

```

67     }
68     return owners[_tokenId];
69 }
70 ...

```

Listing 3. Example of implementation of *ownersOf()* function

The reserved word *override* in the *ownersOf()* function declaration in Listing 4 indicates that it is an implementation of the function declared in *NewERCProposal* interface. In lines 61 to 67, a check is made to see if the argument passed in the function call is bound to the zero address. This is done by calling *isOwnerZeroAddress()* function that will be presented Listing 10. In the case that this function call returns *true*, *revert()* function is called. The *revert()* function, will revert any changes that might have happened during the execution of the initial function call and throw *zeroAddress* error. The error *zeroAddress* accepts as parameters a zero address and the message that zero addresses cannot be queried. In case the call to the *isOwnerZeroAddress* function returns *false*, the function will perform a query on the *owners* state variable by passing the *_tokenId* argument and get a list of addresses representing all owners of a token. In Listing 4, implementations of *tokensOf()* and *shareOf()* functions will be presented.

```

70 ...
71     function tokensOf(address _owner) override external view
72         returns (uint256[] memory) {
73         if(_owner == address(0)) {
74             revert zeroAddress({
75                 _owner: address(0),
76                 _message:
77                     bytes("Zero address can not be queried.");
78             });
79         }
80         return tokens[_owner];
81     }
82
83     function shareOf(address _owner, uint256 _tokenId) override
84         public view returns (uint16){
85         if(_owner == address(0)) {
86             revert zeroAddress({
87                 _owner: address(0),
88                 _message:
89                     bytes("Zero address can not be queried.");
90             });
91         }
92         return share[_tokenId][_owner];
93     }
94 ...

```

Listing 4. Example of implementation of *tokensOf()* and *shareOf()* functions

The role of the *tokensOf()* function is to make it possible to find out all the tokens associated with a specific address. In lines 73 to 79 requirements related to zero address are checked and in case those requirements are met in line 80 array of tokens owned by the address for which the function is called is returned.

The *shareOf()* for the arguments representing the owner and a token, if requirements related to zero address are met, as shown in Lines 85 to 91, will in line 92 return the share

of ownership that *_owner* had over *_tokenId*. In Listing 4, implementations of *safeTransferFrom()* and *transferFrom()* functions will be presented.

```

94 ...
95     function safeTransferFrom(address _from, address _to,
96                             uint256 _tokenId, uint16 _share,
97                             bytes memory _data) override public {
98         if(_to == address(0)) {
99             revert zeroAddress({
100                 _owner: address(0),
101                 _message:
102                     bytes("Tokens can not be sent to zero address."
103                         )
104             });
105         }
106         transferFrom(_from, _to, _tokenId, _share);
107     }
108     function safeTransferFrom(address _from, address _to,
109                             uint256 _tokenId, uint16 _share)
110         override external {
111         safeTransferFrom(_from, _to, _tokenId, _share, "");
112     }
113     function transferFrom(address _from, address _to,
114                          uint256 _tokenId, uint16 _share)
115         override public {
116         checkIfTransferIsPermitted(_from, _tokenId, _share);
117
118         share[_tokenId][_from] -= _share;
119         share[_tokenId][_to] += _share;
120         addToTokensIfNewToken(_tokenId, _to);
121         removeFromOwnersIfNoShare(_tokenId, _from);
122         addToOwnersIfNewOwner(_to, _tokenId);
123         removeFromTokensIfNoShare(_from, _tokenId);
124         emit Transfer(_from, _to, _tokenId, _share);
125     }
126 }
127 ...

```

Listing 5. Example of implementation of *safeTransferFrom()* and *transferFrom()* functions

In Listing 5, starting from line 95, *safeTransferFrom()* function is implemented. Firstly in lines 98 through 104 requirements related to zero address are checked, to be followed by a call to *transferFrom()* function.

In lines 108 through 111, function implementation of *safeTransferFrom()* function is shown for the call that does not have *_date* parameter, or as required, call that has an empty string for *_date*.

Implementation of *transferFrom()* function is shown in lines 113 through 124. Firstly, function *checkIfTransferIsPermitted()* is called to check if necessary conditions for transfer have been met, this function will be presented in Listing 8. If all conditions are met, in lines 119 and 120 the share of ownership will be reduced and increased for old and new owners, to be followed by calls to function *addToTokensIfNewToken()*, *removeFromOwnersIfNoShare()*, *addToOwnersIfNewOwner()*, and *removeFromTokensIfNoShare()* in lines 121 through 124, for adding/removing tokens/owners from *owners* and *tokens* state vari-

ables. These functions will be presented in Listing 12 and 13. In line 125 required *Transfer()* event is being emitted. In Listing 6, implementations of functions *approve()*, *setApprovalForAll()*, *isApprovedForAll()*, and *getApproved()* are presented.

```

127 ...
128     function approve(address _approve, uint256 _tokenId)
129         override external {
130             require(isInOwners(msg.sender, _tokenId),
131                 "Caller is not the owner.");
132             if(_approve == address(0)) {
133                 revert zeroAddress({
134                     _owner: address(0),
135                     _message: bytes
136                         ("Zero address can not be approved")
137                 });
138             }
139             approved[_tokenId] = _approve;
140             emit Approval(msg.sender, _approve, _tokenId);
141         }
142
143     function setApprovalForAll(address _owner, address _operator)
144         override external {
145         approvedForAll[_owner] = _operator;
146     }
147
148     function isApprovedForAll(address _owner, address _operator)
149         override external view returns (bool) {
150         return approvedForAll[_owner] == _operator;
151     }
152
153     function getApproved(uint256 _tokenId) override external view
154         returns (address) {
155         return approved[_tokenId];
156     }
157 ...

```

Listing 6. Example of implementation of *approve()*, *setApprovalForAll()*, *isApprovedForAll()*, and *getApproved()* functions

In Listing 6, starting with line 128 *approve()* function is implemented. In lines 129 and 130 requirement that the caller has a share in the ownership of a token is checked. The requirement that zero address can not be approved is checked in lines 131 through 137. If all requirements are met in line 138 state variable *approved* is updated with the new approval and in line 139 *Approval()* event is emitted.

In lines 142 through 145 function *setApprovalForAll()* is implemented by mapping *_owner* and *_operator* in *approvedForAll* state variable in line 144.

Functions *isApprovedForAll()* and *getApproved()* are implemented in lines 147 through 150 and 152 through 155 respectively returning results of calls to *approvedForAll* and *approved* state variables.

```

49 ...
50     function mint(uint _tokenId) external {
51         require(msg.sender == creator,
52             "Sender not creator address.");
53         addToOwnersIfNewOwner(creator, _tokenId);
54         addToTokensIfNewToken(_tokenId, creator);

```

```

55         share[_tokenId][creator] = maximumShare;
56         emit Transfer(address(0), creator, _tokenId, maximumShare);
57     }
58     ...

```

Listing 7. Example of implementation of *mint()* function

The *mint()* is not declared in *NewERCProposal* interface, but it represents a common solution for the initial creation of tokens and usually, it is only available for the address that initially deployed the contract and that is why that address is preserved in the *creator* state variable. The function accepts the parameters *_tokenId*, which represents the identifier of the token to be created. In lines 52 and 53, the requirement that the function call came from the *creator* address is checked, and if this requirement is met in lines 54 through 56 functions required for the creation of new token are called in a similar way as it was the case with *transferFrom()* function. In line 57, *Transfer()* event is emitted. Helper function *checkIfTransferIsPermitted* is presented in Listing 8.

The remaining, helper functions will be presented in the following listings: in Listing 8 *checkIfTransferIsPermitted()* function, functions *isInOwners()* and *isInTokens()* are presented in Listing 9, to be followed by the implementation of *isOwnerZeroAddress()* functions in Listing 10. Listing 11 presents implementations of functions *getIndexOfOwner()* and *getIndexOfToken()* functions, while in Listing 12 *addToOwnersIfNewOwner()* and *removeFromOwnersIfNoShare()* are presented. Listing 13 holds implementations of functions *addToTokensIfNewToken()* and *removeFromTokensIfNoShare()*, to be concluded with error declarations in Listing 14.

```

157     ...
158     function checkIfTransferIsPermitted(address _from,
159                                       uint256 _tokenId, uint16 _share)
160                                       internal view{
161         if(_from == address(0)) {
162             revert zeroAddress({
163                 _owner: address(0),
164                 _message: bytes
165                     ("Transfers from zero address are not allowed.")
166             });
167         }
168
169         if (!isInOwners(msg.sender, _tokenId)
170             && !(msg.sender == approved[_tokenId]
171                 && !(msg.sender == approvedForAll[_from]))) {
172             revert notOwnerOrApproved({
173                 _tokenId: _tokenId,
174                 _from: _from
175             });
176         }
177
178         if (shareOf(_from, _tokenId) < _share) {
179             revert notOwningBigEnoughShare({
180                 _tokenId: _tokenId,
181                 _from: _from,
182                 _owningShare: shareOf(_from, _tokenId),
183                 _transferringShare: _share
184             });
185         }
186     }

```


187 ...

Listing 8. Example of implementation of *checkIfTransferIsPermitted()* function

```

187 ...
188     function isInOwners(address _address, uint256 _tokenId)
189         internal view returns (bool) {
190         address[] memory allOwners = owners[_tokenId];
191         for (uint i=0; i < allOwners.length; i++) {
192             if (allOwners[i] == _address ) {
193                 return true;
194             }
195         }
196         return false;
197     }
198
199     function isInTokens(uint256 _tokenId, address _address)
200         internal view returns (bool) {
201         uint256[] memory allOwned = tokens[_address];
202         for (uint i=0; i < allOwned.length; i++) {
203             if (allOwned[i] == _tokenId ) {
204                 return true;
205             }
206         }
207         return false;
208     }
209 ...

```

Listing 9. Example of implementation of *isInOwners()* and *isInTokens()* functions

```

209 ...
210     function isOwnerZeroAddress(uint256 _tokenId)
211         internal view returns (bool) {
212         address[] memory allOwners = owners[_tokenId];
213         for (uint i=0; i < allOwners.length; i++) {
214             if (allOwners[i] == address(0) ) {
215                 return true;
216             }
217         }
218         return false;
219     }
220 ...

```

Listing 10. Example of implementation of *isOwnerZeroAddress()* function

```

220 ...
221     function getIndexOfOwner(uint256 _tokenId,
222         address _owner)
223         internal view returns (int){
224         for(uint i = 0;
225             i < owners[_tokenId].length; i++){
226             if(_owner == owners[_tokenId][i])
227                 return int(i);
228         }
229         return -1;
230     }
231

```

```

232     function getIndexOfToken(uint256 _tokenId,
233                             address _owner)
234         internal view returns (int){
235     for(uint i = 0; i < tokens[_owner].length; i++){
236         if(_tokenId == tokens[_owner][i])
237             return int(i);
238     }
239     return -1;
240 }
241 ...

```

Listing 11. Example of implementation of *getIndexOfOwner()* and *getIndexOfToken()* functions

```

241 ...
242     function addToOwnersIfNewOwner(address _owner, uint256 _tokenId
243 )
244         internal {
245     if (!isInOwners(_owner, _tokenId)) {
246         owners[_tokenId].push(_owner);
247     }
248 }
249     function removeFromOwnersIfNoShare(uint256 _tokenId,
250 address _from) internal {
251     if (shareOf(_from, _tokenId) == 0) {
252         int i = getIndexOfOwner(_tokenId, _from);
253         if (i != -1) {
254             owners[_tokenId][uint(i)] =
255                 owners[_tokenId][owners[_tokenId].length - 1];
256             owners[_tokenId].pop();
257         }
258     }
259 }
260 ...

```

Listing 12. Example of implementation of *ownersOf()* function

```

260 ..
261     function addToTokensIfNewToken(uint256 _tokenId, address _owner
262 )
263         internal {
264     if (!isInTokens(_tokenId, _owner)) {
265         tokens[_owner].push(_tokenId);
266     }
267 }
268     function removeFromTokensIfNoShare(address _owner,
269 uint256 _tokenId) internal {
270     if (shareOf(_owner, _tokenId) == 0) {
271         int i = getIndexOfToken(_tokenId, _owner);
272         if (i != -1) {
273             tokens[_owner][uint(i)] =
274                 tokens[_owner][tokens[_owner].length - 1];
275             tokens[_owner].pop();
276         }
277 }

```

```
278     }
279     ...
```

Listing 13. Example of implementation of *ownersOf()* function

```
279     ...
280     error zeroAddress(address _owner, bytes _message);
281     error notApproved(address _from);
282     error notOwnerOrApproved(uint256 _tokenId, address _from);
283     error notOwningBigEnoughShare(uint256 _tokenId, address _from,
284                                   uint16 _owningShare,
285                                   uint16 _transferringShare);
286     error documentHashMustBeProvided(address _from, address _to,
287                                       uint256 _tokenId,
288                                       uint16 _share,
289                                       bytes _documentHash);
290 }
```

Listing 14. Example of implementation of *ownersOf()* function

6. Conclusion

BT has for some time been described as a technology that could be used in fields other than cryptocurrency and fintech. Various possible applications that would benefit from the advantages and characteristics of BT have been identified. The most common applications of BT in those filed are tied to smart contracts and either fungible or NFTs. NFTs are recognized as a possible way to represent unique items from the real world in a blockchain network. What was identified as possible improvements related to NFTs was a standard set of APIs for the representation of fractional ownership of NFTs. In this paper, a proposal for a new ERC is made in a form of a UML class diagram and an interface written in Solidity programming language. Also, the implementation of the proposed interface is presented in a form of a smart contract, together with all the required constraints. The adoption of such a new ERC would define a standard set of APIs for exchanging function calls that would solve the issue that was usually resolved by combining multiple standards.

Future research could be directed into the possible optimization of smart contracts implementing the proposed new ERC. The costs associated with running smart contracts on the Ethereum network are related to the gas spent during the prices of execution of a transaction. Limiting those costs should be of concern. This concern is of especially big importance in use cases where there are a significant number of NFTs managed by a single smart contract. Preliminary research on the proposed implementation shows that the amount of gas spent related to the transaction could vary between 140.000 and 2.800.000 gas. Another possible research might be related to cost comparison between the application of a single smart contract that will manage multiple NFTs, or multiple smart contracts, each representing a single NFT.

References

1. Abdelmaboud, A., Ahmed, A., Abaker, M., Eisa, T., Albasheer, H., Ghorashi, S., Karim, F.: Blockchain for iot applications: Taxonomy, platforms, recent advances, challenges and future research directions. *Electronics* 11(4) (2022)

2. Aggarwal, S., Kumar, N., Chelliah, P.: Cryptographic consensus mechanisms. *Advances in Computers* 121(4), 211–226 (2021)
3. Alamri, M., Jhanjhi, N., Humayun, M.: Blockchain for internet of things (iot) research issues challenges & future directions: A review. *International Journal of Computer Science and Network Security* 19(5), 248–258 (2019)
4. Alotaibi, L., Alshamrani, S.: Smart contract: Security and privacy. *Computer Systems Science and Engineering* 38(1), 93–101 (2021)
5. Azzi, R., Chamoun, R., Sokhn, M.: The power of a blockchain-based supply chain. *Computers & Industrial Engineering* 135 (2019)
6. Bao, H., Roubaud, D.: Non-fungible token: A systematic review and research agenda. *Journal of Risk and Financial Management* 15(215) (2022)
7. Bennett, R., Miller, T., Pickering, M., Kara, A.: Hybrid approaches for smart contracts in land administration: Lessons from three blockchain proofs-of-concept. *Land* 10(2) (2021)
8. Buterin, V.: Ethereum white paper - a next generation smart contract & decentralized application platform. https://blockchainlab.com/pdf/Ethereum_white_paper_a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (2015), accessed: 2022-12-04
9. Cai, W., Wang, Z., Ernst, J., Hong, Z., Feng, C., Leung, V.: Decentralized applications: The blockchain-empowered software system. *IEEE Access* 6, 53019–53033 (2018)
10. Cao, X., Zhang, J., Wu, X., Liu, B.: A survey on security in consensus and smart contracts. *Peer-to-Peer Networking and Applications* 15, 1008–1028 (2022)
11. Casino, F., Dasaklis, T., Patsakis, C.: A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics* 36, 55–81 (2019)
12. Chowdhury, M., Ferdous, M., Biswas, K., Chowdhury, N., Kayes, A., Alazab, M., Watters, P.: A comparative analysis of distributed ledger technology platforms. *IEEE Access* 7, 167930–167943 (2019)
13. Dai, H., Zheng, Y., Zhang, Y.: Blockchain for internet of things: A survey. *IEEE Internet of Things Journal* 6(5), 8076–8094 (2019)
14. Dowling, M.: Is non-fungible token pricing driven by cryptocurrencies? *Finance Research Letters* 44 (2022)
15. Entriken, W., Shirley, D., Evans, J., Sachs, N.: Eip-721: Non-fungible token standard. <https://eips.ethereum.org/EIPS/eip-721> (2018), accessed: 2022-10-23
16. Ethereum: Ethereum improvement proposals. <https://eips.ethereum.org/> (2022), accessed: 2022-12-04
17. Giancaspro, M.: Is a 'smart contract' really a smart idea? insights from a legal perspective. *Computer Law & Security Review* 33(6), 825–835 (2017)
18. Guido, R., Mirabelli, G., Palermo, E., Solina, V.: A framework for food traceability: Case study–italian extra-virgin olive oil supply chain. *International Journal of Industrial Engineering and Management* 11(1), 50–60 (2020)
19. Hassija, V., Chamola, V., Krishna, D., Kumar, N., Guizani, M.: A blockchain and edge-computing-based secure framework for government tender allocation. *IEEE Internet of Things Journal* 8(4), 2409–2418 (2021)
20. Hood, D.: The most expensive nfts ever sold. <https://www.business2community.com/nft/most-expensive-nft> (2022), accessed: 2022-11-23
21. Huo, R., Zeng, Z., Wang, J., Shang, W., Chen, T., Huang, S., Wang, R., Yu, Y., Liu, Y.: A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges. *IEEE Communications Survey & Tutorials* 24(1), 88–122 (2022)
22. Hussien, H., Yasin, S., Yan, J., Udzir, N., Ninggal, M., Salman, S.: Blockchain technology in the healthcare industry: Trends and opportunities. *Journal of Industrial Information Integration* 22 (2021)

23. Kassen, M.: Blockchain and e-government innovation: Automation of public information processes. *Information Systems* 103 (2022)
24. Le, T., Hsu, C.: A systematic literature review of blockchain technology: Security properties, applications and challenges. *Journal of Internet Technology* 22(4), 789–801 (2021)
25. Lu, y.: The blockchain: State-of-the-art and research challenges. *Journal of Industrial Information Integration* 15, 80–90 (2019)
26. Luu, L., Chu, D., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. pp. 254–269. Association for Computing Machinery, New York, NY, United States (2016)
27. Majeed, U., Khan, L., Yaqoob, I., Kazmi, S., Salah, K., Hong, C.: Blockchain for iot-based smart cities: Recent advances, requirements, and future challenges. *Journal of Network and Computer Applications* 181 (2021)
28. Marjanović, J., Dalčeković, N., Sladić, G.: Blockchain-based model for tracking compliance with security requirements. *Computer Science and Information Systems* 20(1), 359–380 (2023)
29. Matulevicius, R., Iqbal, M., Elhadjamor, E., Ghannouchi, S., Bakhtina, M., Ghannouchi, S.: Ontological representation of healthcare application security using blockchain technology. *INFORMATICA* 33(2), 365–397 (2022)
30. Mohsin, A., Zaidan, A., Zaidan, B., Albahri, A., Albahri, M., Alsalem, M., Mohammed, K.: Ontological representation of healthcare application security using blockchain technology. *INFORMATICA* 33(2), 365–397 (2022)
31. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf> (2009), accessed: 2022-11-28
32. Ning, X., Ramirez, R., Khuntia, J.: Blockchain-enabled government efficiency and impartiality: using blockchain for targeted poverty alleviation in a city in china. *Information Technology for Development* 27(3), 599–616 (2021)
33. Park, A., Kietzmann, J., Pitt, L., Dabirian, A.: The evolution of nonfungible tokens: Complexity and novelty of nft use-cases. *IT Professional* 24(1), 9–14 (2022)
34. Peng, K., Li, M., Huang, H., Wang, C., Wan, S., Choo, K.: Security challenges and opportunities for smart contracts in internet of things: A survey. *IEEE Internet of Things Journal* 8(15), 12004–12020 (2021)
35. Politou, E., Casino, F., Alepis, E., Patsakis, C.: Blockchain mutability: Challenges and proposed solutions. *IEEE Transactions on Emerging Topics in Computing* 9, 1082–1986 (2021)
36. Risius, M., Spohrer, K.: A blockchain research framework: What we (don't) know, where we go from here, and how we will get there. *Business & Information Systems Engineering* 59(6), 385–409 (2017)
37. Saberi, S., Kouhizadeh, M., Sarkis, J., Shen, L.: Blockchain technology and its relationships to sustainable supply chain management. *International Journal of Production Research* 57(7), 2117–2135 (2019)
38. Saini, H., Dash, S., Kumar Pani, S., Jos e Sousa, M., Rocha, A.: Blockchain-based raw material shipping with poc in hyperledger composer. *Computer Science and Information Systems* 19(3), 1075–1092 (2022)
39. Shahaab, A., Lidgley, B., Hewage, C., Khan, I.: Applicability and appropriateness of distributed ledgers consensus protocols in public and private sectors: A systematic review. *IEEE Access* 7, 43622–43636 (2019)
40. Sladić, G., Milosavljević, B., Nikolić, S., Sladić, D., Radulović, A.: A blockchain solution for securing real property transactions: A case study for serbia. *ISPRS International Journal of Geo-Information* 10(1) (2021)
41. Stefanovic, M., Pržulj, D., Stefanović, D.: Making smart contracts smarter. In: *Book of abstracts of Symorg 2022*. pp. 10–12. Faculty of Organizational Sciences, University of Belgrade, Belgrade, Serbia (2022)
42. Stefanović, M., Pržulj, D., Ristić, S., Stefanović, D., Nikolić, D.: Smart contract application for managing land administration system transactions. *IEEE Access* 10, 2169–3536 (2022)

43. Szabo, N.: Smart contracts: Building blocks for digital markets. *Extropy*, 16, 50–53, 61–63 (1996)
44. Vogelsteller, F., Buterin, V.: Eip-20: Epc-20 token standard. <https://eips.ethereum.org/EIPS/eip-20> (2015), accessed: 2022-10-23
45. Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., Wang, F.: Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49(11), 2266–2277 (2019)
46. Woznica, A., Kedziora, M.: Performance and scalability evaluation of a permissioned blockchain based on the hyperledger fabric, sawtooth and iroha. *Computer Science and Information Systems* 19(2), 659–678 (2022)
47. Wu, K., Ma, Y., Huang, G., Liu, X.: A first look at blockchain-based decentralized applications. *Software: Practice and Experience* 51, 2033–2050 (2021)
48. Xuan, S., Zheng, L., Chung, I., Wnag, W., Man, D., Du, X., Yang, W., Guizani, M.: An incentive mechanism for data sharing based on blockchain with smart contracts. *Computers and Electrical Engineering* 83 (2020)
49. Yang, L.: The blockchain: State-of-the-art and research challenges. *Journal of Industrial Information Integration* 15, 80–90 (2019)
50. Yaqoob, I. and Salah, K., Jayaraman, R., Al-Hammadi, Y.: Blockchain for healthcare data management: opportunities, challenges, and future recommendations. *Neural Computing & Applications* 34(44), 11475–11490 (2022)
51. Young, M.: Over 44 million contracts deployed to ethereum since genesis: Research. <https://cryptopotato.com/over-44-million-contracts-deployed-to-ethereum-since-genesis-research/> (2022), accessed: 2022-11-16
52. Zafar, S. and Bhatti, K., Shabbir, M., Hashmat, F., Akbar, A.: Blockchain for healthcare data management: opportunities, challenges, and future recommendations. *Annals of Telecommunications* 77(1-2), 13–32 (2021)
53. Zhao, J., Fan, S., Yan, J.: Overview of business innovations and research opportunities in blockchain and introduction to the special issue. *Financial Innovation* 2(28) (2016)
54. Zheng, Z., Xie, S., Dai, H., Chen, X., Chen, J., Weng, J., Imran, M.: An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems* 105, 475–491 (2020)
55. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* 14(2), 352–375 (2018)

Miroslav Stefanović received his B.S. degree in information management in 2014 and the M.S. degree in information systems engineering in 2016 from the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia. He received the Ph.D. degree in industrial engineering and engineering management from the same institution in 2023. From 2015 to 2016, he was a Teaching Associate and since 2016, he is a Teaching Assistant at the University of Novi Sad, Faculty of Technical Sciences, Department of Industrial Engineering and Engineering Management, Chair for Information and Communication Systems. His research interests include blockchain technologies, especially the implementation of blockchain technology in fields other than cryptocurrency, mainly e-government and land administration systems.

Dorđe Pržulj received his B.S. degree in mechanical engineering in 1999 and the M.S. degree in industrial engineering and engineering management in 2004 from the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia. He received the Ph.D.

degree in industrial engineering and engineering management from the same institution in 2013. From 1999 to 2013 he was a Teaching Assistant and from 2013 to 2018 Assistant Professor at the University of Novi Sad, Faculty of Technical Sciences, Department of Industrial Engineering and Engineering Management, Chair for Information and Communication Systems. Since 2018 he has been an Associate Professor at the same institution. His research interests include land administration systems, service oriented architecture, microservices, ontologies, domain specific languages, and blockchain applications in information systems. Đorđe Pržulj has published in several international information systems journals.

Sonja Ristić works as a full professor at the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia. She received two bachelor's degrees with honors from the University of Novi Sad, one in Mathematics, Faculty of Science in 1983, and the other in Economics from Faculty of Economics, in 1989. She received her Mr (2 years) and Ph.D. degrees in Informatics, both from the University of Novi Sad, Faculty of Economics, in 1994 and 2003. From 1984 until 1990 she worked with the Novi Sad Cable Company NOVKABEL–Factory of Electronic Computers. From 1990 till 2006 she was with Novi Sad School of Business, and since 2006 she has been with the University of Novi Sad, Faculty of Technical Sciences. Her research interests include database systems, software engineering, model-driven software engineering and domain specific languages. She is the author or co-author of over 100 papers, and 10 industry projects and software solutions in the area.

Darko Stefanović received his B.S. degree in mechanical engineering in 1999 and the Mr degree in industrial engineering and engineering management in 2005 from the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia. He received the Ph.D. degree in industrial engineering and engineering management from the same institution in 2012. From 2001 to 2012, he was a Teaching Assistant and from 2012 to 2017, Assistant Professor at the University of Novi Sad, Faculty of Technical Sciences, Department of Industrial Engineering and Engineering Management, Chair for Information and Communication Systems. Since 2017 he is Associate Professor at the same institution. He is also a vice-dean for Science and International Cooperation at the University of Novi Sad, Faculty of Technical Sciences, and head of Chair of Information and Communication Systems. His research interest includes ERP systems, e-learning systems, e-government systems, data mining, and business process mining in production planning. Darko Stefanovic has published in several international information systems journals. s

Darko Čapko received the Ph.D. degree from the University of Novi Sad, in 2012. He is a Full Professor at Faculty of Technical Sciences, University of Novi Sad and CSO at Eternal, Novi Sad. He has published over 80 articles and participated in more than 20 projects. His research interests are related to distributed algorithms, cryptography, blockchain and artificial intelligence.

Received: January 27, 2023; Accepted: May 04, 2023.

A Novel Multi-objective Learning-to-rank Method for Software Defect Prediction

Yiji Chen, Lianglin Cao, and Li Song

Jiujiang University
Jiujiang 332005, China
chenyiji1984@jju.edu.cn
charlies_navy@aliyun.com
songli@jju.edu.cn

Abstract. Search-Based Software Engineering (SBSE) is one of the techniques used for software defect prediction (SDP), in which search-based optimization algorithms are used to identify the optimal solution to construct a prediction model. As we know, the ranking methods of SBSE are used to solve insufficient sample problems, and the feature selection approaches of SBSE are employed to enhance the prediction model's performance with curse-of-dimensionality or class imbalance problems. However, it is ignored that there may be a complex problem in the process of building prediction models consisting of the above problems. To address the complex problem, two multi-objective learning-to-rank methods are proposed, which are used to search for the optimal linear classifier model and reduce redundant and irrelevant features. To evaluate the performance of the proposed methods, excessive experiments have been conducted on 11 software programs selected from the NASA repository and AEEEM repository. Friedman's rank test results show that the proposed method using NSGA-II outperforms other state-of-the-art single-objective methods for software defect prediction.

Keywords: Search-Based Software Engineering, software defect prediction, multi-objective optimization algorithm, ranking method.

1. Introduction

Data quality [17], software metric, and classification algorithm are the main factors in constructing a promising prediction model [27]. Thereby, many problems need to be addressed in the process of construction. Machine learning algorithm is the most popular classification algorithm for SDP tasks, and it has a minimum requirement for the number of training samples [8]. Because of the expensive efforts, the samples are difficult to be collected in some systems, thus causing insufficient samples. Where engineers and researchers use more metrics to collect information from software features, redundant and irrelevant features can affect the efficiency of predictive models, leading to the curse of dimensionality problems. In addition, the number of normal samples is much more than the number of defective samples called the class imbalance problem. For this problem, Yu et al. [33] propose two extended resampling strategies that effectively handle imbalanced defect data to predict the number of defects.

Since SBSE was proposed by Harman [14], it has attracted more scholars and addressed the above problems in software defect prediction. A learning-to-rank method fits

a linear classifier model to allow the data with insufficient samples problem. A feature selection method solves the curse-of-dimensionality and class imbalance problem. Especially, a search-based optimization algorithm is employed to search optimal feature subsets [25],[13], [24]. Although these approaches can address a single problem for SDP, there are few methods for a complex problem which is consisted of the above problems.

Even though ensemble predictors like Random Forest, k-Nearest Neighbors, Support Vector Machine, etc., are state-of-the-art for SDP tasks, these do not obtain a promising performance on NASA datasets. In our previous work [15], the experimental results show that the performance of these set predictors without the feature selection methods is quite the same as that of random predictors. In other words, obtaining a promising SDP model on NASA datasets is easier with other assist methods like feature selection. In addition, some metrics like AUC [12], F-Measure, etc., are always used to evaluate the performance of SDP models. However, fault-percentile-average (FPA) is frequently employed to estimate the performance of SDP models in ranking tasks. In this study, two multi-objective optimization algorithms are employed to obtain promising regression predictors in SDP ranking tasks, optimizing the parameters of regression predictors and searching for an optimal feature subset. FPA is used to evaluate the performance of the proposed methods in solving a complex problem, including insufficient samples, curse-of-dimensionality, and class imbalance.

The remainder of this paper is organized as follows: some related work is introduced in Section 2. The details of the SBSE multi-objective ranking method are described in Section 3. Section 4 shows the experimental studies conducted on NASA datasets. Finally, conclusions and future work are drawn in Section 5.

2. Literature Review

Software defect prediction is an important technique that can guide engineers to assign limited resources to focus on probable defect-proneness [6]. SBSE is one of the most popular approaches to improving the performance of SDP. Generally speaking, the ranking method and feature selection using SBSE effectively solve different problems in SDP tasks.

2.1. SBSE Ranking Method

The classification and ranking models are the two most frequently used prediction models. The goal of the ranking model is to predict an order of software modules based on the predicted scores of each module, and the goal of the classification model is to predict whether a software module has a defect or not. Where there is an insufficient sample problem, the ranking model is more efficient than the classification model in predicting the defect-prone software modules. The ranking method is used to build a ranking model, including the point-wise, pair-wise, and list-wise approaches. The SBSE ranking method, called Learning-To-Rank (LTR), is one of the list-wise approaches that optimizes performance measures to obtain a ranking model.

Yang et al. [29] first use CoDE to obtain the coefficients of the linear models instead of classification models by least squares (LS) and use FPA to evaluate the ranking performance. The experimental results on five data sets show that the proposed method is

competitive with others ranking methods employing machine learning algorithms. Based on this investigation, they improve their studies that a feature selection method InfoGain is used to select a subset of metrics for the ranking task. Their empirical studies demonstrate that the feature selection method is beneficial to obtain a linear model based on data sets with large metrics [30]. However, the performance of the proposed LTR method depends too much on the effectiveness of InfoGain.

Buchari et al. [4] propose a novel LTR approach using the Chaotic Gaussian Particle Swarm Optimization algorithm (CGPSO) to optimize the ranking performance. Compared with the LTR using CoDE, it improves the performance of FPA on most of the eleven data sets. Peng et al. [20] propose the NABC algorithm to search the optimal coefficients of the linear models. It also obtains a better FPA than the LTR using CoDE. An empirical study of the LTR method [32] compares 23 ranking algorithms on 41 data sets from the PROMISE repository. In the comparison, the LTR method obtains the best ranking performance. Li et al. [16] consider the SDP problem as multiple goals to be optimized so that the revised NSGA-II is used to optimize the ranking performance FPA and prediction accuracy for the ranking task. The experimental results indicate that multiple-objective optimization algorithms can further improve the performance of the LTR.

Based on these investigations, one can use the LTR method to improve the ranking performance for SDP with insufficient samples. However, more research needs to be done on addressing SDP alongside other problems using the LTR method.

2.2. SBSE Feature Selection Method

Various software metrics have been proposed to provide vital information for constructing SDP models. Moser et al. [18] propose that adopting change metrics is more beneficial for predictive performance than static code attributes. Bell et al. [3] propose to increase the developer metrics to improve the prediction performance further. Choudhary et al. [7] propose that mixed metrics are the best choice the more metrics used in the dataset, the more serious the dimension problem. The SBSE feature selection method is an effective approach to reducing redundant and irrelevant features. Additionally, it can be used to search for an optimal feature subset to address class imbalance problems.

Balogun et al. [2] propose a study of performance analysis of feature selection methods using exhaustive and heuristic search. The performance of 7 search methods is evaluated using four classification algorithms on 5 data sets from the NASA repository. Although using search methods for feature selection can effectively improve prediction accuracy, the impact of the class imbalance problem on prediction performance cannot be evaluated.

Turabieh et al. [26] provide a novel feature selection algorithm with a layered recurrent neural network for software fault prediction. Genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO) algorithms are randomly selected to search for an optimal feature subset in each iteration. The results of performing 5-fold cross-validation experiments on 19 data sets selected from the PROMISE repository using 20 metrics show that the proposed method improves the performance measure AUC of the classification models.

Proposed by Balogun et al. [1], the performance of 13 SBSE feature selection methods is verified on 7 data sets. It can be concluded that feature selection based on the meta-heuristic search methods outperforms others. An empirical study is shown by Nguyen

et al. [19], because of the efficiency of swarm-based intelligence algorithms. These have been embedded in feature selection to search for an optimal feature subset. Rostami et al [21] showed more feature selection applications using swarm intelligence algorithms.

Based on the above investigations, the SBSE ranking methods enhance the performance of ranking models with insufficient samples problem. The SBSE feature selection approaches improve the efficiency of the training process to obtain a classification model by reducing redundant and irrelevant features. However, few studies consider the complex problem, which contains insufficient samples, class imbalance, and curse-of-dimensionality. Thereby, the issue is still an open question to be addressed. In this paper, we propose a novel multi-objective learning-to-rank method using two multi-objective optimization algorithms, which optimizes two objectives consisting of the optimal coefficients of the linear model and the optimal features subsets. Besides, the 10-fold cross-validation approach is used to verify the class imbalance problems.

3. Proposed Methodology

In this section, MSFFA [5] and NSGA-II [10] are employed to optimize the performance of the ranking model based on some datasets with a complex problem. Minimizing the size of feature subsets and maximizing FPA values are the two goals to achieve for complex problems. Additionally, the 10-fold cross-validation method is employed for the class imbalance problem. The 10-fold cross-validation makes little sense because splitting the datasets into ten folds does not affect the class imbalance problem and may only deteriorate it further. In this case, if the SDP model achieves promising performance, it can be shown that the adopted method solves well the class imbalance problem of SDP.

3.1. Optimization Algorithms

FA is an efficient swarm intelligence algorithm proposed by Yang [31]. In the search space, due to the self-learning and self-organizing capability of the population, it can effectively search for optimal solutions to objective functions by evaluating the fitness of each firefly location. The pseudo-code of FA is expressed in Alg. 1.

To enhance the efficiency of FA, two strategies were proposed in our previous work, including the multi-swarm strategy and the free strategy. While the multi-swarm strategy reduces the redundant attractions, the free strategy guides the population to adaptively change its state to balance the search ability between exploration and exploitation.

Each individual can update its position followed two status. In each interaction, where the weakness firefly moves according to the FA rules shown in Equ. 1, the brightness firefly follows free rules expressed in Equ. 2.

$$X_i(t+1) = x_i(t) + \beta_0 e^{(-\gamma r_{ij}^2)} (x_j(t) - x_i(t)) + \alpha \varepsilon \quad (1)$$

Where γ stands for the light absorption coefficient, β_0 expresses the attractiveness when $\gamma = 0$, r_{ij} represents the Euclidean distance between the firefly i and the firefly j , and $x_i(t)$ indicates the position of firefly i in t^{th} iteration. α is a control parameter and ε is a random vector in $[0,1]$.

$$x_i(t+1) = x_i(t) + \mu \left(\frac{t+1}{t} \right)^{times_i} (x_i(t) - x_r) \quad (2)$$

Algorithm 1 Firefly Algorithm

```

Initialize population and generate  $N$  fireflies  $i$ ,  $i = 1, 2, \dots, N$ , the maximum evaluations
MaxFEs;
while FEs  $\leq$  MaxFEs do
  for  $i=1$  to  $N$  do
    for  $j=1$  to  $i$  do
      if fitness( $i$ ) > fitness( $j$ ) then
        Update the location of the firefly  $i$  and evaluate the fitness( $i$ );
      end if
    end for
  end for
  Rank the fitness for all fireflies and find the best solution;
   $t=t+1$ ;
end while
Output the optimal solution;

```

$x_i(t)$ represents the position of the current firefly, x_r is the position of a firefly random selected in the entire population, t is the t^{th} iteration, μ is a random value between $[0, 1]$, $times_i$ is the number of times that i has moved, the new position is $x_i(t+1)$.

NSGA-II is the other multi-objective algorithm used in this study. We use distribution indexes of NSGA-II for crossover and mutation operators as 20, the crossover probability is 0.9 and the mutation probability is 0.05. The pseudo-code of NSGA-II is expressed in Alg.2

Algorithm 2 NSGA-II

```

Initialize: Set population size (number of solutions) to  $N$ , and randomly generate  $N$  solutions that
compose population  $P_0$ . Sort the solutions in  $P_0$  using fast non-dominated sorting, and compute
the non-dominated rank value of each solution.
Evaluate the fitness of the multi-objective function for each solution, and sort the solutions in  $P_0$ ,
and compute the rank value of each solution. Set the generation number  $t = 0$ ;
while  $t \leq t_{max}$  do
  Use binary tournament selection to select individuals from  $P_t$  for crossover and mutation to
  generate the offspring population  $Q_t$ ;
  Combine solutions in  $P_t$  and  $Q_t$  to get  $R_t = P_t \cup Q_t$ ;
  Sort  $R_t$  based on non-domination rank value and crowding distance, and select  $N$  elitist indi-
  viduals to compose the new population  $P_{t+1}$ ;
   $t=t+1$ ;
end while
Return Pareto-optimal solutions in  $P_{t+1}$ ;

```

3.2. Learning-to-rank method Using Multi-objective Optimization Algorithm

Once the features of a software module X are extracted by d metrics, it can be expressed as

$$X = (x_1, x_2, \dots, x_d) \quad (3)$$

The task of the proposed method is to predict the defect number of the module, which can be denoted as $f(X)$. We study a simple linear model which is good and realistic for SDP proposed by Weyuker et al [28]:

$$f(X) = \sum_{i=1}^d a_i x_i \quad (4)$$

If the parameters a_i is fixed, the prediction model is obtained.

Considering insufficient samples, we use optimization algorithms instead of machine learning algorithms to optimize the parameters to obtain prediction models. Obtaining an ordered list according to $f(x)$ is not a good choice. We use FPA to evaluate the obtained prediction models proposed by [28]. Different from other learning-to-rank methods [29][30][20]), we add another task of the proposed method is to select more essential features to improve the efficiency of the obtained prediction models. Once the performance of the obtained prediction model is enhanced by reducing the redundant and irrelevant features, the proposed method can also address the curse-of-dimensionality problem or class imbalance problem.

A similar study proposed by Yang et al.[30] uses filter-based feature selection methods before training a prediction model. We know those wrapper-based feature selection methods are more competitive than filter-based feature selection methods. We employ a wrapper-based feature selection method using optimization algorithms to optimize the goals of SDP for the feature selection task.

From the above goals of SDP for the ranking task, a multi-objective optimization algorithm is used to obtain a promising prediction model addressing insufficient samples, class imbalance, and curse-of-dimensionality. Therefore, two fitness functions are designed to evaluate the performance of the prediction models obtained by the multi-objective optimization algorithm. First, FPA is used to evaluate the ranking performance. The equation of FPA is defined as Equ.5.

$$f_{FPA} = \frac{1}{k} \sum_{m=1}^k \frac{1}{n} \sum_{i=k-m+1}^k n_i \quad (5)$$

Setting k as the modules number, f_1, f_2, \dots, f_k listed in an increasing order of predicted defect number, n_i as the actual defect number of the modules i , and $n = n_1 + n_2 + \dots + n_k$ as the total number of defects. FPA is an average of the proportions of actual defects in the top i predicted modules, the larger FPA, the better the performance.

Next, The ratio of selected features to total features is another fitness function shown in Equ.6.

$$f_{ratio} = \frac{S}{T} \quad (6)$$

S represents the number of selected features, T as the total number of features. The solution representation is defined that the solution is encoded as a binary vector of length equal to the total number of features shown in Fig. 1

In Fig. 1, N indicates the total number of features, and R is the selection threshold. x_i is a vector between 0 to 1, which indicates the index of the i^{th} feature. So that one optimization algorithm can search the index of the features to compose an optimal feature subset. If the searched value x_i is greater than R , the i^{th} feature is selected so that y_i is set to 1. Otherwise, where the searched value x_j is less than R , the j^{th} feature is not selected

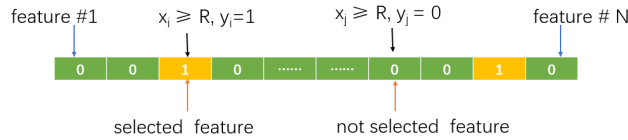


Fig. 1. An example of a feature selection solution.

that y_j is set to 0. R is set to 0.5 in this paper. Therefore, the smaller the ratio of selected features, the more effective the ranking model is.

Additionally, the last task of the proposed method is to verify the class imbalance problem, in which the 10-fold cross-validation method is employed. The 10-fold cross-validation makes little sense because splitting the datasets into ten folds does not affect the class imbalance problem and may only deteriorate it further. In this case, if the SDP model achieves promising performance, it can be shown that the adopted method solves well for the class imbalance problem of SDP.

3.3. Proposed Multi-Objective learning-to-rank Algorithm

This study uses MSFFA and NSGA-II as search techniques to obtain a set of Pareto-optimal solutions for prediction models. The procedure of the proposed algorithm is shown in Fig. 2

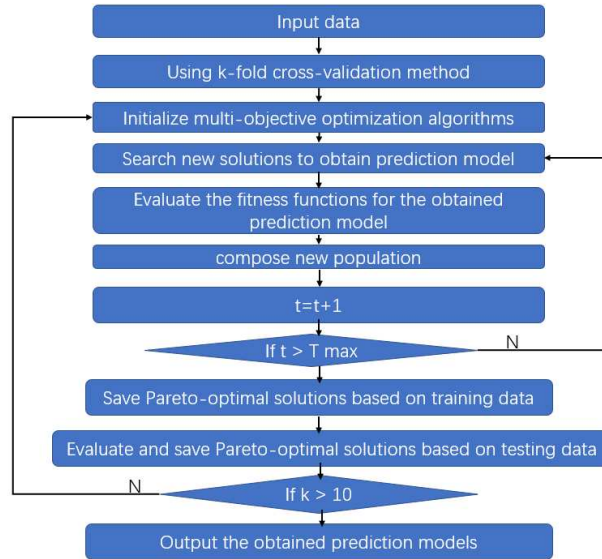


Fig. 2. The procedure of the proposed algorithm

First, the 10-fold cross-validation method is used in the search process to obtain a fair prediction model. Then, MSFFA and NSGA-II are employed to search for better solutions for both fitness functions. Last, when the termination condition is reached, a set of Pareto-optimal solutions are searched and saved. The average value of the best-level solutions is computed and saved in experimental results. The pseudo-code of the proposed algorithm is shown in Alg.3.

Algorithm 3 Proposed Multi-Objective learning-to-rank Algorithm

Initialize: load data, set data-length to D , set population size (number of solutions) to N , and randomly generate N solutions that compose population P_0 .

Using k-fold cross-validation method;

for k = 1: to 10 **do**

Evaluate the fitness of the multi-objective function for each solution, and sort the solutions in P_0 , and compute the rank value of each solution. Set the generation number $t = 0$;

while $FES \leq MaxFES$ **do**

Use MSFFA (NSGA-II) to search new solutions that compose population Q_t ;

Evaluate the fitness of the multi-objective function for each solution in Q_t ;

Combine solutions in P_t and Q_t to get $R_t = P_t \cup Q_t$;

Sort R_t based on non-domination rank value and crowding distance, and select N elitist individuals to compose the new population P_{t+1} ;

$t=t+1$;

end while

Return Pareto-optimal solutions in P_{t+1} ;

end for

4. Experimental Result and Discussion

In the experiments, 11 software programs selected from NASA [22] and AEEEM [9] are used to verify the performance of the proposed method for a complex problem. More details of the programs are shown in Tab.1.

Table 1. The details of the programs selected from NASA datasets and AEEEM datasets

	program	features	modules	defective modules	ratio
NASA	PC1	40	1107	76	0.074
	MC1	38	9466	68	0.007
	MW1	39	403	31	0.077
	JM1	21	10878	2102	0.193
	CM1	37	327	42	0.128
	KC1	21	2107	325	0.154
AEEEM	Eclipse JDT Core	15	997	463	0.464
	Eclipse PDE UI	15	1562	401	0.257
	Equinox framework	15	439	279	0.636
	Mylyn	15	2196	677	0.308
	Apache Lucene	15	691	103	0.149

From Tab.1, it can be seen that not only a few samples can be used to train prediction models, but also more metrics are used to extract software features for most software

programs. Besides, the number of defective modules is far more than that of normal modules. Therefore, a complex problem may exist in most of the programs, that the proposed approach's performance can be evaluated by employing these software programs in this study.

To investigate the efficiency of the proposed approach, four single objective learning-to-rank algorithms are compared to build a ranking model for SDP, including CoDE, NABC, PSO proposed by D'Ambros et al.[11], and BSO proposed by Shi et al.[23]. A maximum of 1000 individual evaluations is the termination iteration condition for all algorithms. To verify the performance of the proposed multi-objective ranking methods, the 10-fold cross-validation method is employed to build ranking models on all datasets. For each dataset, the samples are divided into ten equal parts, and one of them is selected as the testing set in turn, and the remaining part is used as the training set to train a ranking model.

All experiments are performed on a computer with Intel Core i7-8700 CPUS, MATLAB language, and Windows 10 platform chosen for building the experimental environment.

The average of training FPA and the average of testing FPA are recorded in Table.2 and Table.3. From Table.2, one can see that the proposed multi-objective ranking methods obtain the best performance of training FPA on 4 out of 11 data sets, and PSO obtains the best performance of training FPA on the remaining data sets. In the Table.3, it shows that the proposed methods provide the best testing FPA to ranking models on 7 out of 11 data sets, and NABC receives the best performance of testing FPA on the remaining testing data sets.

Table 2. The performance of FPA using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA in training data sets

	Single-objective methods				Proposed multi-objective methods	
	CoDE	NABC	PSO	BSO	NSGA-II	MOMSFFA
PC1	0.5433	0.5422	0.5452	0.5443	0.5457	0.5413
MC1	0.8967	0.8956	0.9001	0.8971	0.8993	0.8945
MW1	0.5141	0.5133	0.5156	0.5141	0.5157	0.5127
JM1	0.7414	0.7378	0.7600	0.7426	0.7604	0.7169
CM1	0.7334	0.7316	0.7355	0.7336	0.7358	0.7321
KC1	0.7981	0.7986	0.8015	0.7980	0.7979	0.7927
jdt	0.8150	0.8200	0.8251	0.8210	0.7971	0.7885
pde	0.7590	0.7668	0.7787	0.7692	0.7575	0.7430
luce	0.8392	0.8494	0.8618	0.8468	0.8198	0.8110
equ	0.7799	0.7832	0.7911	0.7797	0.7880	0.7807
mylyn	0.7499	0.7744	0.7951	0.7581	0.6701	0.6021

Interestingly, both the best training FPA and the best testing FPA are obtained by the proposed multi-objective ranking methods on six software programs selected from the NASA repository. In other words, there may be a complex problem in these programs that multi-objective ranking methods can improve the performance of FPA and reduce the redundant features to enhance the efficiency of ranking models. Based on five programs selected from the AEEEM repository, the single-objective ranking methods are superior to the proposed multi-objective ranking methods. It is to say that reducing the number of features is not beneficial to improve the performance of FPA on AEEEM data sets.

Table 3. The performance of FPA using MOMSFFA, CoDE, NABC, PSO, BSO, and NSGA-II in testing data sets

	Single-objective methods				Proposed multi-objective methods	
	CoDE	NABC	PSO	BSO	NSGA-II	MOMSFFA
PC1	0.5439	0.5455	0.4875	0.5461	0.5479	0.5529
MC1	0.8914	0.8892	0.5789	0.8882	0.8951	0.8832
MW1	0.5265	0.5247	0.4970	0.5253	0.5280	0.5310
JM1	0.7213	0.7194	0.5972	0.7067	0.7401	0.7034
CM1	0.7335	0.7316	0.6448	0.7324	0.7411	0.7344
KC1	0.7931	0.7930	0.6291	0.7905	0.7986	0.7964
jdk	0.7992	0.8082	0.7537	0.7973	0.7771	0.7485
pde	0.7316	0.7474	0.7121	0.7438	0.7447	0.7284
luce	0.7982	0.8098	0.7662	0.7831	0.7802	0.7750
equ	0.7668	0.7652	0.6335	0.7682	0.7863	0.7752
mylyn	0.7460	0.7567	0.6874	0.6905	0.6674	0.6131

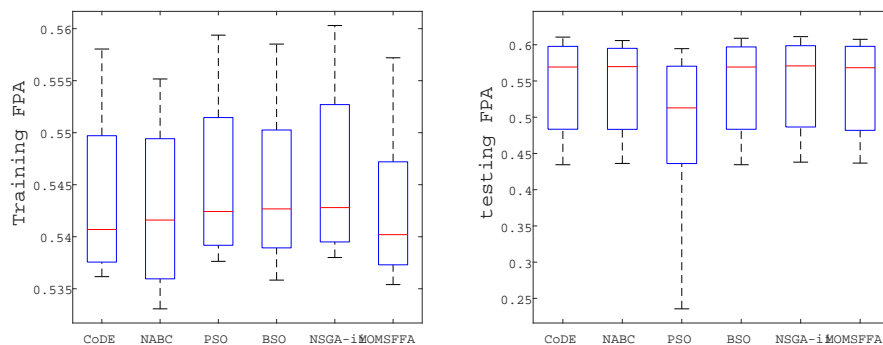
Additionally, the Friedman test is used in the significance test and employed in the rank-sum test. In the proposed study, it is used to obtain the rank order of the competitors' performance in SDP ranking tasks. Based on the numerical results of training FPA and testing FPA, the average rankings of the competitors are shown in Table.4. One can see that the proposed NSGA-II ranking method obtains the best performance to build ranking models for SDP. Although the proposed MOMSFFA ranking method is inferior to all single-objective ranking methods, the ranking models using MOMSFFA are more stable than those with single-objective ranking methods.

Table 4. Average Rankings of the algorithms based on the experimental results of FPA using Friedman test

Algorithm	Ranking
CoDE	3.4318
NABC	3.3636
PSO	3.5455
BSO	3.3864
NSGA-II	2.5909
MOMSFFA	4.6818

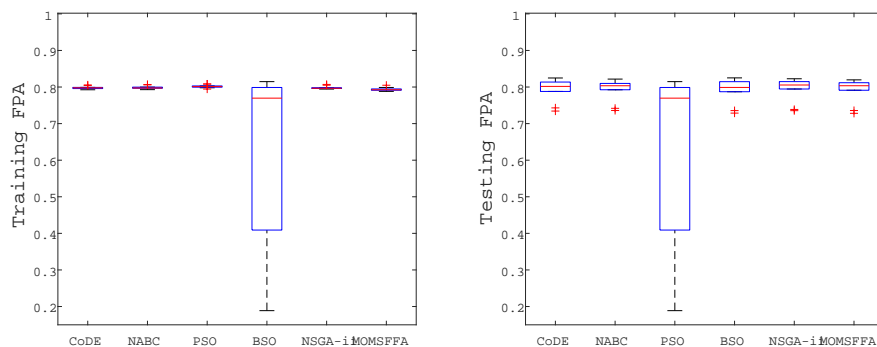
To analyze the performance of these competitors in-depth, an intuitive comparison of competitors in terms of FPA results can be seen that boxplots of the training FPA and the testing FPA based on each program are shown in Fig.3– 13.

To comprehensively investigate the performance of the proposed multi-objective ranking methods, the solutions of reducing features based on all data sets are recorded in Table.5. One can see that the proposed multi-objective ranking methods reduce features on all data sets. Thereby, the efficiency of building the ranking model is improved. Considering the performance of FPA, it can be concluded that the proposed multi-objective ranking method can address the complex problem of most of the programs selected from the NASA repository. The complex problem may not exist in the programs selected from the AEEEM repository that the single objective ranking methods are superior to the proposed approaches to improve the performance FPA of ranking models. However, sometimes



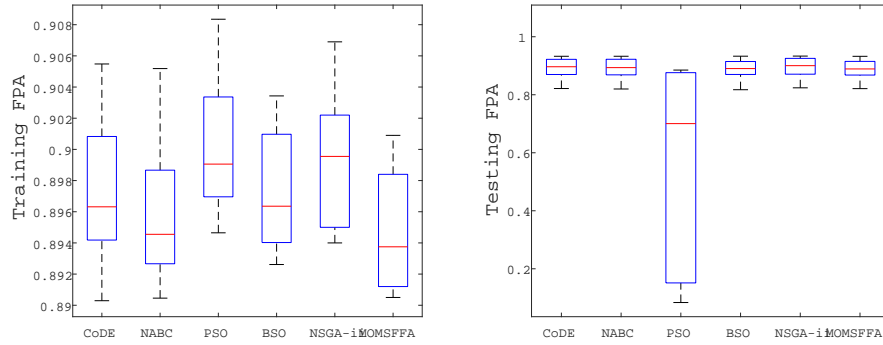
All methods achieve similar FPA performance. Using NSGA-II is slightly better than other methods in the training set and employing MOMSFFA slightly better than the other approaches in the testing set. The ranking model obtained by PSO shows performance degradation in testing set. In other words, compared with single-objective ranking methods, the proposed multi-objective ranking methods are beneficial to build a ranking model based on PC1.

Fig. 3. The performance FPA of ranking models based on program PC1 using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



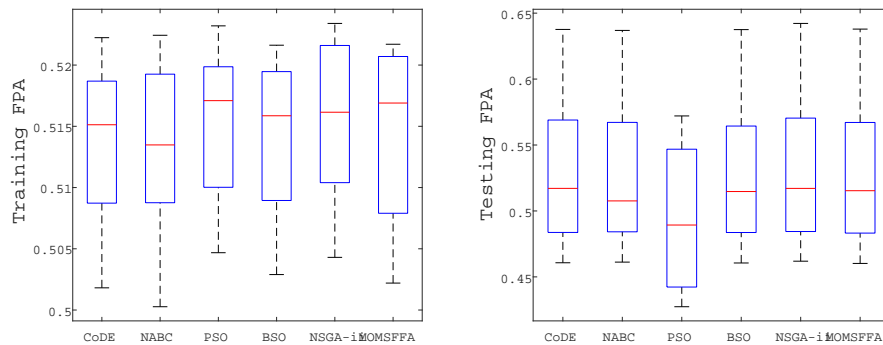
The proposed multi-objective ranking methods are slightly better than CoDE and NABC both in the training FPA and testing FPA, significantly better than BSO in training FPA, and superior to PSO in testing FPA. In other words, compared with single-objective ranking methods, the proposed multi-objective ranking methods are beneficial to build a ranking model based on KC1.

Fig. 4. The performance FPA of ranking models based on program KC1 using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



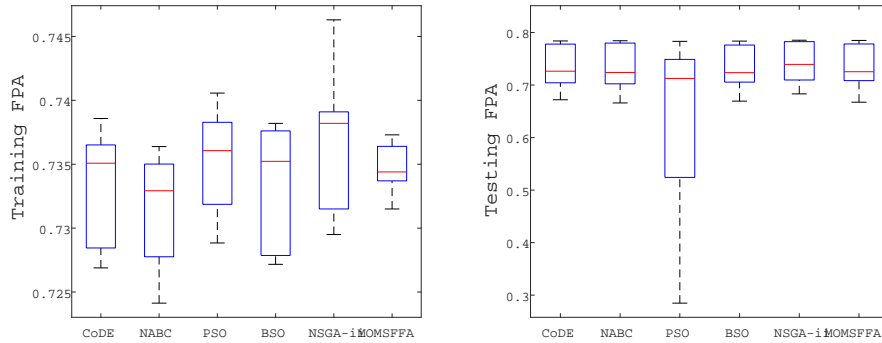
The proposed NSGA-II ranking method obtains the best performance both in the training FPA and testing FPA. Although the performance of the proposed MOMSFFA ranking method is inferior to that of the single objective ranking methods in training FPA, it is similar to these in the testing FPA. In other words, compared with single-objective ranking methods, the proposed ranking method using NSGA-II is beneficial to build a ranking model based on MC1.

Fig. 5. The performance FPA of ranking models based on program MC1 using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



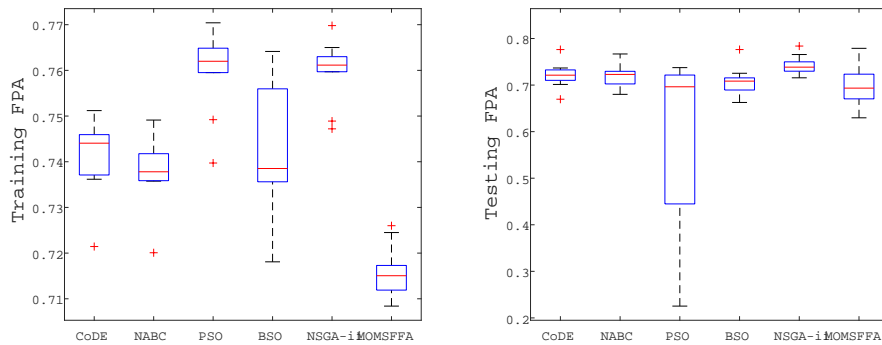
The proposed multi-objective ranking methods are superior to CoDE, NABC, and BSO both in the training FPA and testing FPA. Although the performance of the proposed MOMSFFA ranking method is similar to that of PSO in training FPA, it is significantly better than PSO in testing FPA. In other words, compared with single-objective ranking methods, the proposed MOMSFFA ranking method is beneficial to build a ranking model based on MW1.

Fig. 6. The performance FPA of ranking models based on program MW1 using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



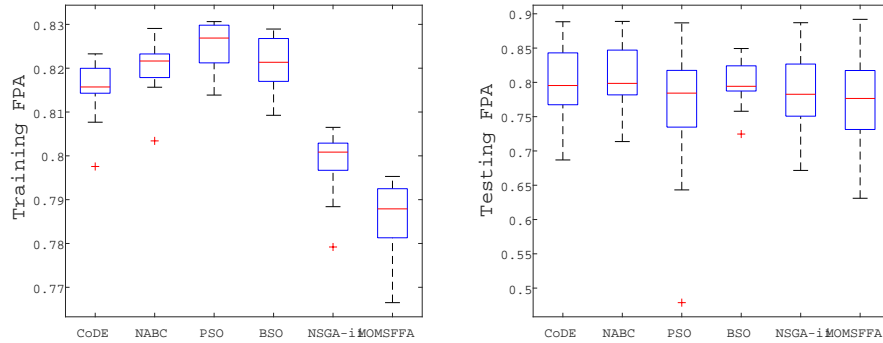
The proposed NSGA-II ranking method obtains the best performance in training FPA and in testing FPA. It can be concluded that compared with single-objective ranking methods, the proposed NSGA-II ranking method is beneficial to build a ranking model based on CM1.

Fig. 7. The performance FPA of ranking models based on program CM1 using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



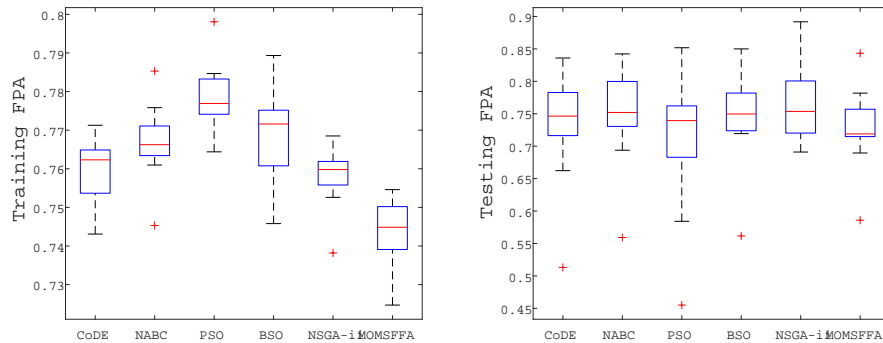
The proposed NSGA-II ranking method is superior to CoDE, NABC, and BSO both in the training FPA and testing FPA. Although the performance of the proposed NSGA-II ranking method is similar to that of PSO in training FPA, it is significantly better than PSO in the testing FPA. In other words, compared with single-objective ranking methods, the proposed NSGA-II ranking method is beneficial to build a ranking model based on JM1.

Fig. 8. The performance FPA of ranking models based on program JM1 using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



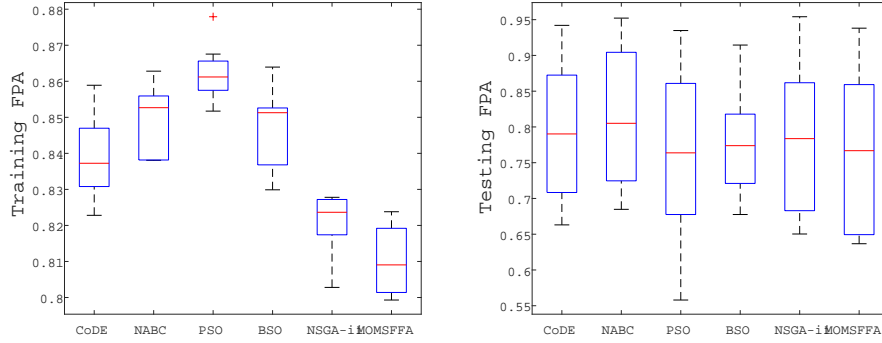
Where PSO obtains the best performance in the training FPA, NABC receives the best solutions in the testing FPA. Although the proposed multi-objective ranking methods are inferior to all single objective approaches in the training FPA and in the testing FPA, NSGA-II and MOMSFFA reduce the number of features from 15 to 4 and 2, respectively. In other words, where single objective methods can improve the ranking performance of FPA on JDT, multi-objective algorithms can enhance the efficiency of building ranking models on JDT.

Fig. 9. The performance FPA of ranking models based on program Eclipse JDT Core using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



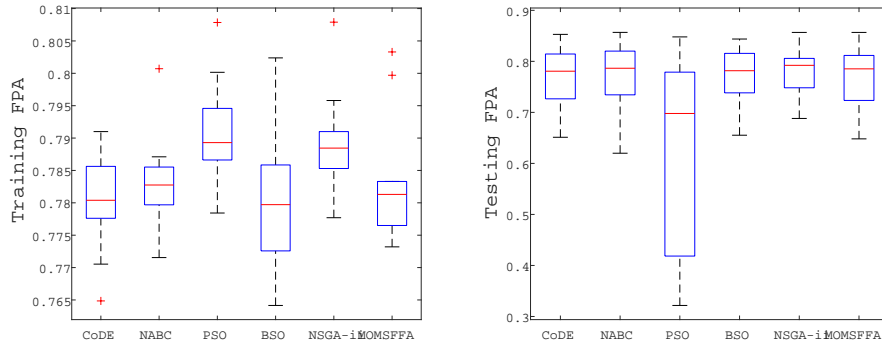
Where PSO obtains the best performance in training FPA, NSGA-II receives the best solutions in testing FPA. Considering the other objective that reducing the number of features, NSGA-II is the best choice to build a ranking model on PDE.

Fig. 10. The performance FPA of ranking models based on program Eclipse PDE UI using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



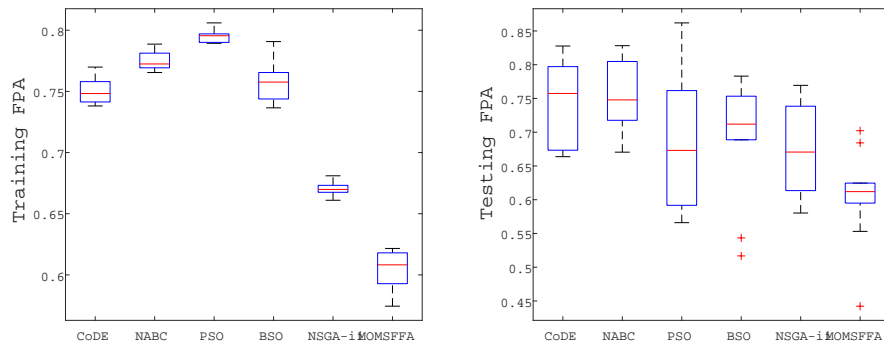
Where PSO obtains the best performance in training FPA, NABC receives the best solutions in testing FPA. The performance of the proposed multi-objective ranking methods is inferior to that of all single-objective approaches in training FPA but the performance of all methods is similar in testing FPA. In other words, the proposed multi-objective ranking methods are more stable than single-objective algorithms on Apache Lucene. Considering the other objective that mining the numbers of selected features, the proposed multi-objective ranking methods are beneficial to build ranking models on Apache Lucene.

Fig. 11. The performance FPA of ranking models based on program Apache Lucene using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



Where PSO obtains the best performance in training FPA but provides the worse performance in testing FPA, NSGA-II receives similar solutions to PSO in training FPA but obtains the best solutions in testing FPA. Considering the other objective that mining the numbers of selected features, the proposed NSGA-II ranking method is beneficial to build ranking models on Equinox framework.

Fig. 12. The performance FPA of ranking models based on program Equinox framework using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA



Where PSO obtains the best performance in training FPA, NABC receives the best solutions in testing FPA. All single-objective approaches are superior to the proposed multi-objective ranking methods in training FPA and in testing FPA. In other words, single objective methods are beneficial to build ranking models to improve the performance APF on Mylyn.

Fig. 13. The performance FPA of ranking models based on program Mylyn using CoDE, NABC, PSO, BSO, NSGA-II, and MOMSFFA

feature selection is more important than using metrics that the proposed multi-objective ranking methods can be used to enhance the efficiency of building the ranking models on these datasets.

Additionally, The time cost of all competitors to build the ranking model is saved in Table.6. One can see thoes single-objective ranking methods are superior to multi-objective ranking methods on 8 programs. The proposed MOMSFFA ranking method is more efficient on KC1, jdt, and pde. Considering the similar solutions of all competitors to the complex problem of these 3 programs, MOMSFFA is the best choice to build ranking models.

From the above experimental results, the advantage of each optimization algorithm should be analyzed in the process of building ranking models. One can see that six styles of evolutionary algorithms are used to optimize the ranking performance based on 11 software programs. Although PSO performs advantages of global search ability and convergence to obtain better solutions to the training FPA on most of the AEEEM data sets, it receives the worst ranking performance of testing FPA. It is to say that it is not a good choice to use PSO to build ranking models based on these data sets. Considering the performance of ranking models obtained by CoDE, NABC, and BSO, where NABC has advantages of the global search ability to CoDE and BSO on AEEEM data sets, BSO performs the best global search ability than CoDE and NABC on NASA data sets, NABC obtains the best solutions on AEEEM data sets, and CoDE is superior to NABC and BSO on convergence for most of data sets. Compared with the above single-objective ranking methods, where NSGA-II has the advantage of global search ability on most NASA data sets, MOMSFFA not only searches stable solutions to build ranking models for PC1 and MW1 but also converges fast for KC1, jdt, and pde. It can be concluded that where CoDE is beneficial to reduce the time cost of building ranking models based on most

Table 5. The experimental results of selected features used to build ranking models on all data sets

	Single-objective methods				Proposed multi-objective methods		
	CoDE	NABC	PSO	BSO	NSGA-II	MOMSFFA	
PC1	40	40	40	40	6		13
MC1	38	38	38	38	5		14
MW1	39	39	39	39	5		13
JM1	21	21	21	21	2		5
CM1	37	37	37	37	5		12
KC1	21	21	21	21	2		5
jdt	15	15	15	15	1		5
pde	15	15	15	15	2		4
luce	15	15	15	15	2		3
equ	15	15	15	15	1		4
mylyn	15	15	15	15	2		3

Table 6. The time cost using MOMSFFA, CoDE, NABC, PSO, BSO, and NSGA-II

	MOMSFFA	CoDE	NABC	PSO	BSO	NSGA-II
PC1	37.8	36.2	39.7	37.4	36.8	37.9
MC1	2135	1671	2260	1968	1840	1842
MW1	11.7	7.1	8.6	8.2	9.8	7.5
JM1	2640	2164	2796	2605	2565	2407
CM1	6.1	5.5	5.6	5.3	5.8	5.7
KC1	100	108	111	112	107	110
jdt	27.9	29.3	30.7	29.2	29.8	34
pde	54.5	58.1	61.5	58.3	58.5	62.6
luce	16.6	15.5	15.9	15	15.8	16.7
equ	6.1	5.6	5.4	5.1	5.9	5.5
mylyn	86.9	79.8	83.5	77.8	78.8	87.2

datasets, NABC is the best choice to build ranking models on AEEEM datasets, the proposed multi-objective algorithms obtain the best performance of global search ability on most of NASA data sets. The most important advantage of the proposed multi-objective ranking method is that it enhances the efficiency of ranking models by reducing redundant features. Thereby, the proposed multi-objective ranking methods can address the complex problem in NASA data sets.

The performance of an SDP model mainly depends on the quality of the datasets. In other words, where single-objective optimizers can obtain one goal in the SDP tasks, multi-objective optimizers can archive multi-goals in the SDP tasks. However, if the datasets have only one problem to solve, the performance of all the optimizers seems the same. It is to say that the more problems in the dataset, the greater the performance difference between the single-objective optimizer and the multi-objective optimizer.

5. Threats to Validity

In this section, we discuss three validity threats to the experimental results of our work. One threat to validity is that the proposed method may not obtain promising results on other datasets. The metrics have the most impact on the ranking task for different datasets. In other words, the metrics cause different effects on different datasets for SDP problems. Another threat to validity is that all the compared methods are single-objective evolutionary algorithms, which optimize a single goal on datasets for the ranking task. The performance of those methods depends on the quality of the datasets. Generally speaking, single-objective evolutionary algorithms obtain good performance on datasets with a single problem and worse results on datasets with complex problems. It is the reason that the proposed methods perform better than other competitors on NASA datasets and perform worse than other competitors on AEEEM datasets. The last threat to validity is that we use Friedman's rank test to statistically analyze the six competitors and use FPA as the evaluation measure. Our work can also use the Wilcoxon rank test and mean square error (MSE).

6. Conclusion

In this study, two multi-objective ranking methods are proposed to address a complex problem which is consisted of insufficient samples, curse-of-dimensionality, and class imbalance. Compared with four single objective algorithms based on 11 software programs, although learning-to-rank methods improve the ranking performance for SDP, curse-of-dimensionality and class imbalance are ignored in building ranking models for software defect prediction. Where the proposed multi-objective ranking methods are used to build ranking models, the ranking performance of these ranking models is enhanced by selecting the related feature subsets. In other words, the proposed multi-objective ranking methods can address the complex problem of SDP.

The main reason for the outperformance of the proposed methods on NASA datasets should be discussed. Where NASA uses 40 metrics to collect the information for programs, AEEEM employs only 15 metrics to dig the information for its software system. In other words, as the number of metrics increases, the relationship between them and

the performance of prediction methods becomes more complex. From our experimental results, feature selection methods always enhance the performance of the SDP tasks, especially on datasets using a large number of metrics. However, the feature selection methods have little impact on the SDP tasks of AEEEM datasets. One can see that the proposed multi-objective ranking methods outperform single-objective ranking methods in addressing complex problems in datasets. It can be concluded that the optimized goal of reducing redundant and irrelevant features is effective for SDP tasks on datasets with a large number of metrics, which is missed in ranking tasks using single-objective optimization algorithms. In addition, NSGA-II and MSFFA win the best performance compared with other single-objective optimization algorithms for most SDP ranking tasks. It is to say that NSGA-II and MSFFA obtain a good balance between exploration and exploitation to solve complex problems in SDP ranking tasks.

In the future, the performance of the proposed methods will be verified by more publicly available datasets containing more features.

Acknowledgments. The author would like to thank those who gave help and guidance to this work. This work was funded by two projects, the National Natural Science Foundation of China (62266024) and the Science and Technology Foundation of Jiangxi Province (20202BABL202019).

References

1. Balogun, A.O., Basri, S., Jadid, S.A., Mahamad, S., Al-momani, M.A., Bajeh, A.O., Alazzawi, A.K.: Search-based wrapper feature selection methods in software defect prediction: an empirical analysis. In: Computer Science On-line Conference. pp. 492–503. Springer (2020)
2. Balogun, A.O., Basri, S., Abdulkadir, S.J., Hashim, A.S.: Performance analysis of feature selection methods in software defect prediction: a search method approach. Applied Sciences 9(13), 2764 (2019)
3. Bell, R.M., Ostrand, T.J., Weyuker, E.J.: The limited impact of individual developer data on software defect prediction. Empirical Software Engineering 18(3), 478–505 (2013)
4. Buchari, M., Mardiyanto, S., Hendradjaya, B.: Implementation of chaotic gaussian particle swarm optimization for optimize learning-to-rank software defect prediction model construction. In: Journal of Physics: Conference Series. vol. 978, p. 012079. IOP Publishing (2018)
5. Cao, L., Ben, K., Peng, H.: Enhancing firefly algorithm with multiple swarm strategy. Journal of Intelligent & Fuzzy Systems 41(1), 99–112 (2021)
6. Chen, L., Fang, B., Shang, Z., Tang, Y.: Tackling class overlap and imbalance problems in software defect prediction. Software Quality Journal 26(1), 97–125 (2018)
7. Choudhary, G.R., Kumar, S., Kumar, K., Mishra, A., Catal, C.: Empirical analysis of change metrics for software fault prediction. Computers & Electrical Engineering 67, 15–24 (2018)
8. Cowlessur, S.K., Pattnaik, S., Pattanayak, B.K.: A review of machine learning techniques for software quality prediction. Advanced Computing and Intelligent Engineering pp. 537–549 (2020)
9. D’Ambros, M., Lanza, M., Robbes, R.: An extensive comparison of bug prediction approaches. In: 2010 7th IEEE working conference on mining software repositories (MSR 2010). pp. 31–41. IEEE (2010)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 6(2), 182–197 (2002)

11. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. pp. 39–43. Ieee (1995)
12. Fawcett, T.: An introduction to roc analysis. *Pattern recognition letters* 27(8), 861–874 (2006)
13. Hancer, E., Xue, B., Zhang, M., Karaboga, D., Akay, B.: Pareto front feature selection based on artificial bee colony optimization. *Information Sciences* 422, 462–479 (2018)
14. Harman, M.: The relationship between search based software engineering and predictive modeling. In: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. pp. 1–13 (2010)
15. Li, J., Song, L., Cao, L.: An improved firefly algorithm with distance guided selection strategy and its application. *Journal of Intelligent & Fuzzy Systems* 43(1), 889–906 (2022)
16. Li, X., Yang, X., Su, J., Wen, W.: A multi-objective learning method for building sparse defect prediction models. In: 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS). pp. 204–211. IEEE (2020)
17. Mauša, G., Galinac-Grbac, T., Dalbelo-Bašić, B.: A systematic data collection procedure for software defect prediction. *Computer Science and Information Systems* 13(1), 173–197 (2016)
18. Moser, R., Pedrycz, W., Succi, G.: A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. In: Proceedings of the 30th international conference on Software engineering. pp. 181–190 (2008)
19. Nguyen, B.H., Xue, B., Zhang, M.: A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation* 54, 100663 (2020)
20. Peng, H., Deng, C., Wu, Z.: Best neighbor-guided artificial bee colony algorithm for continuous optimization problems. *Soft computing* 23(18), 8723–8740 (2019)
21. Rostami, M., Berahmand, K., Nasiri, E., Forouzandeh, S.: Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence* 100, 104210 (2021)
22. Shepperd, M., Song, Q., Sun, Z., Mair, C.: Data quality: Some comments on the nasa software defect datasets. *IEEE Transactions on Software Engineering* 39(9), 1208–1215 (2013)
23. Shi, Y.: Brain storm optimization algorithm. In: International conference in swarm intelligence. pp. 303–309. Springer (2011)
24. Song, X.F., Zhang, Y., Guo, Y.N., Sun, X.Y., Wang, Y.L.: Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. *IEEE Transactions on Evolutionary Computation* 24(5), 882–895 (2020)
25. Tran, B., Xue, B., Zhang, M.: Variable-length particle swarm optimization for feature selection on high-dimensional classification. *IEEE Transactions on Evolutionary Computation* 23(3), 473–487 (2018)
26. Turabieh, H., Mafarja, M., Li, X.: Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. *Expert systems with applications* 122, 27–42 (2019)
27. Wang, S., Liu, T., Nam, J., Tan, L.: Deep semantic feature learning for software defect prediction. *IEEE Transactions on Software Engineering* 46(12), 1267–1293 (2018)
28. Weyuker, E.J., Ostrand, T.J., Bell, R.M.: Comparing the effectiveness of several modeling methods for fault prediction. *Empirical Software Engineering* 15(3), 277–295 (2010)
29. Yang, X., Tang, K., Yao, X.: A learning-to-rank algorithm for constructing defect prediction models. In: International Conference on Intelligent Data Engineering and Automated Learning. pp. 167–175. Springer (2012)
30. Yang, X., Tang, K., Yao, X.: A learning-to-rank approach to software defect prediction. *IEEE Transactions on Reliability* 64(1), 234–246 (2014)
31. Yang, X.S.: Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. pp. 169–178. Springer (2009)

32. Yu, X., Bennin, K.E., Liu, J., Keung, J.W., Yin, X., Xu, Z.: An empirical study of learning to rank techniques for effort-aware defect prediction. In: 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 298–309. IEEE (2019)
33. Yu, X., Liu, J., Yang, Z., Jia, X., Ling, Q., Ye, S.: Learning from imbalanced data for predicting the number of software defects. In: 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE). pp. 78–89 (2017)

Yiji Chen earned the Ph.D. degree in the WonKwang University, Korea. He is a lecturer at Jiujiang University, China. His main research interests include software testing, big data marketing, and artificial intelligence.

Lianglin Cao earned the Ph.D. degree in the Naval University of Engineering, Wuhan, China. He is a lecturer at Jiujiang University, China. His main research interests include swarm intelligence algorithm, software quality assurance.

Li Song is a lecturer at Jiujiang University, China. His main research interests include swarm intelligence algorithm and artificial intelligence.

Received: August 30, 2022; Accepted: May 04, 2023.

Multi-perspective Approach for Curating and Exploring the History of Climate Change in Latin America within Digital Newspapers

Genoveva Vargas-Solar¹, José-Luis Zechinelli-Martini², Javier A. Espinosa-Oviedo^{1,3},
and Luis M. Vilches-Blázquez⁴

¹ CNRS, Univ Lyon, INSA Lyon, UCBL, LIRIS, UMR5205, F-69622 Villeurbanne, France
genoveva.vargas-solar@cnrs.fr

² Fundación Universidad de las Américas Puebla, 72820 San Andrés Cholula, Mexico
jose-luis.zechinelli@udlap.mx

³ CPE, Univ Lyon, 69616 Villeurbanne, France
javier.espinosa@cpe.fr

⁴ Universidad Politécnica de Madrid, 28660 Boadilla del Monte, Spain
luis.vilches@upm.es

Abstract. This paper introduces a multi-perspective approach to deal with curation and exploration issues in historical newspapers. It has been implemented in the platform LACLICHEV (Latin American Climate Change Evolution platform).

Exploring the history of climate change through digitalized newspapers published around two centuries ago introduces four challenges: (1) curating content for tracking entries describing meteorological events; (2) processing (digging into) colloquial language (and its geographic variations⁵) for extracting meteorological events; (3) analyzing newspapers to discover meteorological patterns possibly associated with climate change; (4) designing tools for exploring the extracted content.

LACLICHEV provides tools for curating, exploring, and analyzing historical newspaper articles, their description and location, and the vocabularies used for referring to meteorological events. This platform makes it possible to understand and identify possible patterns and models that can build an empirical and social view of the history of climate change in the Latin American region.

Keywords: data curation, metadata extraction, data collections exploration, data analytics.

1. Introduction

Ninety-seven per cent of climate scientists agree that climate-warming trends over the past century are very likely due to human activities⁶. Some observation reports and studies reveal that the planet's average surface temperature has risen about 2.0 degrees Fahrenheit (1.1 degrees Celsius) since the late 19th century. The hypothesis is that this change has been mainly driven by increased carbon dioxide and other human-made atmospheric emissions.

⁵ In Iberoamerica, Spanish has variations in the different countries, even if all Spanish-speaking people can perfectly understand each other.

⁶ <https://climate.nasa.gov/scientific-consensus/>

Technological advances have allowed understanding of phenomena and complex systems by collecting many different types of information. Data collections are exported under different releases with different sizes and formats (e.g., CSV, text, excel), sometimes with various quality features. Tools helping to understand, consolidate and correlate data collections are crucial. Even if there is an increasing interest in analysing digital data collections for performing historical studies on climatologic events, the history of climate behaviour is still an open issue that has not revealed missing knowledge. Long historical data studies could make it possible to compute more complete models of climatic phenomena and the conditions in which they emerged. However, meteorology is a young science that started around the 19th century. It is supported by more or less recent data, making it challenging to run an analysis that can give more historical pictures of climatic evolution and its implications using observations instead of extrapolations. Those willing to promote changes in the behaviour of society and industry to reduce emissions that have a role in climate change must convince civil society of the importance of the challenges. For this reason, our work addressed the problem of collecting and analyzing the history of meteorological events to explore how they were described, lived and perceived by civil society. In this sense, the digitalization of data collections has an increasingly vital role in collecting vast amounts of *hidden* data. Thus, considering that digital archives become more easily accessible every time and contain explicit and implicit spatio-temporal information, researchers in GIScience [18], are becoming aware of these new data sources [10], [9], [34], [41]. Moreover, digital data collections make it possible to have an analytic vision of the evolution of environmental, administrative, economic and social phenomena. In this context, our work deals with data collections that report the emergence of meteorological events (e.g., temperature changes, avalanches, river flow growth, or volcano eruptions). However, the digitized collections have some implicit issues. They are often riddled with Optical Character Recognition (OCR) errors that hamper the performance of information retrieval systems. Therefore, handling OCR errors is one of the two significant problems for information retrieval from collections of historical documents. On the other hand, these sources' problems are related to historical language changes since digitized texts are written in the language of their origin.

This paper proposes an extended description of the Latin American Climate Change Evolution platform called LACLICHEV [37]. The objective of LACLICHEV is to provide an integrated platform to expose and study meteorological events described in historical newspapers that are possibly related to the history of climate change in Latin America. In this sense, we hypothesize that the history (in Latin America) is contained in newspaper articles in digital collections available in national libraries of four countries, namely Mexico, Colombia, Ecuador, and Uruguay. Considering this starting point, LACLICHEV addresses the following issues (see Figure 1):

- i First, newspaper archaeology, by chasing articles about climatological events using specific vocabulary to discover as many articles as possible (see the left side of the Figure 1). The challenge is choosing adequate vocabulary to increase the chances of getting articles about climatologic events.
- ii Second, once an article talks about a climatologic event, it is tagged with Geo-Temporal metadata specifying what happened, where and when it happened, its duration and geographical extent (see the centre of Figure 1). The objective is to build a climatologic event history of empirical observations.

- iii Finally, on top of this history, the objective is to run analytics questions and visualize results in maps given that the content is highly spatial (see the right side of the Figure 1).

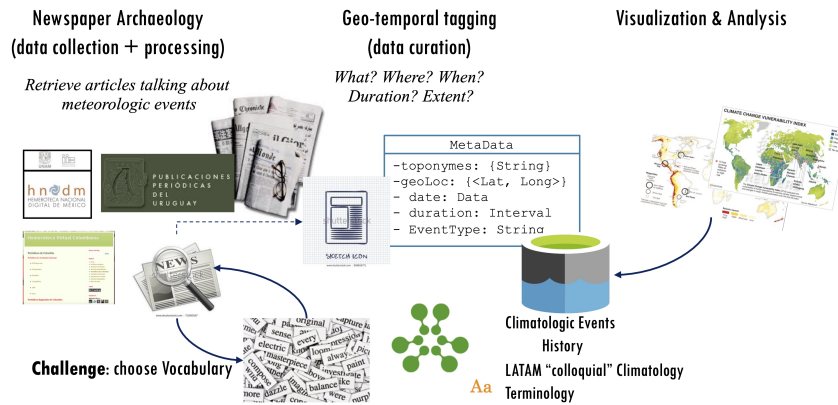


Fig. 1. Problem Statement

The main contribution of our work is LACLICHEV. It is a data collections exploration platform that applies data collections curation and exploration techniques. These techniques are combined with data retrieval, data analytics, and visualization for understanding the content of articles that report historical meteorological events. These data with high geospatial and temporal content can be aggregated into maps. Maps give a one-shot view of the history of meteorological events observed from the empirical perspective of civil society before the emergence of meteorology as a science [39, 6].

The second contribution is a meteorological event knowledge model that provides several perspectives to describe an event. Perspectives organize metadata that represent such an event is reported through empirical narratives that can appear in newspaper articles, as in the context of our work.

The third contribution is the experimental use of LACLICHEV to build the history of climate change in Latin America from digital newspapers. To track meteorological events, we explored newspapers to search articles that could report such events, the conditions in which they happened, their duration, the places in which they occurred, and their impact in terms of an approximate number of casualties and the kind of damages, etc. As an experimental scenario, we chose the XVIII and XIX centuries, which define a golden age for newspapers in Latin American countries [13], namely, Mexico, Colombia, Ecuador, and Uruguay.

The remainder of this paper is organized as follows. Section 2 introduces the general architecture of LACLICHEV and its functions implemented by its main modules. Section 3 describes the knowledge model we propose for modelling meteorological events as described in empirical narratives written in natural language. Section 4 describes the general

curation and exploration processes implemented by LACLICHEV to deal with the curation and exploration of historical newspaper articles potentially reporting on climatologic events. It also describes the use cases that we conducted to evaluate it. Section 5 studies approaches that promote datasets exploration for defining the type of analysis possible on top of them. Finally, Section 6 concludes the paper underlying the contribution and discusses future work.

2. LACLICHEV for Curating and Exploring Historical Newspapers Articles

Figure 2 shows the general architecture of LACLICHEV organised into three layers:

- i frontend with an interface providing functions for curating articles and creating events descriptions; and giving access to explore the event history containing curated articles reporting meteorological events;
- ii backend with the meteorological event history stored in a document management system (see number 1 in Figure 2) and modules for curating (pre-processing and tagging the textual content of newspaper articles - number 2 in Figure 2) and exploring events (see number 3 in Figure 2);
- iii external layer connecting to document providers that are available through servers accessible on the Web and APIs exported, for example, by libraries.

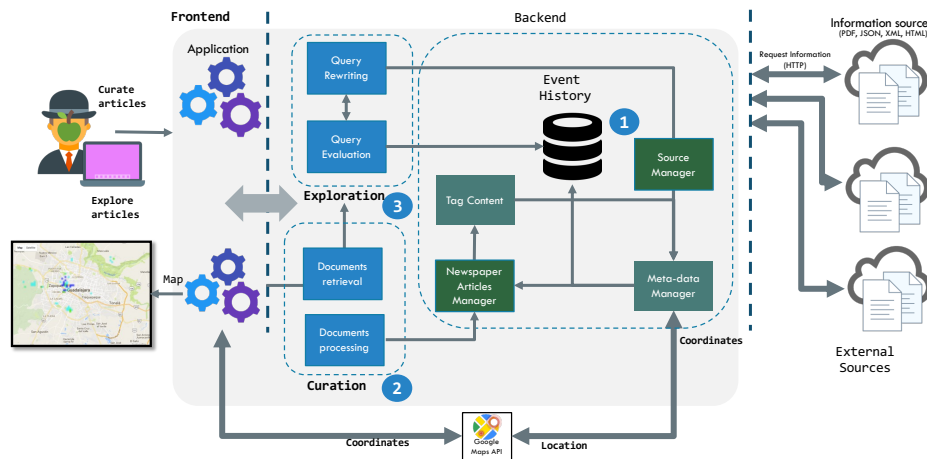


Fig. 2. General functional architecture of LACLICHEV

In the following sections, we describe the core layer of LACLICHEV, namely the backend with its main components, the meteorological event's history, and associated curation modules used to feed the history and exploration modules to process queries to explore this history.

2.1. Meteorological Events History

The event history (storing metadata describing an event from several perspectives, see number 1 in Figure 2) is based on an event knowledge model that we proposed and that is described in Section 3. Through this knowledge model, it is possible to represent the empirical description of a meteorological event with metadata from several perspectives: descriptive (the vocabulary used for describing a meteorological event and the statistics of its use); linguistic (the structure of the sentences used in a narrative describing a meteorological event); the meteorological perspective (represents factual data about an event, location, duration, type, intensity, etc.); and the domain knowledge perspective (meta-data about empirical and factual observations provided by meteorology experts, e.g., the fact that strong rainfall can correspond to more the 75 mm/hr rain).

Metadata is stored in persistence support, a key-value or a document store, depending on the technology adopted by each library. In contrast, the raw documents remain archived in a different server or the same store. LACLICHEV uses a document store (i.e., MongoDB⁷) for storing geo-temporally tagged meteorological events. These events' history provides an interface for performing querying and analytics tasks on top of it. The digital collection can be initially queried by filtering the documents by region, country, or year. Digital libraries offer front-ends for performing this classic information retrieval process. For example, select newspapers published in Uruguay (i.e., geographic filter) between 1800-1810 (i.e., temporal filter). It can also perform analytics queries. For example, locate events during the XIX century, enumerate and locate the most famous meteorological events in the region, and create a heat map of the events in Latin America that happened in the last ten years of the XIX century.

2.2. Curating Newspapers Content Modules

The backend of LACLICHEV includes of a set of modules devoted to implement different operations of data curation (see number 2 in Figure 2). The objective of curating (historical) newspaper articles is to build a meteorological events history that newspaper articles reporting events with metadata, providing as much information as possible about the reported event.

Figure 3 shows the newspapers curation process that is a semi-automatic process devoted to:

- find articles reporting this type of events within digital collections available in existing digital libraries repositories;
- geo-tag interactively and store those articles that actually report such events for building a meteorological event database.

LACLICHEV relies on a knowledge graph that integrates a thesaurus classifying meteorological event types, Wordnet and a glossary defining meteorologic characteristics of meteorological events.

Curation process. Curation tasks can be recurrent and include a human-in-the-loop strategy for validating and adjusting results. For example, suppose an event is geo-tagged to associate it with a geographic location, and the event is described in an article about

⁷ This is a recurrent storage strategy when building databases as a result of processing textual content [32].

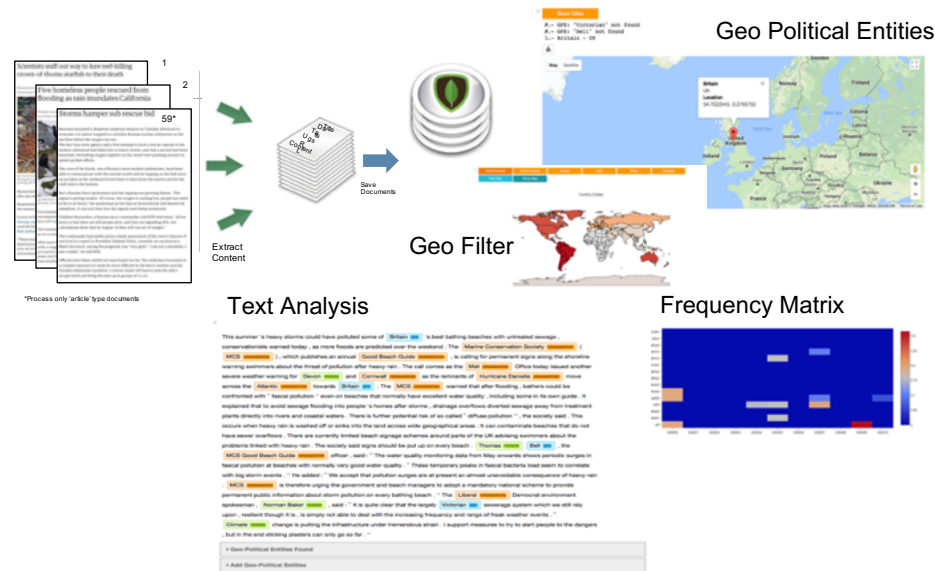


Fig. 3. Newspapers curation process

Montevideo news from Uruguayan newspaper collections. In that case, a human will verify that the geographic location refers to Montevideo in Uruguay and not Minnesota (United States).

During this phase, articles referring to meteorological events are geo-temporally tagged to associate them with the region and/or time window in which they happened. The data analyst validates tags. Since the result can contain a significant number of articles, the user can use three tools to understand the content of the result. The tools let her/him manipulate a terms frequency matrix and heat map.

She/he can also explore the content of the article text using a view that provides information about the context in which the terms are potentially describing an event that appears in the document. For example, the name of geographic locations in the document might refer to the event’s location and the region it touched, and a list of geopolitical entities (e.g., school, public building, etc.) to determine the damages caused by the event.

Curation functions provided by the backend modules. The data analyst can perform the following curation actions:

- Correct the terms associated with meteorological events that might not be used in such a sense in the text. Indeed, some social and political demonstrations are often described as meteorological events. For a classic automatic text analysis process, this cannot be easy to identify and filter. For example, an article entitled “*Stormy weather within the ails of the senate in Ecuador*” has nothing to do with the types of events considered but a political one.
- Determine whether personal names correspond to the event’s name (e.g. hurricane or storm’s name). If that is the case, this information will be used to insert the event into the history.
- Verify whether the names of cities, regions, and countries correspond to

geographic entities. The system underlines the names of patronyms, and the data analyst can see the location of the possible geographic entities. Thus, the user can also confirm whether the article refers to the geographic place that she/he is searching for. For instance, if “Santa Clara” is underlined, it can refer to a point of interest, city, or village.

- Determine the date of the event and its characteristics. The temporal terms and adjectives are also underlined to let the data analyst click on those that describe the event.
- Determine the type of damages caused by the event by exploring those terms that describe such information.

The previous actions are used to complete the representation of the articles’ content (extracted dynamically) and identify meteorological events more accurately since the data analyst, or domain expert knowledge is used (see Figure 4 showing LACLICHEV interfaces for curation). Note that one event can be described by several articles. In that case, the information stemming from the different sources is loosely integrated by performing the union of the content by applying some rules. For example, suppose the dates reported in two articles do not entirely correspond (variation of the day or the hour). In that case, the date of the event is modelled as an interval computed by processing the dates. If the dates are too disparate, the system keeps a set of dates. A similar process is done with locations; in this case, the system defines a region. A user can define a threshold of the size of a region associated with an event according to its type. Otherwise, the system keeps a set of geographical points.

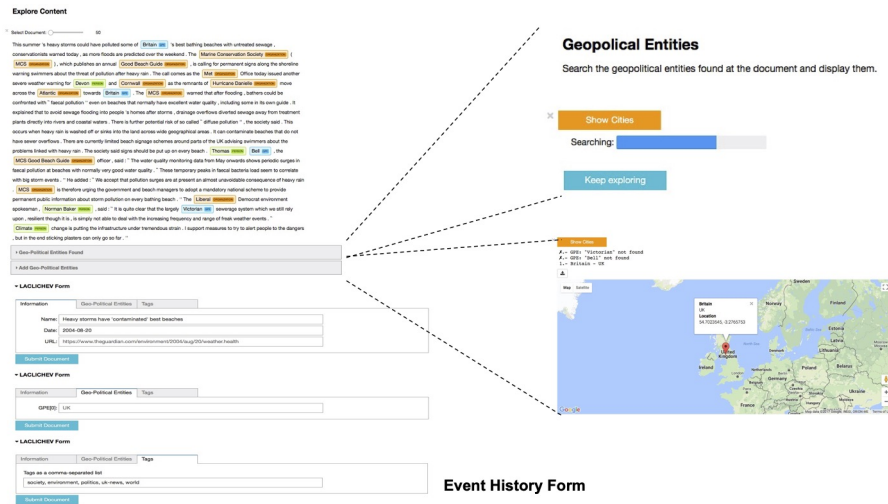


Fig. 4. Event curation process interface for tagging events

2.3. Exploring the Collections of Digital Newspapers

Newspaper articles are explored by conjunctive or disjunctive keyword queries, where keywords can belong to several vocabularies (see number 3 in Figure 2). For example,

search articles reporting heavy storms and rivers flooding. The query expressed by a data analyst is automatically completed by using rewriting techniques that consider synonyms, more specific or more general concepts [11]. Thus, three tools can be used for exploring meteorological events depending on expert knowledge of what she/he is looking for.

The rewriting process produces several proposals that the data analyst can adjust and then choose to be evaluated (see details in Section 4). Each chosen query is evaluated using information retrieval techniques, including the article's text stemming for extracting the terms and constructing a frequency matrix that provides occurrence statistics of the representative terms of the text content within a collection of documents.

In general, information retrieval processes do not exhibit this matrix; it is an internal data structure representing the content of the documents and is used to answer queries. In our approach, this frequency matrix is accessible to the data scientist because it provides an aggregated view of the content of a document collection. Additionally, we compute and exhibit a terms heatmap for a given documents collection to provide a more economical (i.e., consolidated) view of the collection's content. Our approach provides an interactive interface that lets data scientists manipulate these data structures to define the piece of collections she/he want to explore.

The data scientist can explore them and then decide whether the collection can describe meteorological events and the documents that might be closer to her requirements. She/he can decide eventually to explore some documents directly or reformulate the query. Once a result containing articles that potentially answer the query has been computed, the user can explore the result and validate the selection elements during the next step of the data exploration workflow.

- *Filtering*. Retrieving factual information, for example, filtering events by region, country or year. For example, Uruguay for the country and between 1800–1810 for the temporal filter.

- *Term frequency*. Understanding the content of digital newspaper collection through the vocabulary used in its articles. Therefore, LACLICHEV exposes the terms frequency matrix and a terms heatmap under an interactive interface. The domain expert can see which are, statistically, the terms most used in the articles, group documents according to the terms used, and choose articles using a specific term.

- *Additional information*. Exploring the content of a specific article using a view that provides information about the name of geographic locations in the document. These locations might refer to the event's location and the region it touched and a list of geospatial features (e.g., school, public building, etc.) to determine, for example, the damages caused by the event.

Exploration Process. Given a document's collection and associated data structures describing the content of its articles, the data scientist can explore articles to determine whether they report meteorological events. This phase integrates the human-in-the-loop. The reason is that newspaper articles use colloquial terms that can be tricky and refer to metaphors that might not denote a meteorological event. Language subtleties are not easy to handle manually, mainly because we are dealing with a language used some centuries ago, which increases the challenge of classifying the content of the articles.

3. Meteorological Events Knowledge Model

We propose a meteorological event knowledge model (see Figure 5) to represent climate event reports in digital documents. The objective is to describe events from different perspectives using the information from the articles and newspapers that report them in an empirical form and complete their description with domain knowledge also described in the model. Newspapers do not describe events scientifically; however, we need to locate and profile them by approximating quantitative characteristics to picture the past climate situation in the region. The different perspectives give context to the quantitative features derived/deduced from the descriptions. As shown in Figure 5, events are associated with the newspaper article(s) that describe them (reading from right to left). Each article can have metadata that curates it, pointing to its “raw” content that has been processed and annotated with linguistic labels.

Classes of documents associated with an event (class *Event* in the figure) contain variables that describe its characteristics, like the date it happened or the geographical scope.

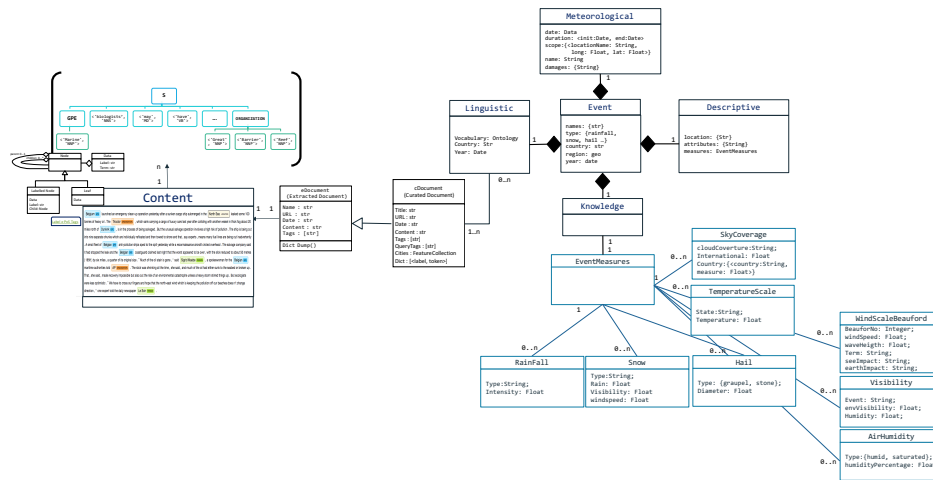


Fig. 5. Event Data Model

According to the perspectives, the event knowledge model provides concepts for representing a *meteorological event*. Each aspect of the knowledge model is implemented using different data structures with associated operations to support exploration actions. The following lines describe the different perspectives of an event and are represented by the event model: descriptive, meteorologic, linguistic and knowledge domain profiles. These perspectives are described in the following sections.

3.1. Descriptive Profile

In newspaper articles, there is no generic list of attributes used for describing a meteorological event. Indeed, meteorological events are described in different ways in historical

newspaper articles, depending on the author. However, we can often collect information related to location, date, duration, scope, and damages. Meteorological features (like millimetres of precipitations, wind speed, temperature, pressure, etc.) can be explicitly described in articles or deduced according to the description of the event. For example, an event reported in Montevideo describing an overflow of the river implies winds higher than 100 km/h and rain of more than 10 ml/hour, according to the knowledge provided by meteorologists. This knowledge domain is used to complete the reported event's meteorologic features.

We combine scientific knowledge produced a posteriori with empirical observations reported in colloquial narratives centuries before. This strategy can help to estimate the location of the events. Of course, we could have tried a more appropriate approach correlating the location of the event referred to in the article with ancient distributions and organisation of the territory to have a more precise location of the events. For example, we could have looked for the urban distribution of the city in the publication year of the article. Then, compare this result with contemporary maps and have a more accurate location of the events according to the modern urban distribution of the city. For instance, an event reported in Montevideo city's "Rambla" sector in 1910 corresponds to a new quarter today. We will develop this approach in our future work.

3.2. Linguistic Perspective

The linguistic perspective gathers the terms used for describing an event in one or several articles belonging to a given newspaper. We propose a tree-based data structure, named *content tree* for representing the content of a historical newspaper article. The tree corresponds to each sentence's grammatical analysis in the article's textual content commonly used in Natural Language Processing (NLP) techniques [4]. The **content tree**, as shown below, consists of a set of sentences. A **sentence** is defined as a set of nodes representing grammatical elements of a sentence and leaves representing the terms composing a sentence in a specific article. We use existing classic NLP techniques because we do not aim at contributing to extending or providing novel ways of using them. The objective is to choose adapted methods for processing the meteorologic newspaper texts.

In Spanish, we use a simplified grammatical model defined by the following simplified Backus-Naur Form (BNF) specification⁸. The simplified specification allowed to process the type of articles we explored, of course an extension of the representation in the next versions of LACLICHEV will allow process other texts describing meteorological phenomena for example in historical novels with narratives about major events:

```
<sentence> ::= <noun-sentence> | <verb-sentence>
<noun-sentence> ::= <named-entity> <conjunction>
                    <noun-sentence>
<noun-sentence> ::= <noun>
<verb-sentence> ::= <subject> <predicate>
<subject> ::= <article> <noun>
<predicate> ::= <verb> <direct-object>
```

⁸ We have also used a BNF for English to explore the use of LACLICHEV with other languages. This work is out of the scope of this paper and concerns the next version of LACLICHEV.


```

<direct-object> ::= <article> <noun>
<article> ::= EL | LA | UN | UNA
<noun> ::= "Spanish nouns"
<verb> ::= "Spanish verbs"

```

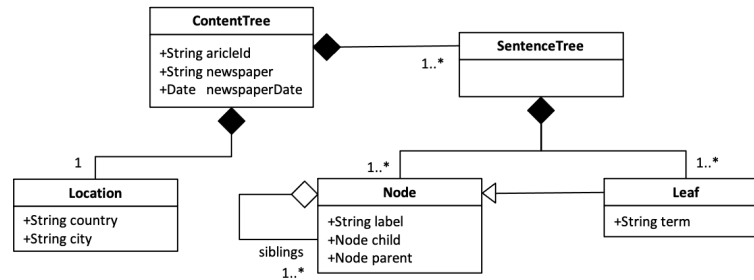


Fig. 6. UML class diagram representing the general structure of a content tree

As shown in Figure 6, a **node** represents a type of grammatical element given in a specific linguistic model defined for a specific language. It is labelled adopting the entity labels produced by classic natural language processing tools known as Part Of Speech (POS) tags. For instance, *noun, proper singular* (NNP), *noun, plural* (NNS), *verb, modal auxiliary* (MD), *Geopolitical entity* (GPE), or Organization. In the case of subjects (NNP), they can be grouped into more general entities that identify geographic locations (GPE), places, names, and organization⁹.

A **node** has children, where each child can also be a Node or a Leaf, and a set of siblings, which are other nodes. A **leaf** specializes in a node, and it represents a term contained in the article. A term is a string with a parent, a **node** means a POS tag.

According to the model, the **ContentTree** represents a document's content where the vocabulary used is determined by a **Location** in a country and a city. These classes represent that the same language, Spanish, varies among countries and cities. Recall that in different locations, people describe meteorological events using different vocabulary.

Every article in a newspaper is associated with its content tree. A data analyst or expert domain can explore the articles by navigating their content trees without reading the full content. For example, *retrieve articles reporting heavy storms in Uruguay in December 1810*. Nodes are related through two relation types: instance, correlation. The relation of type correlation describes two terms that appear in the same sentence with a given distance given by the number of intermediate terms.

3.3. Meteorological Perspective

The meteorological perspective characterizes the event with attributes used to describe it in one or several newspaper articles. Nevertheless, not all the attributes can have an associated value since there might be no evidence within the articles that report it. Attributes,

⁹ A full list of POS tags can be found in <https://www.cms.gov/>

like the date of the event, its geographical scope, or the location of the damaged regions, are computed by navigating through the *content tree* of every article reporting the event.

```
MeteorologicEvent: <date: Data,
    duration: <init:Date, end:Date>
    scope: {<locationName: String,
    long: Float, lat: Float>}
    name: String, damages: {String}>
```

3.4. Knowledge Domain Perspective

The knowledge domain perspective describes meteorological events using knowledge domain statements created by experts of the National Library of Uruguay. This knowledge has been associated with events through manual analysis of newspaper collections and meteorologists interacting. This knowledge can help interpret the empirical information reported in the articles and complete the information associated with the event description. For example, if the river was flooding due to a storm, it is possible to estimate the wind speed and the approximate litres of rain. The knowledge domain perspective is modelled as a glossary. Figure 7 shows the intuition of its structure.

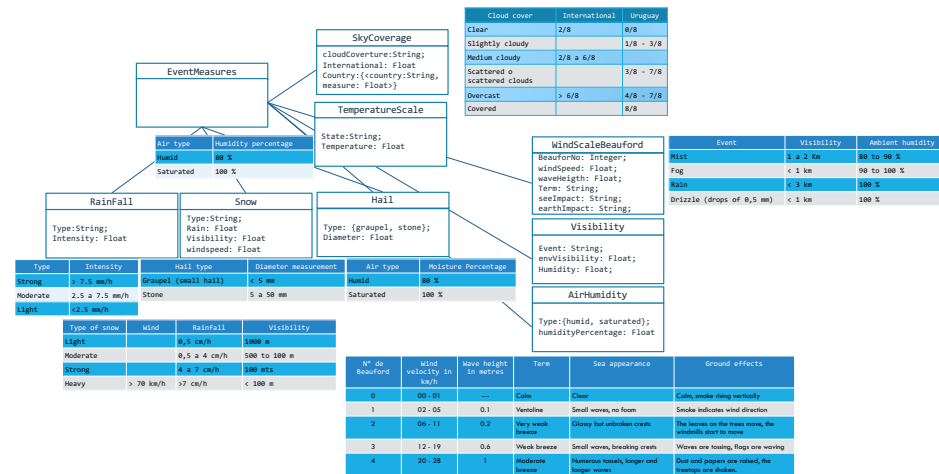


Fig. 7. Climate events glossary

Modelling the empirical knowledge about meteorological events is critical when curating newspapers’ descriptions. It represents the interpretation of the emerging content by observing the phenomena and associating it with metering techniques available today. The principle can be stated as follows: *“today, based on the metrology performed during meteorological events, we know that when the river floods, there is an approximate wind speed and more than “x” litres/meter² of rain. So we can estimate the conditions in which the events could have happened in the past.*

4. LACLICHEV in Action

LACLICHEV is a client-server system for executing the human-in-the-loop tasks that implement the data exploration process. We have configured LACLICHEV to process historical newspapers of four countries provided by the national libraries of each country. The curated event history has been explored by the librarians of the participating countries. The idea was not to experimentally test the system but to calibrate it according to the characteristics of the digital collections.

4.1. Building a Latin American Meteorological Event History

We have worked with the national libraries of Mexico, Colombia, Ecuador, and Uruguay to access their newspapers' digital collections. For our experiments, we worked with the collections of the XVIII and XIX centuries of newspapers written in Spanish with the linguistic variations of those mentioned above Latin American countries. The National Libraries of these countries manage historical newspapers with about 4 to 7 million images of newspapers between the XVIII and XIX centuries, depending on the country. For example, the National Library in Mexico maintains 7 million images of digital national newspaper collections. In Colombia the newspaper library is made up of publications published between the end of the 18th century and the first half of the 20th century, including: "El Papel Periódico Ilustrado", "Diario Político de Santafé de Bogotá", "El Alacrán", "El Mosaico", "Semanario del Nuevo Reyno de Granada". It includes newspaper collection from Ecuador and Argentina, namely "La Verdad Desnuda (Guayaquil, Ecuador) and "Vida Intelectual" (Santa Fe, Argentina). The current version of LACLICHEV processed around 19 million images in the newspapers of the fourth countries. The event history has curated 800 different meteorological events.

We curated collections and generated the vocabulary used on articles identified as reporting a meteorological event (see Figure 8). Digital newspaper collections remain in the initial repositories that belong to the libraries. Then, terms and links to the OCR (Optical Character Recognition) archives containing documents with articles reporting meteorological events were stored in distributed histories managed in each country. As shown in Figure 8, the process consists of five steps usually used in natural language processing techniques: sentence segmentation, tokenization, speech tagging, entity and relation detection. LACLICHEV implements these phases in Python, relying on the NLTK library.

The first phase of the pre-processing process of newspapers leads to graphs representing the content of the articles and classic inverse index and frequency matrices used for performing exploration queries.

Besides curating the data collections' content, we wanted to discover linguistic variations in different Latin American countries to describe meteorological events. People's language and variations can picture civilians' perception of these events, consequences, and associated explanations. Thus, local vocabularies were created out of the terms used in newspapers' articles (see Figure 9). For example, referring to a storm as a stormtrooper¹⁰ Then we updated and enriched through queries, exploration and analytic activities, these vocabularies through human-in-the-loop actions. Data analysts tagged "colloquial" terms

¹⁰ In Mexico, a storm is called a "chaparrón" and in Uruguay, it is called a "chubasco".

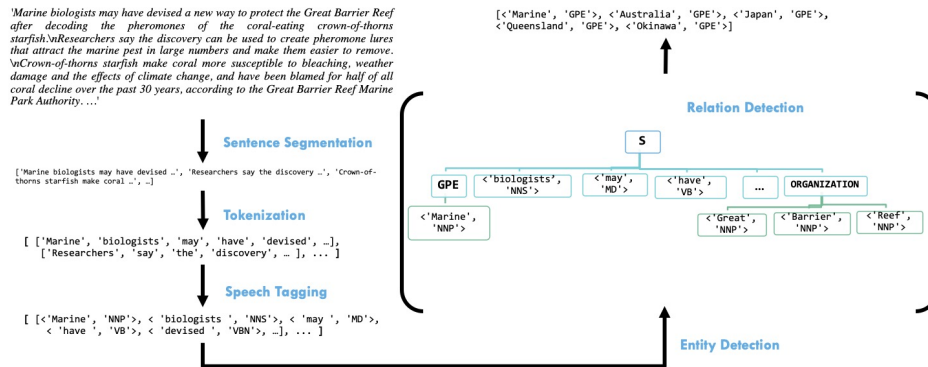


Fig. 8. Pre-processing text pipeline

used to describe meteorological events and associated them with more scientific terms. These terms can be then used for defining keyword queries for exploring newspaper datasets.

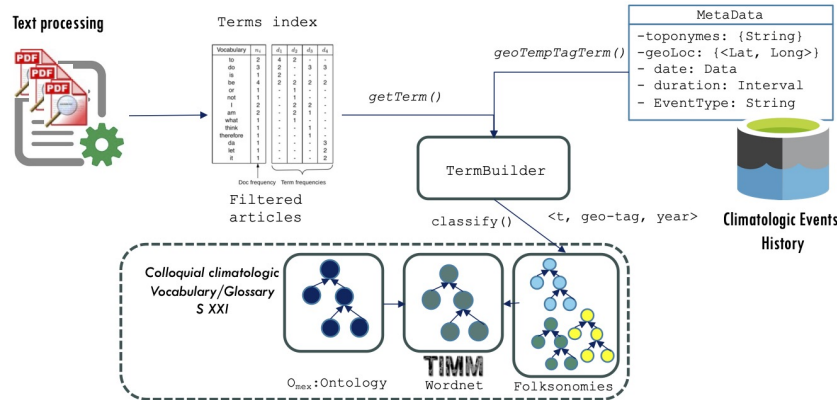


Fig. 9. Collecting colloquial vocabulary

4.2. Curating Data Collections

LACLICHEV proposes functions that data scientists may exploit through diverse functionalities. Next, we present the type of functions of LACLICHEV’s API (application programming interface). The implementation of these functions were adapted to the case of historical newspaper collections:

- **Curating data collections** by exploring and processing their content for building the history of meteorological events possibly related to climate change in the considered Latin American countries. The functions for processing texts in Spanish are the core of LACLICHEV. They were coupled with other functions to extract, derive and associate as much data as possible to articles describing meteorological events.

Curation tasks were performed on a collection of textual digital documents with minimum associated metadata, particularly those used by digital libraries that own the collections. Each library adopts its metadata schema, but they generally specify the newspaper's name, the country, the date and number, the number of pages, and the window time in which it circulated. Libraries export the metadata schema used to describe these resources and align them to standards used by digital libraries. For example, the editions of the collection of Uruguayan newspapers were published during the first 10 years of the XIX century.

The curation process generated data structures that provide an abstract representation of the content of each article describing an event. A frequency matrix integrated the terms representing the content of articles extracted from the different libraries' collections. This matrix was sharded and allocated to the servers devoted to interacting with each library. This strategy implies having queries evaluated on different servers. This distributed query evaluation was supported by an inverted index that provided information about the documents containing specific terms and their location. With the inverted index, the curation process also created initial vocabularies, classified by location (country and city) and year. These vocabularies classified the terms used to describe climatologic events in the different Hispanophone countries in LATAM. The temporal dimension allowed to store information about their evolution.

Querying the event's history of already tagged events can be done by keyword oriented queries (e.g., locate the most famous events in Mexico during the XVIII century). Users decide to use some terms that can belong to any of the vocabularies generated in the pre-processing phase. LACLICHEV applies query rewriting techniques to extended user-expressed queries with synonyms, subsuming and general terms. The particular characteristic of this task is that the user (i.e., data analyst) can interact and guide the process according to her/his knowledge and expectations about what she/he expects to explore and search. The first result of this process, based on a "queries as answers approach", is a set of queries that can potentially provide the largest number of results stemming from the collections of the different libraries. The details of the approach we proposed for LACLICHEV is detailed in the following section.

- **Analytics operations and analysis results** are generally presented within maps (e.g., how did rainy periods evolved in the region?). In the current version of LACLICHEV analytics queries cannot be expressed in the frontend. They are implemented manually through notebooks running on top of the event history. The analytics queries concerning aggregative queries on the event's history, for example, the number of events happening in a country within a specific time window. The average wind speed and millimetres of water per hour deduced for events regarding rainfalls and hurricanes in Montevideo. Classifying the terms used for describing specific types of events. The event history is a curated and clean data collection on top of which other analytics models can be applied for discovering knowledge. This characteristics open analytics perspectives for future uses of LACLICHEV. For example, applying supervised learning for analysing newspapers

articles and determining whether they describe meteorological events. This can allow to semi-automatise the curation process and enhance it with a recommendation system.

- **Managing vocabularies**, adding terms, guiding their classification and studying the linguistic connections between the terms used in the different countries. The vocabularies in the current version of LACLICHEV are implemented as RDF ontologies, and it relies on SPARQL mechanisms for querying them. This version mainly addresses the construction of vocabularies and their maintenance as new terms are identified in events' descriptions.

Next subsections describe exploration techniques implemented for meteorological events in the history built through the newspaper articles in the Latin American countries we used.

4.3. Query Rewriting

Queries-as-answers Exploration. Data analysts can express queries that can potentially explore historical newspaper content to find articles describing meteorological events. The aim is to have a good balance between precision and recall despite the ambiguity of the language (Spanish variations in naming meteorological events). The domain experts must express “clever” queries that can exploit the collections to achieve this goal.

Queries can be initially conjunctive and disjunctive expressions combining terms chosen from the built-in vocabularies or not. Then, queries are rewritten in an expression tree where nodes are conjunction and disjunction operators and leaves are terms, according to an input query expressed as a conjunction and disjunction of terms potentially belonging to a meteorological vocabulary.

Our approach for rewriting queries is based on a “queries-as-answers” process. This technique rewrites user queries into queries that can produce more precise results according to the explored dataset content. Queries as answers proposed by LACLICHEV consist of a list of frequently used queries. Thus, we focus on the following aspects:

Extending Query Alternatives using Hypernyms and Synonyms. An initial conjunctive or disjunctive query is rewritten by extending it with general and more specific terms, synonyms, etc. The terms used to express the query are colloquial vocabulary for denoting meteorological events. The rewriting process can be automatic or interactive, in which case the system proposes alternatives, and the user can validate the proposed terms. For example, if the query is “*heavy storms*”, the query can be completed by adding “*heavy stormtrooper*”, “*heavy storm dust*”. It can also be rewritten with synonyms for the adjective *heavy*. In that case, it creates a combinatorial set of rewritten queries.

Note that the colloquial vocabulary stems from the articles of the curated newspapers. As they are curated, the terms used in the articles feed a vocabulary that is first organised in the frequency matrix produced when texts are processed as part of the curation process.

Then we use Wordnet¹¹ to look for associated terms and synonyms that help address concepts used in different Spanish-speaking countries. We do not translate the query terms to other languages because our digital data collections contain Spanish newspapers. LACLICHEV allows equivalent terms searching to morph a query. For a new term, LACLICHEV generates a node with the operator and then connects the initial term with the equivalent terms in a disjunctive expression subtree. Thereby, more general terms are

¹¹ <http://timmm.ujalen.es/recursos/spanish-wordnet-3-0/>

collected and related to the initial term with these terms in a conjunctive expression subtree. The result is a new expression tree corresponding to an extended query Q_{ExT} . The query morphing algorithm behind LACLICHEV is described in [35].

Extending Query Alternatives using Cultural Terms. Use local vocabularies for generating new query expression trees that substitute the terms used in Q'_{ExTi} with equivalent terms used in a target country (e.g., blizzard instead of a heavy storm). This will result in transformed expression trees each one using the terms of a country ($Q''_{ExT1} \dots Q''_{ExTj}$) [38].

We call metaphorically “folksonomies” a series of vocabularies created by processing newspaper articles “local” vocabulary. We make and feed each vocabulary according to the country of origin of the processed newspaper article. This lets us extract the vocabulary used during the XVIII and XIX centuries for describing meteorological events in Latin American countries (i.e. Mexico, Colombia, Ecuador, and Uruguay). Using this information, LACLICHEV can answer the following queries: *How have terms used to describe meteorological events changed between XIX-XX c.? Which are standard terms used to describe meteorological events across Latin American countries? Which is the distance between terms used in XIX-XX c.? Which are the most popular terms used in XIX c. for describing meteorological events?*

Defining Filters using Knowledge Domain. We also use domain knowledge for rewriting the queries. We have a knowledge base provided by domain experts that contains some meteorological event rules. For example, rules state that in the presence of a heavy storm: R1. the wind speed is higher than 118 km/h; R2. the rivers can grow and produce big waves; R3. there are rains between 2,5 7,5 mm/h; R4. the range of surface that can be reached by a 100 km wind speed storm is of 1000 km.

Our approach uses this information to generate possible queries that help the domain expert better precise her/his query or define several queries that can represent what she/he is looking for. For example, the previous initial query “ Q_1 : heavy storm” is rewritten into new additional queries: “ Q_{11} : heavy storm *or storm with wind speed > 100 km*” (using R1). “ Q_{12} : storms with 100 km speed that reached Mexico City” (using R2 and knowing the initial point and geographic information). “ Q_{13} : storms touching villages 500 km around Mexico city happening in the same period” (R4). Instead of having a long query expression, our approach proposes queries that the domain expert can choose and combine. Note that the system first generates queries, not answers. The answer to a query is a family of possible queries with some associated samples. The user can then choose those queries that she wants to execute.

A climate glossary associates a term referring to a meteorological event with terms of the LODE ontology¹²). LODE is an ontology for historical publishing events as Linked Data and physical variables describing events. This information generates new queries, which help users discover more details about historical meteorological events.

Using the climate glossary for transforming Q_{ExT} into queries with terms that can serve as filters. There are variables concerning meteorology concepts in the glossary, like wind speed, rain volume/hour, and the water level of seas, rivers, and lakes. Other variables involve geographic aspects, like the location of an event and the scope of land it reaches. Finally, other variables concern damages caused by a climate event with specific physical and geographic characteristics. These different options generate queries com-

¹² <http://linkedevents.org/ontology/>

binning variables of the same group and different groups. For example, “heavy storms with winds higher than 150 km/h”, “heavy storms with rains higher the 10 mm per square metre”, and “heavy storms with rivers’ overflow”. The result is a set of queries $Q^{ExT1} \dots Q^{ExTj}$.

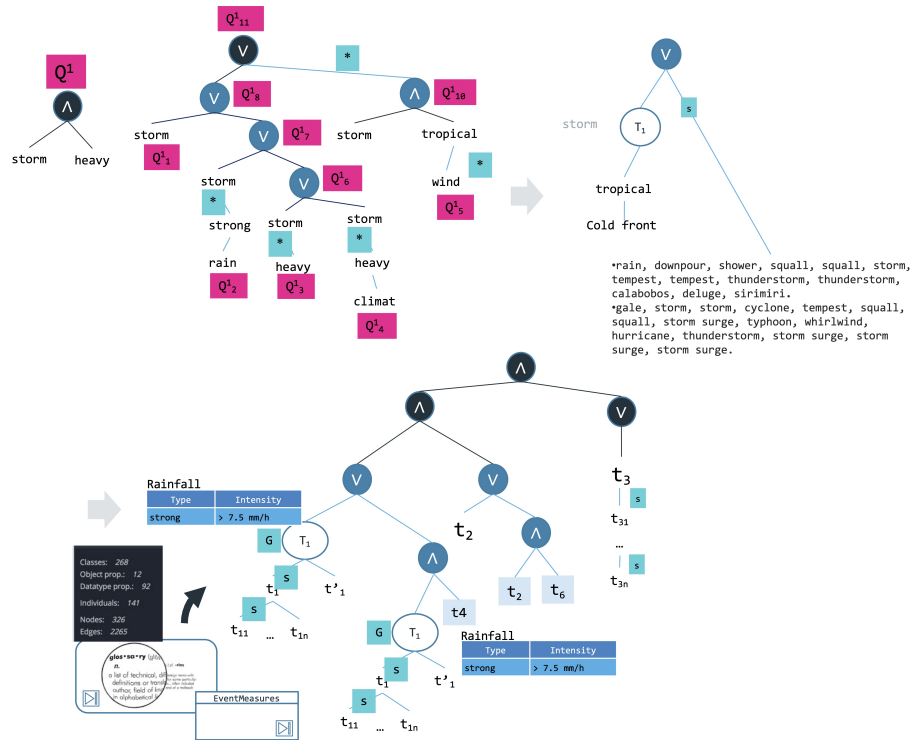


Fig. 10. Queries as answers example

Figure 10 shows an example of the general principle of the queries as answers approach adopted by LACLICHEV. The system rewrites an initial conjunctive query heavy, storm adding concepts (i.e., terms) related to the terms “storm” and “heavy”. The figure only shows the rewriting process of the term “storm” for pedagogic reasons. Then in a second round, the system rewrites the query adding synonyms of the terms, as shown in the upper right side of the figure. Finally, the query is rewritten according to the rules stated in the glossary. LACLICHEV performs a ranking process for the rewritten queries according to the coverage of their potential answer. The queries with the biggest coverage (those that include the largest subset of events in the history database). The algorithms to estimate the coverage of a documents collection are proposed in [35].

4.4. Evaluating Queries

The evaluation process of the query is performed first on top of the curated event store. The result is a set of items (events) that answer, to some extent, the query. We also started to generate maps depicting the events reported in the history [6, 39].

Analytics Queries. LACLICHEV provides and maintains the meteorological event's history on top of which users can visualize information and perform analytical tasks. For example, LACLICHEV can answer spatio-temporal queries like:

- Q_1 *Locate meteorological events in the XVII century,*
- Q_2 *Enumerate and locate the most famous events in the region or in a specific country,*
and
- Q_3 *Create a heat map of events in Latin America in the last years of the XIX century.*

The objective is to answer analytic queries that imply aggregating information stored in the event's history. For example, *How did rainy time evolve in time in the region?, In which way was climate different between XVII and XIX centuries? How did vocabulary evolve from colloquial to scientific and standardized in the XX century?*

4.5. Scope and Limitations

LACLICHEV is running its first version; we expect to enrich the number of digital newspapers digitalised in the libraries. These new items will imply a new curation process that will improve the event history in two directions. First, more articles will describe the already curated events; this will complete the information stored in the history. Second, with more events, we will further test and enhance the analytics queries that require to have a specific volume of data to generate representative maps and analyses about the meteorological events that happened in the past.

In future versions, and with more curated events, LACLICHEV is willing to answer prediction queries like *Could it have been possible to predict the evolution of climate behaviour from the data in XVIII and XIX centuries?.* This query requires collecting, curating, and preparing more newspaper articles and other complementary data. However, it concerns future work.

Another limitation of the current LACLICHEV is that it does not provide the adapted mechanisms for exploring the linguistic aspects of the vocabulary. It gathers the terms and organises them in a mesh data structure. Still, it does not provide tools for curating the languages and allowing an analytics exploration of their use of meteorological across countries and time.

5. Related Work

Historical analysis of climate behaviour can explain climatologic phenomena and Earth's climate behaviour. There exist several scientific efforts to study the history of climate change. The *Climate of the Past* [1], for example, is an international scientific journal dedicated to the publication and discussion of research articles, short communications, and review papers on Earth's climate history. The journal covers all temporal scales of climate change and variability, from geological time to multidecade studies of the last

century. The Government of Canada provides access to historical observations on climate in Canada starting from 1840 [2]. However, these data collections are disconnected and use different reference variables and observation criteria. They are very heterogeneous and tight to their region. This ad-hoc characteristic is why data curation and exploration processes are essential to extract knowledge that can be digitally analyzed and correlated.

Several domains address aspects that converge in our work, particularly those with certain originality, like data exploration techniques, geographic information retrieval and visualization. The following lines summarise the methods and approaches related to those proposed for LACLICHEV.

5.1. Data Curation

Data curation [14, 33] is the art of processing data to maintain it and improve its interest, value, and usefulness through its lifecycle, i.e. improving the quality of the data. Therefore, it implies (i) discovering data collections of interest; (ii) cleaning and transforming new data; (iii) semantically integrating it with other local data collections; and (iv) deduplicating the resulting composites if required. Data curation provides the methodological and technological data management support to address data quality issues, maximizing the usability of the data for analytics and knowledge discovery purposes.

Existing commercial and academic systems provide solutions for curating data [36, 29, 40]. They provide operations for modelling and extracting metadata from raw data collections, and they provide tools for exploring them. Prominent commercial examples are Apache Atlas¹³ and SolR¹⁴. Apache Atlas is a framework for governance and management of metadata. It offers curation functions for metadata typing and classification, data lineage, and exploration functions such as data source search.

SolR is a document indexing system including XML files, comma-separated value (CSV) files, data extracted from tables in a database, and files in standard file formats such as Microsoft Word or PDF. Indexing documents can be used as an essential general-purpose curation operation. Its major exploration features include full-text search, hit highlighting and faceted search. Other solutions are built on top of these tools for providing end-to-end general-purpose systems for curating and exploring data, for example, ATLAN¹⁵.

5.2. Data Exploration

The emergence of the notion of data exploration provides different perspectives of the data and tools for helping data scientists choose and compound datasets adapted for target experiments [23, 5]. The tools [17] include functions like “data grooming” [27], which denotes transforming raw data into analyzable data with various data structures. Other approaches [24] focus on transforming human-readable data into machine-readable data considering inconsistencies in data formatting given that they are produced under different conditions. The idea is to exhibit processes, digital spaces, and systems that host datasets and provide them with access to understand the conditions in which data are processed.

¹³ <https://atlas.apache.org>

¹⁴ <https://solr.apache.org>

¹⁵ <https://atlan.com>

Data Grooming denotes transforming raw data into analysable data with various data structures. Multi-scale queries propose to split a query into multiple queries executed on different database fragments and then perform a union of those queries. This allows scaling the query size as the user gets more confident in her query. Result set post-processing and query morphing go on the premise that the user probably does not need the exact answer to a query. Result set post-processing assumes an array of simple statistical information such as min, max, and mean to be more helpful, especially on massive data sets. Query morphing assumes queries can be wrongly formulated. Query morphing still focuses on answering the query given by the user but will also use a small portion of resources in searching data around the original query. *Query Morphing*. Another trend regarding data exploration is to tackle the lack of knowledge a user may have on the dataset. Query morphing and queries as an answer are rewriting techniques that compute alternative queries (e.g. adding terms) that can potentially better explore a dataset than an initial query. Approaches such as interactive query expansion (IQE) [30, 8, 19] have shown the importance of data consumers in the data exploration process. Users' intention helps navigate the unknown data, formulate queries and find the desired information. In most occurrences, user feedback acts as vital relevance criteria for the following query search iteration. The key challenge is identifying bad queries using statistical information or massive scientific databases and identifying interesting queries to return. Identifying bad queries can be done using a list of frequently used queries and returning them based on user feedback.

SVD/PCA [15] is probably the most known algorithm for exploring data sets. It is used to reduce high-dimensional data represented as a matrix. From a practical perspective, it searches for the combination of weighted attributes that expresses the most information, allowing data analysts to work with the more useful 2 or 3-dimensional graphs. From a geometric perspective, these techniques search for the vectors with the highest variance and then express the original matrix according to this new system of dimensions. Using Eigenvalues makes it possible to estimate the amount of information in each dimension [12].

Visualization and Summarization. are essential to understand the data and maintain it. The field of visual analytics seeks to provide people with better and more effective ways to understand and analyze these large datasets while also enabling them to act upon their findings immediately [22]. Visual analytics provides technology [26, 28] that combines human and electronic data processing strengths. Structured query languages and the graphical interface developed over the top are the standard procedure for accessing data in a database. Many tools exist to perform data visualization with web visualization tools such as D3.js or other tools such as Matlab [20] or R programming language [16]. One of the most critical steps of these tools is to let the data analyst move from confirmatory data analysis (using charts and other visual representations to present results) to exploratory data analysis (interacting with the data/results). This has led to visual data exploration and visual data mining [7].

5.3. Text Processing in Newspaper Articles

The discovery of knowledge from large-scale text data or semi-structured data is a difficult task that can be addressed with text mining techniques. These techniques extract valuable information to fulfil a user information need. The textual documents available

in unstructured and semi-structured forms can be medical, financial, market, scientific, and other documents. Text mining applies a quantitative approach to analyse a massive amount of textual data and tries to solve the information overload problem.

The combination of transformers and self-supervised pretraining has been responsible for a paradigm shift in NLP, information retrieval (IR), and beyond [25]. The approach in [21] extracts target categories, each including many topics. The method extracts word tokens referring to topics related to a specific category. The frequency of word tokens in documents impacts the document's weight calculated using a numerical statistic of term frequency-inverse document frequency (TF-IDF). The proposed approach uses the title, abstract, and keywords of the paper and the categories of topics to perform a classification process. The documents are classified and clustered into the primary categories based on the highest cosine similarity measure between category weight and documents' weights.

The work proposed in [3] discusses the challenge of processing and analysing historical manuscripts. Authors investigate how deep learning models detect and recognise handwritten words in Spanish American notary records. For dealing with natural language (ancient Spanish), professional historians prepared a labelled dataset of 26,482 Spanish words employed in the experiments. The paper [31] proposes a tool that uses raw Spanish text and Spanish event coders for analysing political news articles. The work combines natural language processing techniques, including deep learning and encoders, with the knowledge represented in ontologies to support the automated coding process for Spanish texts.

5.4. Geographic Information Retrieval

Within GIScience domains, some approaches have developed. [10] and [9] combined methods from Geographic Information Retrieval (GIR) and geovisual analytics to obtain new insights from a digital dictionary about the history of Switzerland. In addition, the authors include sentiment analyses to assess how (historical) places were referred to in texts over time and provide ways to access and explore spatio-temporal information contained in many text archives. [34] described a method to supplement existing records of landslides in Great Britain by searching an electronic archive of regional newspapers. Moreover, the authors construct a Boolean search criterion by experimenting with landslide terminology for four training periods. It allowed the discovery of some spatio-temporal patterns of additional landslides identified in newspaper articles. [41] presented a text-mining program that extracted keywords related to floods' geographic location, date, and damages from newspaper analyses of flash floods in Fujairah, UAE, from 2000 to 2018. Furthermore, this work performed geocoding and validating flood-prone areas generated through Geographic Information System (GIS) modeling.

5.5. Discussion

Any query and analysis must be based on a good understanding of the available data collections because the way they are combined and analyzed impacts the quality and accuracy of the results.

Existing solutions are not delivered in integrated environments that data analysts can comfortably use to explore data collections. The technical effort is still necessary to combine several tools to explore and process datasets and go from raw independent data sets

to knowledge, for example, on climate change. Therefore, our research aims to tailor a data exploration environment to help explore digital data collections using a human-in-the-loop approach. In existing solutions, data analysts cannot comfortably explore data collections and design analytics settings, particularly in cases where documents and questions combine scientific observations with empirical observations, like in the case of meteorological events described empirically in the past.

The current version of LACLICHEV did not explore the linguistic aspect, with original or more advanced methods studying texts and combining present and past observations to try to derive conclusions, for example, about climate change.

A technical effort is still necessary to combine several tools to explore and process datasets and go from raw independent data sets to knowledge, understanding and prediction, for example, on climate change. Therefore, LACLICHEV aimed to tailor a data exploration environment that could help explore digital datasets using a human-in-the-loop approach.

Regarding the qualitative assessment of LACLICHEV, we have not run user experience testing to collect feedback and user experience, and we might perform such testing in the future. For the time being, we focus on the analytics such as correlating different descriptions of the “same” event from articles in various newspapers, the location of meteorological events in old maps and their correlation with modern maps. We are working on creating historical cartography of meteorological events that can be confronted with contemporary perceptions of such events.

6. Conclusion and Future Work

The democratisation of access to data collections opens possibilities for exploring content produced over the years and extracting knowledge that can contribute to understanding critical phenomena like climate change. Rather than directly querying collections for searching documents or performing data analytics operations (statistics, correlations), the objective is to let data scientists understand the content of the collections and then decide what kind of queries to ask. Data exploration is a complex and recurrent process that includes calibrating a querying strategy (defining queries as answers) that can increase the scope of content that can be retrieved and possibly analysed to extract evidence around hypotheses or claims. This new paradigm calls for data curation strategies that are well adapted to describe the content of collections with the right metadata and abstractions.

Our work contributes to data curation and exploration adapted for Spanish textual content within digital newspaper collections. Using well-known information retrieval and analytics techniques, we developed a data exploration environment named LACLICHEV that provides tools for understanding the content of collections. We used digital newspaper collections for applying such techniques for building and analyzing the history of meteorological events possibly related to climate change in Mexico, Colombia, Ecuador, and Uruguay. The work reported here is the first step toward this ambitious challenge. We continue enriching data collections, developing and testing solutions for generating and sharing step by step this history.

References

1. Climate of the past: An interactive open-access journal of the european geosciences union. <http://www.climate-of-the-past.net>, european Geosciences Union, Accessed: 2021-04-23
2. Historical climate data. <http://climate.weather.gc.ca>, government of Canada, Accessed: 2021-04-23
3. Alrasheed, N., Prasanna, S., Rowland, R., Rao, P., Grieco, V., Wasserman, M.: Evaluation of deep learning techniques for content extraction in spanish colonial notary records. In: Proceedings of the 3rd Workshop on Structuring and Understanding of Multimedia heritAge Contents. pp. 23–30 (2021)
4. Amavi, J., Ferrari, M.H., Hiot, N.: Natural language querying system through entity enrichment. In: ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium. pp. 36–48. Springer (2020)
5. Amer-Yahia, S., Koutrika, G., Bastian, F., Belmpas, T., Braschler, M., Brunner, U., Calvanese, D., Fabricius, M., Gkini, O., Kosten, C., et al.: Inode: building an end-to-end data exploration system in practice [extended vision]. arXiv preprint arXiv:2104.04194 (2021)
6. Ballari, D.: Visualizar datos georreferenciados, <http://rpubs.com/daniballari/ipgh-visgeoref>
7. Battle, L., Stonebraker, M., Chang, R.: Dynamic reduction of query result sets for interactive visualization. In: 2013 IEEE International Conference on Big Data. pp. 1–8 (Oct 2013)
8. Belkin, N.J.: Some (what) grand challenges for information retrieval. In: ACM SIGIR Forum. vol. 42, pp. 47–54. ACM New York, NY, USA (2008)
9. Bruggmann, A., Fabrikant, S.I.: How does giscience support spatio-temporal information search in the humanities? *Spatial Cognition & Computation* 16(4), 255–271 (2016)
10. Bruggmann, A., Fabrikant, S.I., Janowicz, K., Adams, B., McKenzie, G., Kauppinen, T.: Spatializing a digital text archive about history. In: CEUR Workshop Proceedings. pp. 6–14. No. 1273, CEUR-WS (2014)
11. Carvalho, D.A.S., Souza Neto, P.A., Ghedira-Guegan, C., Bennani, N., Vargas-Solar, G.: Rhone: A quality-based query rewriting algorithm for data integration. In: New Trends in Databases and Information Systems. pp. 80–87. Springer International Publishing, Cham (2016)
12. Chawla, S., Zheng, Y., Hu, J.: Inferring the root cause in road traffic anomalies. In: 2012 IEEE 12th International Conference on Data Mining. pp. 141–150 (Dec 2012)
13. Comesaña, D., Vilches-Blázquez, L.M.: A study of the latin american newspapers from xix-xx centuries with a focus on meteorological events. *Revista de historia de América* (156), 29–59 (2019)
14. Curry, E., Freitas, A., O’Riáin, S.: The role of community-driven data curation for enterprises. In: Linking enterprise data, pp. 25–47. Springer (2010)
15. Feldman, D., Schmidt, M., Sohler, C.: Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In: Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms. pp. 1434–1453. SIAM (2013)
16. Foundation, T.R.: The r project for statistical computing. (2018), <https://www.r-project.org>
17. Gewers, F.L., Ferreira, G.R., Arruda, H.F.D., Silva, F.N., Comin, C.H., Amancio, D.R., Costa, L.d.F.: Principal component analysis: A natural approach to data exploration. *ACM Computing Surveys (CSUR)* 54(4), 1–34 (2021)
18. Goodchild, M.F.: Giscience, geography, form, and process. *Annals of the Association of American Geographers* 94(4), 709–714 (2004)
19. Goswami, P., Gaussier, E., Amini, M.R.: Exploring the space of information retrieval term scoring functions. *Information Processing & Management* 53(2), 454–472 (2017)
20. Inc., T.M.: Matlab (2018), <https://www.mathworks.com/products/matlab.html>
21. Jalal, A.A., Ali, B.H.: Text documents clustering using data mining techniques. *International Journal of Electrical & Computer Engineering* (2088-8708) 11(1) (2021)

22. Keim, D.A.: Visual exploration of large data sets. *Commun. ACM* 44(8), 38–44 (Aug 2001), <http://doi.acm.org/10.1145/381641.381656>
23. Kersten, M.L., Idreos, S., Manegold, S., Liarou, E., et al.: The researcher’s guide to the data deluge: Querying a scientific database in just a few seconds. *PVLDB Challenges and Visions* 3(3) (2011)
24. Kumar, M.S., Rajeshwari, J., Rajasekhar, N.: Exploration on content-based image retrieval methods. In: *Pervasive Computing and Social Networking*, pp. 51–62. Springer (2022)
25. Lin, J., Nogueira, R., Yates, A.: Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies* 14(4), 1–325 (2021)
26. Liu, X., Alharbi, M., Best, J., Chen, J., Diehl, A., Firat, E., Rees, D., Wang, Q., Laramée, R.S.: Visualization resources: A starting point. In: *2021 25th International Conference Information Visualisation (IV)*. pp. 160–169. IEEE (2021)
27. Liu, Y.: Exploring a corpus-based approach to assessing interpreting quality. In: *Testing and Assessment of Interpreting*, pp. 159–178. Springer (2021)
28. Mohammed, L.T., AlHabsby, A.A., ElDahshan, K.A.: Big data visualization: A survey. In: *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. pp. 1–12. IEEE (2022)
29. Rehm, G., Bourgonje, P., Hegele, S., Kintzel, F., Schneider, J.M., Ostendorff, M., Zaczynska, K., Berger, A., Grill, S., Räuchle, S., et al.: Qurator: innovative technologies for content and data curation. *arXiv preprint arXiv:2004.12195* (2020)
30. Ruthven, I.: Re-examining the potential effectiveness of interactive query expansion. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. pp. 213–220 (2003)
31. Salam, S., Khan, L., El-Ghamry, A., Brandt, P., Holmes, J., D’Orazio, V., Osorio, J.: Automatic event coding framework for spanish political news articles. In: *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. pp. 246–253. IEEE (2020)
32. Sodr e, A.P., Floriano, L.E.M., Magalhaes, D., Aguiar, C.D., Pozo, A., Hara, C.S.: Comparing alternative storage models for words extracted from legal texts. In: *Anais Estendidos do XXXVI Simp sio Brasileiro de Bancos de Dados*. pp. 36–42. SBC (2021)
33. Stonebraker, M., Bruckner, D., Ilyas, I.F., Beskales, G., Cherniack, M., Zdonik, S.B., Pagan, A., Xu, S.: Data curation at scale: the data tamer system. In: *Cidr*. vol. 2013 (2013)
34. Taylor, F.E., Malamud, B.D., Freeborough, K., Demeritt, D.: Enriching great britain’s national landslide database by searching newspaper archives. *Geomorphology* 249, 52–68 (2015)
35. Vargas-Solar, G., Farokhnejad, M., Espinosa-Oviedo, J.: Towards human-in-the-loop based query rewriting for exploring datasets. In: *Proceedings of the Workshops of the EDBT/ICDT 2021 Joint Conference* (2021)
36. Vargas-Solar, G., Kemp, G., Hern andez-Gallegos, I., Espinosa-Oviedo, J., Da Silva, C.F., Ghodous, P.: Demonstrating data collections curation and exploration with curare. In: *EDBT/ICDT Conference 2019*. p. 4 (2019)
37. Vargas-Solar, G., Zechinelli-Martini, J., Espinosa-Oviedo, J.A., Vilches-Bl zquez, L.M.: LACLICHEV: exploring the history of climate change in latin america within newspapers digital collections. In: Bellatreche, L., Dumas, M., Karras, P., Matulevicius, R., Awad, A., Weidlich, M., Ivanovic, M., Hartig, O. (eds.) *New Trends in Database and Information Systems - ADBIS 2021 Short Papers, Doctoral Consortium and Workshops: DOING, SIMPDA, MADEISD, MegaData, CAoNS, Tartu, Estonia, August 24-26, 2021, Proceedings. Communications in Computer and Information Science*, vol. 1450, pp. 121–132. Springer (2021), https://doi.org/10.1007/978-3-030-85082-1_11
38. Vargas-Solar, G., Zechinelli-Martini, J.L., Espinosa-Oviedo, J.A.: Computing query sets for better exploring raw data collections. In: *2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*. pp. 99–104. IEEE (2018)

39. Vilches-Blázquez, L.M., Ballari, D.: Unveiling the diversity of spatial data infrastructures in latin america: evidence from an exploratory inquiry. *Cartography and Geographic Information Science* 47(6), 508–523 (2020)
40. Visengeriyeva, L., Abedjan, Z.: Anatomy of metadata for data curation. *Journal of Data and Information Quality (JDIQ)* 12(3), 1–30 (2020)
41. Yagoub, M., Alsereidi, A.A., Mohamed, E.A., Periyasamy, P., Alameri, R., Aldarmaki, S., Alhashmi, Y.: Newspapers as a validation proxy for gis modeling in fujairah, united arab emirates: identifying flood-prone areas. *Natural Hazards* 104(1), 111–141 (2020)

Genoveva Vargas-Solar (<http://www.vargas-solar.com>) is principal scientist (HDR) of the French Council of Scientific Research (CNRS), France. She is regular member of the Mexican Academia of Computing. She has obtained a first PhD degree in Computer Science at University Joseph Fourier (2000) and a second PhD degree in Literature at University Stendhal (2005). Her research concerns the design of data management services guided by Service Level Objectives (SLO) providing methodologies, algorithms, and tools for integrating, deploying, and executing data science pipelines on just in time architectures. She has applied her results to e-Science applications in Astronomy, Biology, social sciences, digital-humanities, and industry 4.0. She is an active militant of data and A+I decolonisation and decolonial feminism in data science in the global south. She is a member of the database interconference diversity and inclusion (D&I) initiative on she runs several gender equalities actions in the global north. She has coordinated several research projects in Europe and Latin America financed by governments and industrial partners. She actively promotes the scientific cooperation in Computer Science between Latin America and Europe particularly between France and Mexico.

José Luis Zechinelli Martini holds a PhD in Computer Science and a master's in information and Communication Systems from Grenoble I University. Since 2016 he is a regular member of the Mexican Academy of Computer Science. He has studied the problems associated with the integration of Big Data collections in different infrastructures and the specification of spatio-temporal query and visualisation languages to retrieve multimedia and multiform data from distributed services. He has addressed the processing of data streams in heterogeneous networks and architectures to provide access, query, and analysis services adaptable to the execution context while maximising computational resources. This research has been carried out thanks to research projects funded by national bodies such as CONACyT, CUDI, ECOS-ANUIES and the VIPE of the UDLAP; and by international bodies like the FP7 framework programme, the Microsoft LACCIR laboratory, the STICAMSUD programme of France.

Javier A. Espinosa-Oviedo (<https://www.espinosa-oviedo.com>) is associate professor of computer science at the CPE engineering school, University of Lyon, and a member of the database group of the LIRIS-CNRS research laboratory. He obtained his PhD in Computer Science from the University of Grenoble in 2013, and his master and bachelor's degree in Computer Science and Computer Systems Engineering, from UDLAP, in Mexico, in 2006 and 2008, respectively. His experience concerns cloud and service-based Big Data management and processing. He has participated in projects (FP7, Horizon 2020, ANR, technology transfer) addressing challenges related to data centric systems applied to urban

computing, built environments, data visualization, social sciences, digital humanities, and e-health. He is fellow of the Mexican Academy of Computer Science (AMEXCOMP) and member of the ACM.

Luis M. Vilches-Blázquez is currently a Research Professor at the Computing Research Center, Instituto Politécnico Nacional (Mexico). His research interests focus on information integration, Linked Data, knowledge graphs, ontological engineering, and geospatial information/data. He co-authored over 80 research papers in conferences, workshops, and journals. He has participated in European projects (Towntology, DIGMAP, DynCoopNet, etc.), Latin American projects (IDEDES and Scenarios for the analysis of new trends in IDE in Latinamerica, etc.) and National projects in Spain (Geobuddies, WEBn+1, Autores 3.0, España Virtual, Ciudad2020, myBigData, etc.). He has also been an active member of the Spanish SDI Working Group and member of AENOR's AEN/CTN148 Technical Committee on Geographic Information and participated in various Working Groups in the context of ISO/TC 211 and the development of the ISO 19150 standard. In addition, he is an active member of the OGC and the IPGH, as well as of multiple programme committees in Conferences and Workshops at the international level and has given numerous invited talks and workshops.

Received: January 20, 2022; Accepted: November 25, 2022.

Matching Business Process Behavior with Encoding Techniques via Meta-Learning: An anomaly detection study^{*}

Gabriel Marques Tavares¹ and Sylvio Barbon Junior²

¹ Università degli Studi di Milano (UNIMI), Milan, Italy
gabriel.tavares@unimi.it

² Università degli Studi di Trieste (UniTS), Trieste, Italy
sylvio.barbonjunior@dia.units.it

Abstract. Recording anomalous traces in business processes diminishes an event log’s quality. The abnormalities may represent bad execution, security issues, or deviant behavior. Focusing on mitigating this phenomenon, organizations spend efforts to detect anomalous traces in their business processes to save resources and improve process execution. However, in many real-world environments, reference models are unavailable, requiring expert assistance and increasing costs. The considerable number of techniques and reduced availability of experts pose an additional challenge for particular scenarios. In this work, we combine the representational power of encoding with a Meta-learning strategy to enhance the detection of anomalous traces in event logs towards fitting the best discriminative capability between common and irregular traces. Our approach creates an event log profile and recommends the most suitable encoding technique to increase the anomaly detection performance. We used eight encoding techniques from different families, 80 log descriptors, 168 event logs, and six anomaly types for experiments. Results indicate that event log characteristics influence the representational capability of encodings. Moreover, we investigate the process behavior’s influence for choosing the suitable encoding technique, demonstrating that traditional process mining analysis can be leveraged when matched with intelligent decision support approaches.

Keywords: Anomaly detection, Meta-learning, Encoding, Process mining, Recommendation.

1. Introduction

Organizations rely on the correct execution of business processes to achieve their goals. However, anomalous instances in event logs are harmful to process quality. This way, stakeholders are interested in detecting and mitigating anomalies so that business processes correspond to their expected behavior. Detecting anomalies is not only beneficial for resource-saving but also to avoid security issues [46]. Process Mining (PM) is devoted to extracting valuable information from organizational business processes. Within PM, conformance checking methods are dedicated to finding anomalies. Conformance

^{*} This article is an extended version of the paper “Process Mining Encoding via Meta-Learning for an Enhanced Anomaly Detection” [41].

techniques compare process models and event logs, quantifying deviations and, consequently, identifying anomalies [39]. Traces not complying with the model are interpreted as anomalous, either from a control-flow or data-flow perspective.

Although conformance checking supports the recognition of anomalous traces, the methods are model-dependent, hindering their applicability since the model is not available in many scenarios. Moreover, resorting to expert knowledge is time-consuming and may often result in subjective and approximate decisions that do not allow the effective automation of PM-related tasks [30]. Various approaches have been proposed for detecting anomalies in business processes as a compliance verification task. For instance, Bohmer and Rinderle-Ma [10] use likelihood graphs to model process behavior and support anomaly detection. The method is applicable to control- and data-flow attributes, although the quality of discovered models limits its performance. Recently, many methods are relying on Machine Learning (ML). To overcome the mismatch at the representational level between PM and traditional data mining [16], a shared characteristic of ML-based approaches is to transform the process or trace representations. The feature vector obtained is capable of describing instances and connections for further ML modelling.

As pointed by Barbon et al. [6], a suitable encoding technique is crucial for the quality of posterior methods applied to the event log. This way, by finding the appropriate encoding, one can improve the identification of anomalous instances. In other words, a suitable encoding technique allows the best representation of typical behavior and adjusted discriminant capacity of anomalous traces. However, considering the multitude of available encoding methods, selecting the correct algorithm is challenging [24,38]. To grasp the algorithm selection problem in the context of PM, we propose an approach based on a Meta-learning (MtL) strategy to recommend the best encoding method for a given event log. The approach aims at maximizing the number of identified anomalies by profiling event log behavior based on its features. MtL has been applied as a recommender system, succeeding in emulating expert decisions for a wide range of applications [23,47]. Taking advantage of structural and statistical lightweight meta-features extracted from event logs, our MtL approach suggests the most suitable encoding technique. For that, it was built using 80 meta-features, trained over 168 event logs (meta-instances) for guiding the best one of eight promising encoding methods (meta-target). Moreover, the proposed approach is easily scalable, allowing the inclusion of additional encoding techniques and log descriptors. Results show that the MtL approach outperforms current baselines and can improve the detection of anomalous traces independently of the applied learning algorithm.

This work is an extension of [41]. The reference work investigated the MtL approach and compared its performance with baselines. In this extension, we propose a more in-depth analysis of the results and their insights, also including more encoding techniques. First, by adopting explainability techniques, we strengthen the understanding of process behavior's impact on the encoding quality. For that, we present an analysis connecting event log features to encoding methods, hence, providing a better problem understanding. Furthermore, we developed experiments to test the internal validity of our research design. The contributions of the presented approach are manifold. The main advantage is providing guided recommendations of the suitable encoding technique for a given event log, which is beneficial for both experts and non-experienced users. For end-users, the recommendation saves experimentation time and speeds up the pipeline design. For ex-

perts, the created meta-database serves as an analytical tool to correlate business process behavior with optimal techniques, giving further insight into the problem.

The remainder of this paper is organized as follows. Section 2 presents an extensive discussion about anomaly detection and the encoding of business processes. Section 3 lays down fundamental aspects regarding the application and where this problem sits in relation to the literature. Section 4 presents the MTL-based methodology proposed in this paper. Moreover, the section explores the extracted meta-features, and the encoding techniques used along with the MTL approach. Section 5 details the experimental setup (including event logs) and compares our approaches's performance with two baselines. Furthermore, a comprehensive anomaly analysis is shown along with a discussion about implications of the different anomalies. Lastly, Section 6 concludes the paper and highlights our contributions.

2. Related Work

This section explores the discussion in literature focusing on anomaly detection and encoding techniques. Since there are no works identifying the best encoding techniques for anomaly detection, the goal is to present how traditional anomaly detection is performed within PM and compare it with ML-based anomaly detection.

2.1. Anomaly Detection

Early on, Bezerra et al. [9] defined anomaly in business processes based on a set of assumptions: (i) the set of traces can be divided in normal and anomalous subsets, (ii) an anomalous execution is infrequent in comparison to normal executions, and (iii) normal behavior generates more comprehensive process models than models generated from anomalous traces. Within business process literature, traditional anomaly detection is performed by conformance checking techniques [9,8]. For that, the conformance approach is based on the comparison between process model and event log [1]. Therefore, non-compliant business instances are interpreted as anomalous while conforming ones are regarded as normal behavior. It is important to note that there are several different conformance checking methods and no consensus of which one is the best [26].

Most traditional techniques rely on token-based replay [9,39,3]. These approaches replay trace sequences into the model by consuming executed activities according to model constraints. Trace fitness is measured by counting missing and remaining tokens. For instance, Bezerra et al. [9] proposed a method that uses domain knowledge to filter the event log and then applied a process discovery technique to generate a model from the filtered log. Then, traces are classified based on model fitting (i.e., non-fitting traces are considered anomalous). However, more recent techniques rely on alignments due to being more robust, especially for logs with noise [26]. These techniques also propose a model-log comparison but directly relate a trace to valid execution sequences allowed by the model. Therefore, alignment methods apply a synchronous approach where normal behavior is defined by the accordance of moves between trace and model. Finally, multiple alignments can be produced, and the goal is to find the optimal one, which can be costly. Although used in industry, conformance checking techniques depend on a high-quality

process model, which is a constraint in many real scenarios. This happens because often models are unavailable, inadequate, or incorrect [5]. Moreover, creating or inferring them from data is complex as the discovery process has to balance between precision and generality [40].

To overcome the issue of identifying anomalous instances in scenarios without an available process model, ML-based techniques have been gaining traction in the literature. Nolle et al. [33] use an autoencoder to model process behavior and detect irregular cases. The same authors in another research use a deep neural network trained to predict the next event [34]. An activity with a low probability score (extracted from the network) is recognized as an anomaly. The paper of Tavares and Barbon ([40]) use language-inspired trace representations to model process behavior. Cases isolated in the feature space are identified as abnormal. Techniques based in deep learning such as [33,34] may be computationally expensive, while traditional ML techniques inject bias when representing traces in high-dimensional spaces.

2.2. Encoding Business Processes

A common characteristic of ML-based approaches is the need to transform the event log representation into formats expected by traditional ML techniques. In other words, one may need to apply a function that projects each trace to a n -dimensional feature space where the anomaly detection can take place. This transformation step is often referred to as encoding.

Trace encoding is often applied [11,17,18,27,44] in PM but has been shallowly discussed in the literature [6]. Due to a mismatch at the representational level, it is mandatory to apply trace encoding when applying data mining techniques to event logs. In the context of predictive models, Leontjeva et al. [27] proposed a sequence encoder based on Hidden Markov Models, whereas Polato et al. [37] used a last state encoding method. Word embedding techniques have also been applied [17]. Koninck et al. [17,40,22] experimented with word2vec and doc2vec to encode traces as words and paragraphs, respectively. Furthermore, the authors also extrapolated these encodings to other representational levels, such as logs and models. Hake et al. [22] used the same word2vec and combined it with recurrent neural networks to label nodes in business models.

Barbon et al. [6] identified three major encoding families: PM-based, word embeddings, and graph embeddings. PM-inspired encoding assumes that measures extracted from conformance checking techniques can be used as features and, hence, as an encoding technique. Word embeddings are traditionally associated with natural language processing and information retrieval. When applied to business processes, these techniques assume that activity names are words and sequence of activities are sentences. This interpretation allows the application of both shallow techniques such as one-hot encoding and modern approaches such as word2vec. It follows that activities and their direct-follows relationships can also be interpreted as nodes and edges in a graph. Naturally, business process models are also represented as graphs, which matches graph-based techniques. Therefore, graph embeddings, which follow up on word embeddings, are also applicable for trace encoding. In this scope, the authors assessed complementary perspectives in trace encoding techniques considering the complexity, time consumption and injected bias, among others. Overall, the authors found that there is no best encoding technique for every scenario (i.e., different event logs may be better encoded by different techniques).

2.3. Meta-learning in Process Mining

Recently, automation approaches based on MtL have been explored in PM [43,42]. The main hypothesis behind these works assumes a relationship between process behavior and optimal pipelines. Barbon et al. [43] use MtL to identify the discovery technique that maximizes performance in multiple criteria. Thus, given a new event log, its behavior is retrieved through descriptive features, which are then associated with the best discovery technique. A more complex issue is investigated in [42]. In this work, the authors propose a technique to recommend the complete pipeline for trace clustering. The pipeline includes the encoding technique, clustering algorithm, and its hyperparameters, and the step's intercorrelation makes the problem more challenging. In both works, experiments have surpassed baseline performances, indicating that there is indeed a relationship between process behavior and PM pipelines. In the same fashion, we extrapolate this assumption for encoding techniques. The aim is to improve ML-based anomaly detection by identifying the suitable encoding technique.

3. Problem Statement

Considering the multiple algorithms that encode event logs, choosing the best method is a challenging task even for experts. Furthermore, in many scenarios, not enough attention is given to this step in the PM pipeline. In this section, we present basic notions and state where the problem sits in the literature.

Definition 1 (Event, Attribute, Case, Event log). Let Σ be the event universe, i.e. the set of all possible event identifiers. Σ^* denotes the set of all sequences over Σ . Let \mathcal{AN} be the set of attribute names. For any event $e \in \Sigma$ and an attribute $a \in \mathcal{AN}$, then $\#_a(e)$ is the value of attribute n for event e . Let C be the case universe, that is, the set of all possible identifiers of a business case execution. C is the domain of an attribute case $\in \mathcal{AN}$. An event log L can be viewed as a set of cases $L \subseteq \Sigma^*$ where each event appears only once in the log.

Definition 2 (Encoding). Let an event log L , encoding is a function f_e that maps L to a feature space, i.e., $f_e : L \rightarrow \mathcal{R}^n$ where \mathcal{R}^n is a n -dimensional real vector space.

Encoding event logs bridges the gap between PM and data mining. That is, once the different log granularity levels are condensed into an n -dimensional numerical feature space, the combination with traditional data mining techniques is natural.

Definition 3 (Algorithm Selection Problem). Let $x \in \mathcal{P}$ be a problem in a problem space, let $f(x) \in \mathcal{F}$ be a function that extracts features from the problem x , let $S(f(x))$ be a function that selects the mapping between the problem space to the algorithm space $A \in \mathcal{A}$, and let $p(A, x)$ be a function that maps the performance of an algorithm to the performance measure space. The goal is to determine $S(f(x))$ (the mapping of problems to algorithms) that maximizes the performance of the algorithm.

In this context, MtL is a strategy to solve the algorithm selection problem. For that, meta-learning aims at mapping the relationship between the problem space and the algorithm performance space. Automatically selecting an encoding technique is, then, beneficial for the end-user. Moreover, it could also provide insights into event log behavior and optimal encoding methods.

4. Methodology

This section presents the proposed methodology to enhance anomaly detection in event logs. The method is based on the combination of encoding representational power with MtL as the learning paradigm. We also present the design details, including all steps of the MtL pipeline.

4.1. Meta-Learning for Anomaly Detection in Process Mining

In this work, we investigate encoding methods that boost the performance of traditional ML algorithms for the anomaly detection task in business processes. For that, our proposed approach relies on MtL. The primary assumption is that the event log characteristics (i.e., descriptors) can support the choice of the best encoding method. The best encoding is the technique that produces the highest anomaly detection rates (i.e., accuracy) when combined with a given ML-induced model. The boosted capability provided by a particular encoding method relies on the correctly and effectively discriminative capacity to represent traces [6]. However, identifying the best encoding is very tricky depending on the expert's experience in the particular domain. Here, we follow the assumption that this experience could be emulated using MtL.

Figure 1 presents an overview of our approach. Starting from a collection of event logs, the first step is the *Meta-feature extraction*, which mines descriptors that characterize the event logs. The extracted meta-features are capable of capturing the business process behavior from complementary perspectives. The idea is that the combination of multiple descriptors creates a representation of log behavior in a vector space. Next, we submit the event logs to encoding techniques. The encoding methods work at the trace level, and the encoded traces serve as input for an ML algorithm aiming to detect anomalies. Hence, we assess a performance metric (namely, F-score) that ranks the encoding algorithms for each event log. This step is called *Meta-target definition*, where each event log is submitted to all encoding methods, and the best encoding (meta-target) is identified by ranking performances. Then, the *Meta-database creation* joins the two previous steps. A database is created using meta-features (descriptors) and the meta-targets (best encoding for a given process) extracted from the logs. Consequently, each meta-instance is a set of log descriptors associated with an encoding technique that leverages anomaly detection performance for that event log. Once the meta-database is created, we induce a *Meta-model* in the *Meta-learner* step. The meta-model learns the distribution of the process data, hence, it associates process behavior to encoding techniques. This way, the meta-model is the final product of our workflow. Given a new event log, its meta-features are extracted and fed to the meta-model, which provides a data-driven recommendation of the best encoding technique for that event log.

4.2. Log descriptors – Meta-Feature Extraction

Extracting high-quality descriptors is fundamental for the performance of the meta-model. Moreover, meta-feature extraction should have a low computational cost, otherwise, the MtL pipeline is unjustified. This way, we selected a group of lightweight features that contains reliable representational capacities. To retrieve a multi-perspective view of event logs, we extract features from several process layers: activities, traces, and logs. These

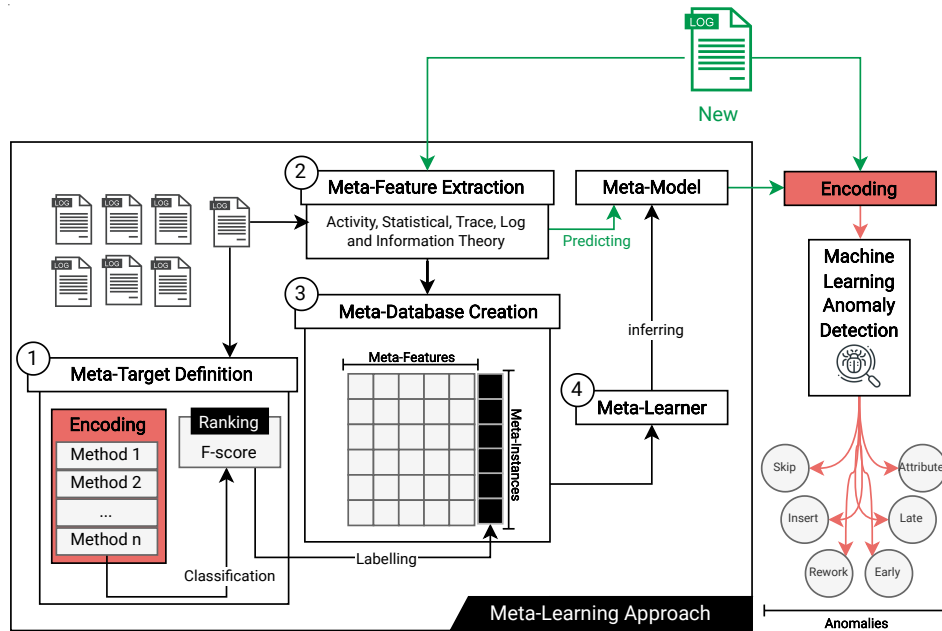


Fig. 1. Overview of the proposed approach for recommending encoding techniques based on process behavior. First, meta-features are extracted from a collection of event logs. Then, by applying each encoding technique in conjunction with a ML classification model, it is possible to rank their performance and find the best encoding method. By joining the two previous steps, we construct a meta-database, which in turn serves as the basis to infer a meta-model aiming at recommending an encoding technique for a new (unseen) event log

features were first proposed in [43], which combines business process features from different sources. Table 1 presents the features and groups they belong.

Overall, 80 features were extracted from the event logs. They capture complementary elements of business processes, containing information such as statistical dispersion, probability distribution shape and tendency, and log complexity.

4.3. Encodings – Meta-Target Definition

In this work, the application of encoding for anomaly detection in PM is a fundamental step toward building the meta-database. Ultimately, the encodings are the meta-targets associated with log features. Given a log and its meta-features, we associate it with an encoding that maximizes the anomaly detection performance. Therefore, encoding techniques play a major role as they can excel in detecting anomalous instances for certain types of log behaviors. The application of encoding in PM has already been explored by several researches [6,17,27,37]. Barbon et al. [6] extensively evaluated trace encoding methods using complexity metrics to assess encoding capacity. Moreover, the authors submit the encoding methods to a classification task for anomaly detection. The

Table 1. Meta-features extracted from event logs. This set of meta-features aim at capturing several complementary levels of business process' behaviors

Family	# Number	Name
all activities	12	number of activities, minimum, maximum, mean, median, standard deviation, variance, the 25th and 75th percentile of data, interquartile range, skewness, and kurtosis coefficients
start activities	12	number of activities, minimum, maximum, mean, median, standard deviation, variance, the 25th and 75th percentile of data, interquartile range, skewness, and kurtosis coefficients
end activities	12	number of activities, minimum, maximum, mean, median, standard deviation, variance, the 25th and 75th percentile of data, interquartile range, skewness, and kurtosis coefficients
trace lenght	29	minimum, maximum, mean, median, mode, standard deviation, variance, the 25th and 75th percentile of data, interquartile range, geometric mean and standard variation, harmonic mean, coefficient of variation, entropy, and a histogram of 10 bins along with its skewness and kurtosis coefficients
trace variants	11	mean, standard variation, skewness coefficient, kurtosis coefficient, the ratio of the most common variant to the number of traces, and ratios of the top 1%, 5%, 10%, 20%, 50% and 75% to the total number of traces
log level	4	number of traces, unique traces, traces ratio and the number of events

work proposes the application of three encoding families to event logs: PM-based encoding, word embedding, and graph embedding. The PM-based encodings are conformance checking techniques that compare an event log to a process model, measuring deviance and producing a fitness value along with token counting results [39]. Word embeddings can naturally be applied in event logs when considering activities and traces as words and sentences [5,17]. These techniques rely on context information captured by neural networks' weights trained for context prediction. Lastly, graph embeddings are techniques that encode graph information, such as nodes, vertices, and their attributes. Graph embeddings are particularly interesting in the PM domain as they can represent process models (with limitations such as not capturing concurrency) and traces, modeling entity links, and long-term relations.

Considering the three encoding families presented in [6], we selected representatives of each encoding family. This way, we aim to reduce the representative bias and evaluate if there is a relation between encoding techniques and log behavior. For PM-based encodings, we used alignments and token-based replay as they have been considered the state-of-the-art conformance checking method [15]. Alignments compare the event log and process model and measure the deviations between the two. For that, it relates traces to valid execution sequences allowed by the model. This evaluation unfolds into three types. Synchronous moves are observed when both the trace and model can originate a move. Model-dependent moves originate only from the model, and log-dependent moves are derived from traces but are not allowed by the model. Synchronous moves represent the expected behavior when model and log executions agree. The alignment technique

searches for an optimal alignment (i.e., when the fewest number of the model- and log-moves are necessary). This process, measured by a cost function, produces a fitness value and other statistics regarding the states consumed by the model. Token-replay follows a similar approach by comparing log and model. For that, it replays traces in the model by consuming executed activities according to model constraints. A conformance measure is produced based on missing and remaining activities.

Word embedding techniques in PM have mostly relied upon *word2vec* and *doc2vec* to encode traces. In Barbon et al. [5], the authors propose the *word2vec* encoding in conjunction with one-class classification to detect anomalies in business processes. Koninck et al. [17] use both *word2vec* and *doc2vec* to encode activity and trace information, respectively. Subsequent work also builds trace representations based on *doc2vec* [28]. In this work, we adopt several text-based encodings: *count2vec*, *doc2vec*, *hash2vec*, one-hot and *word2vec*. The count vectorizer (*count2vec*) encodes words by accounting their frequencies in a text document, producing a matrix of word counts. The one-hot encoding produces a similar matrix but binarizing the values, i.e., accounting for the appearance or not of a word in a document. For both techniques, feature length is determined by the number of unique words in the vocabulary, which tends to produce sparse vectors. The hash vectorizer (*hash2vec*) maps a feature with a word using a hashing function and computes frequencies based on previously mapped indices. Although *hash2vec* allows for predetermined vector sizes, hash collisions may happen. *Doc2vec* is an extension of *word2vec* adapted to documents, independently of their length. *Word2vec* creates numerical representations for words and, for that, a neural network is trained to reconstruct the linguistic context of words in a corpus [32]. The word embeddings come from the weights of the induced neural network. The main advantage is that words appearing in similar contexts produce similar encoding vectors. However, this method is limited to unique word representations. *Doc2vec* extends *word2vec* by adding a paragraph vector in the encoding process [25]. This way, the document context is captured by the encoding.

For the graph embedding family, we employed *node2vec*, another encoding technique built on top of *word2vec*. *Node2vec*'s primary goal is to encode graph data while maintaining graph structure. Given a graph, *node2vec* performs random walks starting from different nodes [21]. This process creates a corpus, which is used as input for *word2vec*. The second-order random walks balance a trade-off between breadth and width, capturing neighbor and neighborhood information. Hence, the method can represent complex neighborhoods given its node exploration approach.

Using the eight defined encodings (*alignment*, *count2vec*, *doc2vec*, *hash2vec*, *node2vec*, one-hot, token-replay and *word2vec*), we performed the *Meta-target definition* step shown in Figure 1. The goal is to identify which encoding enhances the detection of anomalous traces in event logs. For that, we applied a traditional ML pipeline where we combine an encoding technique with a classification algorithm for detecting anomalous traces. This way, each log is first encoded, then, its traces are divided into an 80%/20% holdout strategy where 80% of traces are used for model inferring while 20% of traces are used for testing (i.e., evaluating if the trained model can correctly label traces). This process is repeated 30 times with different splits to avoid outlier performances and biased splits. We employed the Random Forest (RF) algorithm [12] in the meta-target definition step due to its robustness and easiness of interpretability. After 30 iterations, the average F-score for each technique is obtained, allowing us to rank the encoding techniques based

on their average performance. Thus, the best encoding technique for a given log is the one that produces the highest average F-score when combined with an ML technique for the anomaly detection task. We chose F-score as the ranking metric as it successfully balances different performance perspectives.

As an example of the ranking step, consider event log L , encoding techniques E_1 , E_2 and E_3 , and a classification model M . We submit L to E_1 and combine it with M to detect anomalous traces. Following the described steps, this process is repeated multiple times and the average F-score of M is retrieved. The same is done for E_2 and E_3 . Since the model is the same, the only variable influencing the final performances is the encoding method. Let 0.75, 0.6 and 0.85 be the average F-score for E_1 , E_2 and E_3 , respectively. Therefore, the encoding technique that enhances anomaly detection in this scenario is E_3 because it produces the highest average F-score when combined with M . This way, the meta-target definition associates the meta-target (E_3) with its meta-instance (L), as shown in Figure 1.

4.4. Meta-database and Meta-model

We create the meta-database by combining the meta-features extracted from the logs with the defined meta-targets. Once the meta-database is built, the meta-learner step takes place. The meta-learner embeds a traditional ML pipeline, that is, the meta-database is submitted to an ML algorithm that infers a meta-model. The meta-model is the final object produced by the MtL approach. This way, given a new event log (previously unseen), its meta-features are extracted and, based on them, the meta-model is able to recommend a suitable encoding technique. Furthermore, the meta-database is also an interesting byproduct because it provides a mapping between log behavior (meta-features) and optimal encoding techniques (meta-targets). Thus, it can be used to analyze the relationship between the problem space and the performance space.

5. Evaluation

In this section, we present the performance of our approach and compare it with two baseline methods. Moreover, we develop a discussion regarding the impact of the anomalies in the encoding.

5.1. Experimental Setup

This section describes the main aspects of the experimental configuration, such as the used event logs and the algorithm applied in the meta-learner step.

Event logs – Meta-instances The more instances available, the more representative is the meta-database as it contains more examples of business process behaviors. Experiments on anomaly detection benefit from labeled datasets since one can compute traditional performance metrics to evaluate if anomalous cases are indeed captured. Considering these constraints, we built our meta-database from two groups of synthetic event logs composed of a wide range of behaviors originating from 12 different process models and

disturbed by six types of anomalies. We highlight that the use of real event logs is not possible in this scenario as the performance assessment relies on labeled event logs at the trace level. That is, information regarding normality or abnormality of traces is needed. Therefore, since we suffer from the availability of benchmark data and labeled event logs, we chose to proceed with synthetic event logs where we can assert which traces are normal or anomalous. Thus, avoiding bias in the interpretation of results.

The first group of event logs was initially presented by Nolle et al. [34] and replicated in [5] and [40]. Six models were generated using the PLG2 tool [14]. PLG2 randomly generates process models representing several business patterns such as sequential, parallel, and iterative control-flows. Moreover, PLG2 allows the configuration of the number of activities, breadth and width, hence, providing a complex set of models that capture diverse behavior. One additional process model, P2P, extracted from [35] was added to the pool. Then, the authors adopted the concept of likelihood graphs [10] to introduce long-term control-flow dependencies. The likelihood graphs can mimic complex relations between event to event transitions and attributes attached to these events. This way, the control-flow perspective is constrained by probability distributions that coordinate the model simulation. For instance, an activity may follow another given a probability. Combining stochastic distributions with a set of process models leverages the similarity between produced event logs and real-world logs. Four event logs were simulated in each process model, generating a total of 28 logs. The final step added anomalies to the traces within the synthetic event logs, which is a traditional practice in related work [9,10]. We applied six anomaly types for all event logs with a 30% incidence: i) *skip*: a sequence of 3 or less necessary events is skipped; ii) *insert*: 3 or less random activities are inserted in the case; iii) *rework*: a sequence of 3 or less necessary events is executed twice; iv) *early*: a sequence of 2 or fewer events executed too early, which is then skipped later in the case; v) *late*: a sequence of 2 or fewer events executed too late; vi) *attribute*: an incorrect attribute value is set in 3 or fewer events.

The second group of synthetic event logs was proposed by Barbon et al. [6]. The authors also used the PLG2 tool to create five process models representing scenarios of increasing complexity (i.e., a higher number of activities and gateways). Then, the process models were simulated using the *Perform a simple simulation of a (stochastic) Petri net ProM plug-in*³, producing 1000 cases for each log. As a post-processing step, the same anomalies used for the previous set of logs were applied in this set but with different configurations. The authors implemented four anomaly incidences (5%, 10%, 15% and 20%). Moreover, the dataset contains binary and multi-class event logs, meaning that some logs incorporate normal behavior and only one anomaly type (binary), and some logs contain both normal behavior and all anomalies at the same time (multi-class). The latter configuration is especially challenging given the higher complexity as more behaviors are present in the same log. In total, this set contains 140 event logs.

For all event logs, anomalies sit on the event level, but they can be easily converted to the case level. That is, cases containing events affected by any anomaly are considered anomalous cases. Table 2 shows the event log statistics for all event logs used in this work. As demonstrated, the set of logs presents a significant behavioral variation because they contain several different anomalies (combined in binary and multi-class scenarios),

³ <http://www.promtools.org/doku.php>

injection rates, and process characteristics. These characteristics support the creation of a heterogeneous meta-database, increasing business process representability.

Table 2. Event log statistics: each log contains different levels of complexity

Name	#Logs	#Cases	#Events	#Activities	Trace length	#Variants
P2P	4	5k	38k-43k	25	5-14	513-655
Small	4	5k	43k-46k	39	5-13	532-702
Medium	4	5k	28k-31k	63	1-11	617-726
Large	4	5k	51k-57k	83	8-15	863-1143
Huge	4	5k	36k-43k	107	3-14	754-894
Gigantic	4	5k	28k-32k	150-155	1-14	693-908
Wide	4	5k	29k-31k	56-67	3-10	538-674
Scenario1	28	1k	10k-11k	22-380	6-16	426-596
Scenario2	28	1k	26k	41-333	23-30	1k
Scenario3	28	1k	43k-44k	64-348	39-50	1k
Scenario4	28	1k	11k-13k	83-377	1-30	383-536
Scenario5	28	1k	18k-19k	103-406	1-37	637-737

To avoid concerns regarding synthetic data representativeness, we further assess the distribution of log behavior compared to real event logs. For that, we extracted all meta-features (listed in Table 1) of the synthetic event logs plus three real business processes (Business Process Intelligence Challenges (BPIC) 2012⁴, BPIC 2013 and helpdesk⁵). Then, we applied a dimensionality reduction technique, namely, the Principal Component Analysis (PCA) algorithm [45], to visualize the event logs in the feature space. Figure 2 presents the results of the feature space reduced to two dimensions and populated by the event logs. The union of the two Principal Components (PC) explains 92.30% of the data variance. Therefore, most distances in the higher dimension space are respected after the dimensionality reduction. Figure 2 shows that the real event logs sit close to the synthetic event logs in the feature space. Therefore, although no real event logs are employed in the experiments (due to the lack of labels), this analysis indicates that the synthetic event logs are representative of real scenarios.

A complementary mean of assessing data behavior is by extracting the log complexity. For that, we retrieved the complexities of all synthetic event logs and the same three real processes. The measure chosen was the normalized variant entropy, recently proposed by Augusto et al. [4], and is based on graph entropy. According to the authors, graph-based entropy is particularly suited for event logs as it captures size, variation and distance in an integral way. Figure 3 presents the normalized variant entropy distribution in the form of a boxplot. The three red dots refer to the real event logs. As we can see, groups of logs spread across the x-axis. The complexity score for real event logs is similar to the synthetic logs, reaffirming the suitability of the synthetic business processes.

⁴ <https://www.tf-pm.org/resources/logs>

⁵ <https://doi.org/10.17632/39bp3vv62t.1>

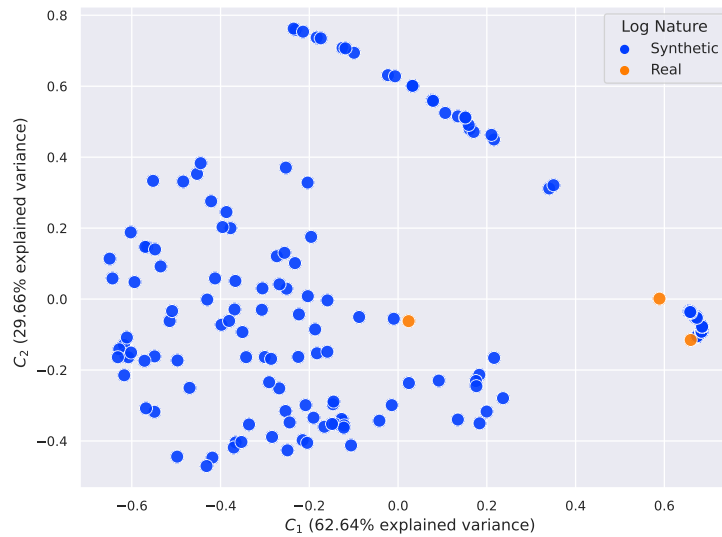


Fig. 2. Reduced feature space after applying PCA. Meta-features capturing log behavior are extracted from both real and synthetic event data. The real event logs populate the same region in the feature space as the synthetic ones

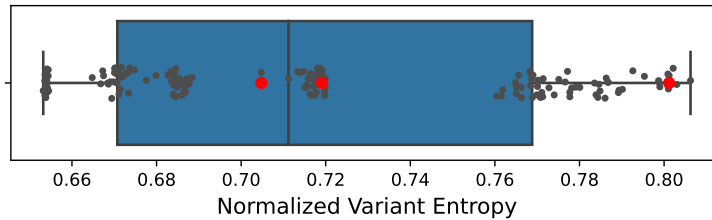


Fig. 3. Boxplot representing the normalized variant entropy extracted from all event logs. Red dots are the three real event logs

Meta-model The last step of the methodology is inferring the meta-model from the meta-data (as seen in Figure 1). In this scenario, we transform the algorithm selection problem into a classification problem. This way, we enable the use of traditional ML classifiers for this task. The meta-model creation is a step related to the usage of a supervised ML algorithm to map meta-features as an inference model. Our proposal was fashioned as a framework, allowing expansions and recombination of algorithms following the constraints established. Considering consistency and the same arguments presented in Section 4.3, we also employ the RF in this step. In particular, a RF model can provide insights into the quality of the features and their contribution to the prediction procedure. A RF model is composed of decision trees as base learners, which are built using the most informative features, reducing the number of features such as a feature selection step. Another impor-

tant aspect is the high predictive power, a simple tuning procedure, and also the reduced possibility of generating an overfitted model. Furthermore, we clarify that deep learning methods are not suitable for tabular data and, hence, cannot be used in this context. To measure the performance of our approach (explored in Section 5.2), we use a traditional configuration of ML pipelines. For that, we divide the meta-instances into two sets: 80% are used for model training while the remaining 20% for performance measurement. This holdout strategy is repeated 30 times (with different data splits) to avoid outlier performances. We assess both F-score and accuracy and report the results in comparison with two baselines.

5.2. Results and Discussion

This section reports the MtL recommendation results. Moreover, it provides an in-depth exploration of anomalous scenarios and their relationship with encoding techniques. Finally, we assess the relevant features used by the meta-model and test the validity of the experimental design.

Meta-learning Performance First, we report the results of the meta-target definition step (i.e., ranking the encodings for each meta-instance). Since we are following a data-driven strategy, it is worth observing the balance regarding meta-targets. For that, Figure 4a shows the frequency of the encoding techniques in the first position in the ranking step (see Section 4.3). The ranking is built using the average F-scores obtained by each encoding algorithm. This analysis brings insights about the balanced scenario when selecting an encoding technique, illustrating the “no free lunch theorem” [2]. Four encoding techniques appeared most frequently in the first position. The alignment method was the best encoding for 42 event logs, while doc2vec was optimal for 35 logs, token-replay for 30 logs and node2vec for 27 logs. In other words, Figure 4a shows that alignment was the meta-target for 42 meta-instances, doc2vec was the meta-target for 35 meta-instances, and token-replay and node2vec were the meta-target for 30 and 27 meta-instances, respectively. These results highlight a balanced distribution between the best-ranked encodings. Regarding the other encodings, one-hot and count2vec were the least frequent best encodings. This outcome is expected as these encodings are very shallow, produce sparse vectors, and do not capture process constraints, such as loops. Word2vec also performed poorly mostly because of the small vocabulary of event logs. This technique might require more examples to produce its best outputs. Finally, hash2vec did not perform so well due to collisions in the mapping space, which decrease the encoding quality and, hence, harm the anomaly detection performance.

Figure 4b reports the performance of our approach for the task of recommending the encoding technique that leverages anomaly detection in event logs. Considering the lack of literature in the area, we compare the MtL performance with two baselines: majority and random selection. Majority regards the encoding method with the highest frequency in the meta-database, hence always recommending the alignment encoding. Although a simple baseline, majority voting is a suitable comparison in ML applications, clearly specifying the minimum performance threshold. Random selection works by arbitrarily selecting one of the possible meta-targets for each event log. This approach simulates a PM practitioner in a scenario without the availability of experts, a common situation in

real environments. From both accuracy and F-score perspectives, our approach outperforms the others with a large advantage. The meta-model obtains an average accuracy of 55% and an average F-score of 0.43. The violin visualization also demonstrates the MtL robustness since the density curve is compressed (i.e., most recordings are near the average mark). The majority approach produced accuracy and F-score averages of 24% and 0.05, respectively. The random method achieves 13% accuracy and 0.11 F-score. It is worth mentioning that these results report the performance for selecting the best encoding method.

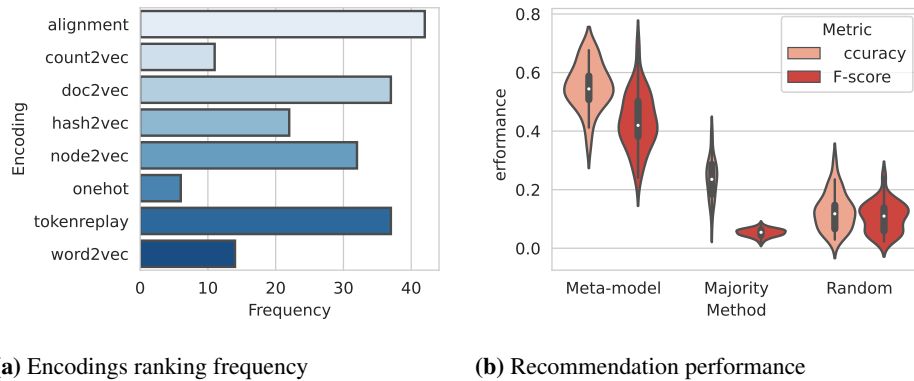


Fig. 4. Encoding ranking extracted from the Meta-target definition step. The ranking is ruled by the F-score obtained by recommending a suitable encoding technique. Given a new event log, the meta-model recommends the best encoding considering the meta-features derived from the log. Fig. 4b demonstrates the meta-model performance in comparison with baseline approaches

Anomaly analysis As introduced in Section 5.1, the event logs contain six different anomaly types. Moreover, a subset of the logs is struck by all six anomalies at the same time. Considering the anomaly perspective, Table 3 reports the F-score performance of all encodings and compares them with the MtL approach. Naturally, detecting anomalous instances in logs affected by *all* anomalies is the most difficult task. Hence, the F-score values are the worst in this scenario. At the same time, this scenario is the most common in real environments, where event logs contain traces with multiple deviation types. Nonetheless, we observe that MtL reports the highest mean F-score, reaching 0.49. It is followed by alignments (0.47), doc2vec (0.43), hash2vec (0.43) and node2vec (0.43). The performance rises considerably in the other anomaly types as the problem is binary in these cases.

Insert, *rework* and *skip* are the most detectable anomalies because they deeply affect the control-flow perspective of traces, therefore, this behavior change is easily captured by the encodings. For these anomalies, MtL reaches the highest performance values, producing F-score values close to 1. For *rework* and *skip* anomalies, hash2vec and node2vec

Table 3. Comparison of anomaly detection performance using fixed encoding methods and MtL recommendation. Mean and standard deviation (in parenthesis) F-score values are reported for each anomaly type. Bold values indicate the best method for each anomaly

Encoding	<i>all</i>	<i>attribute</i>	<i>early</i>	<i>insert</i>	<i>late</i>	<i>rework</i>	<i>skip</i>
alignment	0.47 (0.15)	0.92 (0.04)	0.93 (0.04)	0.98 (0.02)	0.93 (0.03)	0.97 (0.02)	0.98 (0.02)
count2vec	0.42 (0.15)	0.93 (0.04)	0.92 (0.04)	0.93 (0.03)	0.93 (0.04)	0.98 (0.01)	0.98 (0.01)
doc2vec	0.43 (0.21)	0.93 (0.03)	0.94 (0.03)	0.93 (0.03)	0.94 (0.03)	0.94 (0.03)	0.95 (0.03)
hash2vec	0.43 (0.13)	0.93 (0.04)	0.93 (0.04)	0.98 (0.01)	0.93 (0.04)	0.99 (0.01)	0.99 (0.01)
node2vec	0.43 (0.12)	0.93 (0.04)	0.92 (0.04)	0.98 (0.01)	0.93 (0.04)	0.99 (0.01)	0.99 (0.01)
onehot	0.31 (0.1)	0.93 (0.04)	0.92 (0.04)	0.93 (0.03)	0.93 (0.04)	0.93 (0.04)	0.98 (0.01)
token-replay	0.36 (0.08)	0.93 (0.03)	0.94 (0.03)	0.96 (0.02)	0.94 (0.03)	0.97 (0.02)	0.96 (0.02)
word2vec	0.4 (0.14)	0.93 (0.04)	0.92 (0.04)	0.98 (0.02)	0.93 (0.04)	0.98 (0.02)	0.98 (0.02)
MtL	0.49 (0.15)	0.93 (0.03)	0.95 (0.03)	0.99 (0.01)	0.94 (0.03)	0.99 (0.01)	0.99 (0.01)

tie with the MtL approach. Several other encodings follow closely, such as alignment, count2vec and word2vec. The encoding order changes when observing *early* and *late* anomalies. In these scenarios, MtL remains the best technique, reaching 0.95 F-score for *early* and 0.94 F-score for *late*, but now is followed more closely by doc2vec and token-replay, both reaching 0.94 in the two anomalies. Finally, all techniques (except alignments) tie for the *attribute* anomaly. Interpreting performance from the anomaly perspective reinforces the hypothesis that encodings perform differently in different scenarios, that is, log behavior is determinant when choosing the appropriate encoding. Hence, the MtL efficiency in this experiment exposes the influence of event log behavior on the encoding representational power. The results indicate that anomaly detection is enhanced when the relationship between event log descriptors and encodings is appropriately mapped. This mapping is mastered by our proposed MtL method, which outperforms the use of fixed encodings for all event logs.

We compared the F-score obtained by classifying all event logs using statistical analysis grounded on the non-parametric Friedman test to determine any significant differences between the usage of a unique encoding technique and meta-recommended ones. We used the post-hoc Nemenyi test to infer which differences are statistically significant [19]. As Figure 5 shows, differences between populations are significant, i.e., the MtL framework is statistically superior to other methods. Furthermore, other groups with no significant difference can be identified, e.g., alignment, node2vec, hash2vec and token-replay. One-hot encoding was statistically the worst performing encoding technique, separated from other groups and algorithms. Thus, MtL for recommending individual encoding methods to maximize the predictive performance achieved superior results statistically different from the usage of only one encoding. In other words, the performance obtained using MtL was statistically superior to a single encoding technique.

Meta-model Comparison To better assess meta-model performance, we evaluated the same recommendation problem using several traditional classifiers. The choice of classifiers used is based on the relevance in the ML literature and their different nature, therefore, possibly capturing if some heuristics influence in the recommendation quality. Along with RF, the classifiers are: Decision Tree (DT) [13], Logistic Regression (LR), Support

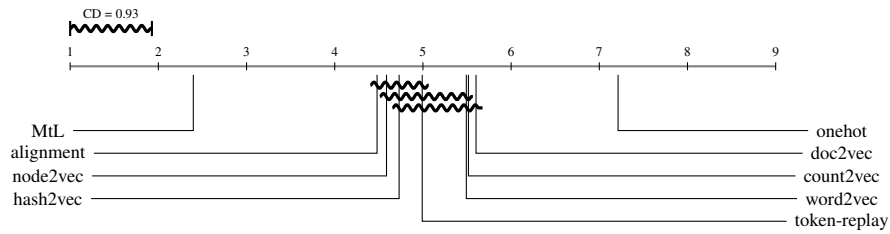


Fig. 5. Nemenyi post-hoc test (significance of $\alpha = 0.05$ and critical distance of 0.93) considering the F-score obtained from all event log classifications

Vector Machine (SVM) [36], k-Nearest Neighbor, and Gradient Boosting (GB) [20]. Figure 6 exposes the performances (accuracy and F-score) of all classification algorithms (in this case, meta-models). RF and GB appear as the best meta-models for both metrics, producing the highest average performances. In terms of F-score, SVM and kNN perform poorly with a significant distance to other methods. DT and LR remain in the middle group, not achieving the best performances, but producing a more stable recommendation performance than the worst algorithms.

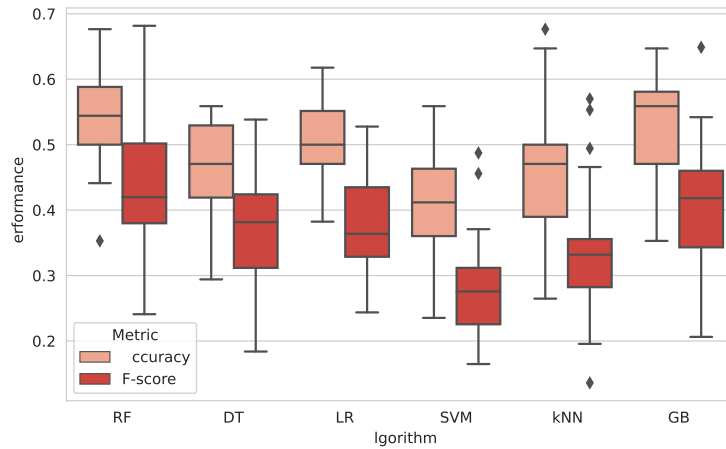


Fig. 6. Performance report for several classifiers in the recommendation task. RF and GB stand out as the most appropriate meta-models

To complement the previous analysis, we performed a statistical analysis again grounded on the non-parametric Friedman test. Figure 7 presents the results of the statistical test. As identified before, RF and GB are not statistically different within themselves. However, this group is superior to other algorithms as their distance is higher than the critical distance. The second group (GB, DT, LR and kNN) separates itself from SVM, which performs quite poorly in general. This analysis confirms the suitability of the RF as

a stable algorithm that performs well without the need of tuning. Automatic ML methods could be used to improve even further the configuration of the meta-model, although this is out of scope of the current research.

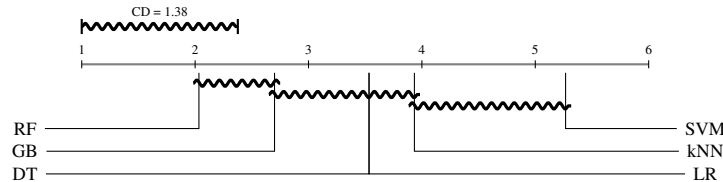


Fig. 7. Nemenyi post-hoc test (significance of $\alpha = 0.05$ and critical distance of 1.38) of several classifiers considering the F-score obtained for recommending the optimal encoding technique

Matching Process Behavior and Encoding Techniques With the goal of providing a more refined analysis, in the next sections we reduce the set of encoding techniques to the best representative of each family: alignment, doc2vec and node2vec. Moreover, considering the results from the previous test, we use the RF algorithm as the meta-model. The recommendations provided via MtL are based on data-driven assumptions constructed using meta-features from historical data. In this section, we present the meta-feature importance for predicting and explain the impact of the meta-features for recommending the encoding method (meta-target) in each anomaly context. The RF importance of the obtained model was used to discuss the relevance of features. To enlighten the interpretation of models, we exploit the Shapley Additive Explanations (SHAP), proposed in [29], explaining the obtained predictions in the different anomaly environments. We chose to apply two interpretation methods (RF importance and SHAP) because they are complementary and capture different aspects of feature relevance. By using RF importance, we can assess the most impactful meta-features for predicting any meta-target. As expected though, these importances are influenced by the bias of the ML model. Complimentarily, SHAP provides a more in-depth analysis, both at the class level (meta-targets) and for subsets of meta-instances (e.g., groups of specific anomalies). This way, we aim to provide a comprehensive study of the relationship between process behavior and optimal techniques.

Figure 8 shows all 80 features sorted by importance (i.e., ranked using RF importance) and colored using different colors for each feature family (*activity*, *trace*, and *log*). The top-ranked features were from trace and activity families with 65% and 35%, respectively, in the top 20 most important features. In general, trace-related features were the most successful in capturing the process behavior, which indicates that information sitting on the case-level may be more important than event-level information when describing normal and anomalous executions. A suitable explanation for such a pattern is that trace features have access to the complete case context. Thus, the difference between cases is more easily detected by this feature family. Particularly, *trace_len_entropy*, *trace_len_hist6* and *trace_len_skewness_hist* were the best from this group. *Trace_len_entropy* captures

the entropy of trace lengths of a process, meaning that anomalies affect the distribution of trace sizes, which turns to be an efficient indicator of process behavior. *Trace_len_hist6* and *trace_len_skewness_hist* refer to a particular histogram bin (the sixth) and the skewness of this histogram, respectively. Histograms are fair data descriptors and have already been used in the PM domain [7]. These results indicate that processes have different distributions, and their asymmetry is an important indicator of process behavior. Since anomalies directly affect the executed activities, activity-based features also produce meaningful content. The standard deviation of the number of unique ending activities (*end_activities_std*) was the most influential feature from this family. Indeed, anomalies affecting the trace tail may be more detectable since the number of possible end activities is more limited. Many of the most important activity-level descriptors are related to start and end activities variance, implying that anomalies substantially affect activity distribution. Log-based features were considered the less important by RF, and none was featured in the top-20. The best-ranked feature from this family is the number of events in the process. On the other hand, the number of traces was useless for this context. The number of variants and their ratio were also less relevant for choosing the appropriate encoding. Usually, trace variants are an indicative of log complexity, and for traditional PM tasks such as process discovery and conformance checking, the number of variants is very influential on the results. However, in this scenario where we aim at finding the best encoding method, variants are less influential because word embedding and graph embedding techniques are used to deal with more extensive corpora. This way, the application of these methods cannot be easily identifiable by the number of variants or their ratio. We also observe that features based on simpler statistical tools such as average and median were predominantly less important than features relying on more powerful statistical concepts.

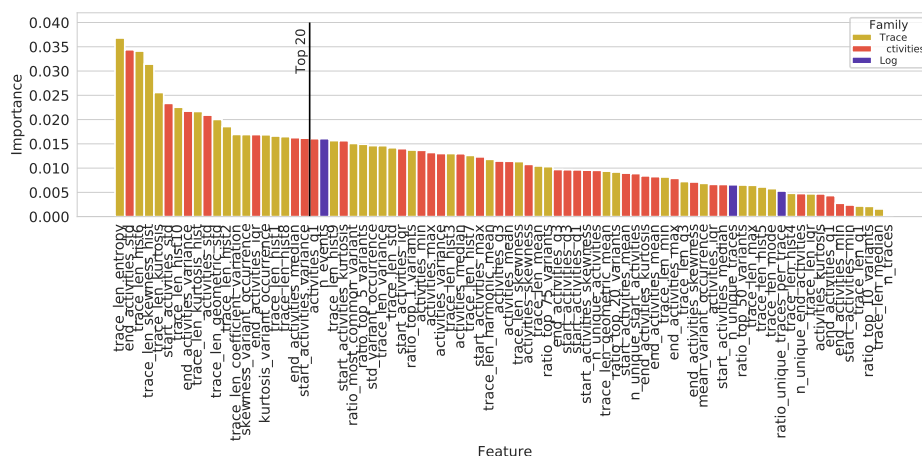


Fig. 8. RF feature importance sorted by their importance and colored by feature family

It is expected that different anomaly scenarios demonstrate particularities when having their pattern modeled, which demands information provided by specific groups of features. We used the SHAP method to comprehend these particularities and pave the way for an extended discussion on the features' importance and their contribution to the description of particular scenarios.

The top 20 most impacting features addressing the recommendation of encoding methods under all anomalies under study are shown in Figure 9. In a scenario comprised of several anomalies, the SHAP method indicated the impact of *trace_len_hist6* and *end_activities_std* as the most influential features. The former mostly supports the decision between *doc2vec* and *node2vec*, with a smaller impact when choosing alignment as the suitable encoding method. Particularly, *trace_len_hist6* was remarkably important for *doc2vec*, meaning that particular distributions are better encoded by this method. Furthermore, *end_activities_std* is more suitable to indicate a decision between alignment and *node2vec*, being the most important meta-feature in recognizing the suitability of *node2vec* for an event log. Moreover, *end_activities_std* and *start_activities_std* (both from the activity family) were the most influential features when associating a process with alignment. Indeed, the configuration of starting and ending activities highly affects the alignment's performance, hence, it was clear to the meta-model how these features affected the decision for alignment. On the other hand, descriptors such as the entropy of trace lengths, number of events, and most frequent activity had a minor impact.

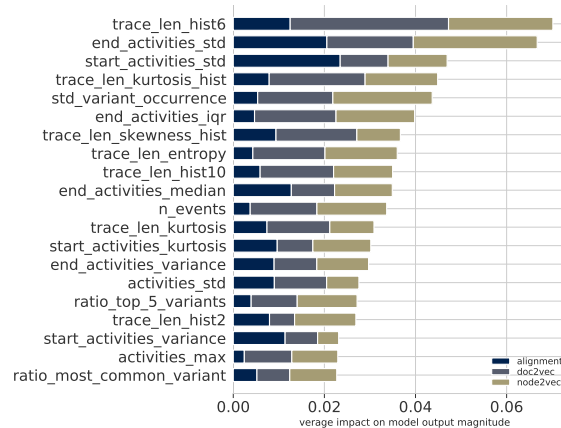


Fig. 9. Top 20 meta-features ranked using SHapley Additive exPlanations (SHAP) for all anomalies

Aiming at a better comprehension of the relationship between anomaly patterns and process behavior, Figure 10 depicts the most influential features for each anomaly type (*attribute*, *early*, *insert*, *late*, *rework*, and *skip*). As in the previous analysis, *end_activities_std* and *trace_len_entropy* had a high impact for most anomalies, confirming again their competence of correctly capturing the process behavior for the studied problem. The exception was the *late* anomaly, in which *trace_len_entropy* occupied only the 13th posi-

tion in the rank. In particular, in a scenario compromised by *late* anomaly (Figure 10d), the *trace.len.kurtosis* performed an important descriptive role, reaching a value that is double of the value of the 2nd most impacting feature. This scenario exposes the importance of observing when a sequence of events is executed too prematurely, creating a deviation in the normal distribution, which is then captured by the *trace.len.kurtosis* meta-feature.

When analyzing the relationship between features importance and specific anomalies, we observe that in *late* and *skip* anomalies, the meta-features had a lower impact in deciding for the alignment method. On the other hand, in *insert* and *rework* the features had a greater impact on the same technique. Indeed, the latter anomalies may be more easily detected by the alignments. The *late* anomaly is also challenging for the other encodings (i.e., the importance for *node2vec* and *doc2vec* was lower with this deviation). The highest average impact for *doc2vec* and *node2vec* happened for *insert* and *early* anomalies, respectively.

The single feature from log family present among the top 20 was *n_events*. This feature contributed to identifying *attribute* and *skip* anomalies, consequently also appearing when recommending an encoding method in a scenario with all anomalies (Figure 9). Again, the trace and activity families were predominant in the top 20 for all anomalies. Although different features are selected and sorted among the top 20, the contributions are similar to those presented using RF importance. In other words, all the presented meta-features contributed to provide the recommendation by the proposed MtL approach, except for *n_traces*. Prematurely ignoring some characteristics interferes in the recommendation of particular scenarios, indicating that the high descriptive power is reached by the combination of multiple descriptors. Such outcome indicates that including additional features might lead to better representations and, hence, an improved recommendation performance. Furthermore, by using the RF algorithm when modeling the recommendation system, we obtained a model composed only of features able to contribute to the final prediction since the RF properties guarantee a model that is built upon features capable of supporting the most accurate results. Overall, considering the results of both analysis, we can conclude that process behavior (captured by meta-features) does influence the quality of the encoded event log. Therefore, characteristics of the underlying business process should be considered when choosing an appropriate encoding technique, a problem that is tackled by our proposal.

Experimental validity Mendling et al. [31] raised a discussion about algorithm engineering and its impact on accuracy results, questioning if improved performances are a merit of algorithm design or a result of biases in both data and technique. We adopted their proposed framework to assess the internal validity of our research design and experiments. As stated by the authors, when the manipulation of research artifacts is causally responsible for an effect, its research design is internally valid. Therefore, experiments built on randomization constrain the effect of confounding factors. Since our dataset might contain biases due to an imbalance in the representational level of meta-features, we apply a resampling strategy based on randomization to assess our research design quality. For that, we compare the resampling results with the previously reported in Figure 4b. Essentially, we submitted the meta-database to a binning procedure using the *trace.len.entropy* meta-feature as it was the best ranked according to the RF importance (Figure 8). Fol-

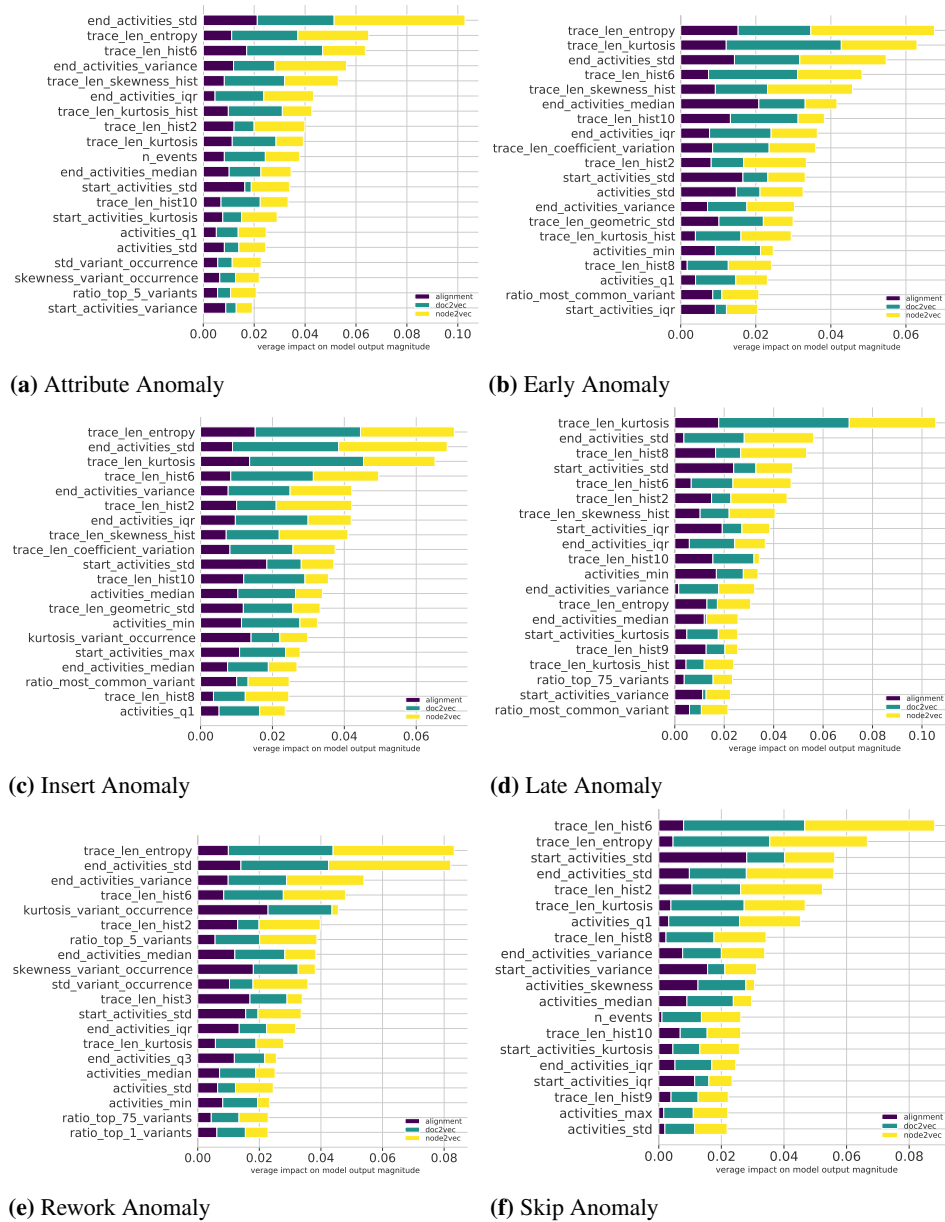


Fig. 10. Top 20 features ranked using SHapley Additive exPlanations (SHAP) for each anomaly type

lowing, we selected a percentage of samples per bin, creating a resampled meta-database, which was later fed to the same research pipeline as described in Section 4. Table 4 reports the results of the resampling experiment with a gradual increase of sampling size

percentages. As expected, when smaller percentages of meta-instances are selected for each bin, we observe a reasonable decrease in both performance metrics. Such behavior is expected simply because the learning model has fewer examples to generalize from. With the increasing percentage of meta-instances, both accuracy and F-score also rise in performance, indicating convergence to the result observed in Figure 4b. This behavior confirms the internal validity of our research design, which, given a meta-database of increasing representational quality, improves the recommendation performance.

Resampling size	10%	20%	30%	40%	50%
Accuracy	0.52	0.56	0.58	0.64	0.68
F-score	0.43	0.50	0.53	0.62	0.65

Table 4. Resampling experiment prediction performance

Complimentarily, we report the confusion matrices for each resampling percentage in Table 5. In general, the performance for each class increases along with the resampling size. However, we note that the alignment class is particularly challenging to identify. For instance, there is a high number of false positives associated with this class. Although they decrease as the meta-database grows, there is still a significant amount of misclassifications with this particular target. This pattern confirms the analysis of feature importance presented in Figures 9 and 10. Considering that it is the most difficult class to decide from (the meta-features have a lower average impact), the misclassifications are a consequence. Doc2vec and node2vec are more easily identified by the classifier, which corroborates to a better performance for these classes.

resampling encoding	10%			20%			30%			40%			50%		
	A	D	N	A	D	N	A	D	N	A	D	N	A	D	N
alignment (A)	0.28	0.46	0.26	0.42	0.33	0.26	0.38	0.41	0.21	0.47	0.34	0.19	0.49	0.33	0.18
doc2vec (D)	0.18	0.57	0.25	0.23	0.58	0.19	0.23	0.63	0.13	0.22	0.68	0.1	0.19	0.74	0.07
node2vec (N)	0.18	0.21	0.61	0.22	0.19	0.59	0.18	0.14	0.68	0.19	0.09	0.73	0.17	0.08	0.75

Table 5. Resampling experiment confusion matrix. This analysis extends Table 4 by providing a more in-depth notion of which encoding techniques are more difficult to be identified and how the performances change for an increasing size of the meta-database

6. Conclusion

Organizations are interested in detecting anomalous instances in their business processes as a method to leverage process quality, avoid resource waste, and mitigate security issues. In this work, we proposed to combine encoding techniques with MtL to enhance the detection of anomalous traces in event logs. Our strategy relies on a powerful set of meta-features extracted from the event logs. We showed its viability by recommending the best

encoding technique with an F-score of 0.43, statistically overcoming the baselines. MtL boosted the anomaly detection by fitting the optimal encoding technique for each event log, statistically outperforming the usage of a single encoding technique.

Furthermore, in this extension of the original research [41], we dove into the influence of meta-features on the recommended encoding technique. For that, we extracted descriptors' importance from the ML model and also exploited the SHAP method to explain the predictions. This analysis leveraged the understanding of which features (and their families) better capture process behavior in the context of anomaly detection. Considering concerns about algorithm engineering, we assessed the internal validity of the research design by applying a resampling strategy. The results indicated the validity of the experiments in this environment. For future works, we plan to include more encoding techniques and propose additional features for the meta-feature extraction step.

References

1. van der Aalst, W.: *Process Mining: Data Science in Action*. Springer Berlin Heidelberg (2016), <https://doi.org/10.1007/978-3-662-49851-4>
2. Adam, S.P., Alexandropoulos, S.A.N., Pardalos, P.M., Vrahatis, M.N.: No Free Lunch Theorem: A Review, pp. 57–82. Springer International Publishing, Cham (2019), https://doi.org/10.1007/978-3-030-12767-1_5
3. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B.F., van der Aalst, W.M.P.: Alignment based precision checking. In: La Rosa, M., Soffer, P. (eds.) *Business Process Management Workshops*. pp. 137–149. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
4. Augusto, A., Mendling, J., Vidgof, M., Wurm, B.: The connection between process complexity of event sequences and models discovered by process mining. *Information Sciences* 598, 196–215 (2022), <https://www.sciencedirect.com/science/article/pii/S0020025522002997>
5. Barbon Jr., S., Ceravolo, P., Damiani, E., Omori, N.J., Tavares, G.M.: Anomaly detection on event logs with a scarcity of labels. In: *2020 2nd International Conference on Process Mining (ICPM)*. pp. 161–168 (2020)
6. Barbon Jr., S., Ceravolo, P., Damiani, E., Tavares, G.M.: Evaluating trace encoding methods in process mining. In: Bowles, J., Broccia, G., Nanni, M. (eds.) *From Data to Models and Back*. pp. 174–189. Springer International Publishing, Cham (2021)
7. Barbon Jr., S., Tavares, G.M., da Costa, V.G.T., Ceravolo, P., Damiani, E.: A framework for human-in-the-loop monitoring of concept-drift detection in event log stream. In: *Companion Proceedings of the The Web Conference 2018*. p. 319–326. WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018), <https://doi.org/10.1145/3184558.3186343>
8. Bezerra, F., Wainer, J., van der Aalst, W.M.P.: Anomaly detection using process mining. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) *Enterprise, Business-Process and Information Systems Modeling*. pp. 149–161. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
9. Bezerra, F., Wainer, J.: Algorithms for anomaly detection of traces in logs of process aware information systems. *Information Systems* 38(1), 33–44 (2013), <https://www.sciencedirect.com/science/article/pii/S0306437912000567>
10. Böhmer, K., Rinderle-Ma, S.: Multi-perspective anomaly detection in business process execution events. In: *On the Move to Meaningful Internet Systems: OTM 2016 Conferences*. pp. 80–98. Springer International Publishing, Cham (2016)

11. Bose, R.P.J.C., van der Aalst, W.M.: Context Aware Trace Clustering: Towards Improving Process Mining Results, pp. 401–412 (2019), <https://epubs.siam.org/doi/abs/10.1137/1.9781611972795.35>
12. Breiman, L.: Random forests. *Machine learning* 45(1), 5–32 (2001)
13. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and regression trees*. Routledge (1984)
14. Burattin, A.: *Plg2: Multiperspective processes randomization and simulation for online and offline settings* (2015)
15. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: *Conformance Checking - Relating Processes and Models*. Springer (2018)
16. Ceravolo, P., Tavares, G.M., Barbon Jr., S., Damiani, E.: Evaluation goals for online process mining: a concept drift perspective. *IEEE Transactions on Services Computing* pp. 1–1 (2020)
17. De Koninck, P., vanden Broucke, S., De Weerd, J.: act2vec, trace2vec, log2vec, and model2vec: Representation learning for business processes. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *Business Process Management*. pp. 305–321. Springer International Publishing, Cham (2018)
18. Delias, P., Doumpos, M., Grigoroudis, E., Matsatsinis, N.: A non-compensatory approach for trace clustering. *International Transactions in Operational Research* 26(5), 1828–1846 (2019), <https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12395>
19. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30 (Dec 2006), <http://dl.acm.org/citation.cfm?id=1248547.1248548>
20. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5), 1189–1232 (2001)
21. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. p. 855–864. KDD '16, ACM, New York, USA (2016)
22. Hake, P., Zapp, M., Fettke, P., Loos, P.: Supporting business process modeling using rnns for label classification. In: Frasincar, F., Ittoo, A., Nguyen, L.M., Métails, E. (eds.) *Natural Language Processing and Information Systems*. pp. 283–286. Springer International Publishing, Cham (2017)
23. He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. *Knowledge-Based Systems* 212, 106622 (2021), <https://www.sciencedirect.com/science/article/pii/S0950705120307516>
24. Kotthoff, L.: Algorithm selection for combinatorial search problems: A survey. In: Bessiere, C., De Raedt, L., Kotthoff, L., Nijssen, S., O'Sullivan, B., Pedreschi, D. (eds.) *Data Mining and Constraint Programming: Foundations of a Cross-Disciplinary Approach*. pp. 149–190. Springer International Publishing, Cham (2016)
25. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Xing, E.P., Jebara, T. (eds.) *Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 32, pp. 1188–1196. PMLR, Beijing, China (22–24 Jun 2014)
26. Lee, W.L.J., Verbeek, H., Munoz-Gama, J., van der Aalst, W.M., Sepúlveda, M.: Re-composing conformance: Closing the circle on decomposed alignment-based conformance checking in process mining. *Information Sciences* 466, 55–91 (2018), <https://www.sciencedirect.com/science/article/pii/S0020025518305413>
27. Leontjeva, A., Conforti, R., Di Francescomarino, C., Dumas, M., Maggi, F.M.: Complex symbolic sequence encodings for predictive monitoring of business processes. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) *Business Process Management*. pp. 297–313. Springer International Publishing, Cham (2015)
28. Luetgen, S., Seeliger, A., Nolle, T., Mühlhäuser, M.: Case2vec: Advances in representation learning for business processes. In: Leemans, S., Leopold, H. (eds.) *Process Mining Workshops*. pp. 162–174. Springer International Publishing, Cham (2021)

29. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 30. Curran Associates, Inc. (2017), <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>
30. Märušter, L., Weijters, A.T., Van Der Aalst, W.M., Van Den Bosch, A.: A rule-based approach for process discovery: Dealing with noise and imbalance in process logs. *Data mining and knowledge discovery* 13(1), 67–87 (2006), <https://doi.org/10.1007/s10618-005-0029-z>
31. Mendling, J., Depaire, B., Leopold, H.: *Theory and practice of algorithm engineering* (2021), <https://arxiv.org/abs/2107.10675>
32. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013), <https://arxiv.org/abs/1301.3781>
33. Nolle, T., Luetzgen, S., Seeliger, A., Mühlhäuser, M.: Analyzing business process anomalies using autoencoders. *Machine Learning* 107(11), 1875–1893 (2018), <https://doi.org/10.1007/s10994-018-5702-8>
34. Nolle, T., Luetzgen, S., Seeliger, A., Mühlhäuser, M.: Binet: Multi-perspective business process anomaly classification. *Information Systems* 103, 101458 (2022), <https://www.sciencedirect.com/science/article/pii/S0306437919305101>
35. Nolle, T., Seeliger, A., Mühlhäuser, M.: Binet: Multivariate business process anomaly detection using deep learning. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *Business Process Management*. pp. 271–287. Springer International Publishing, Cham (2018)
36. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *ADVANCES IN LARGE MARGIN CLASSIFIERS*. pp. 61–74. MIT Press (1999)
37. Polato, M., Sperduti, A., Burattin, A., Leoni, M.d.: Time and activity sequence prediction of business process instances. *Computing* 100(9), 1005–1031 (Sep 2018), <https://doi.org/10.1007/s00607-018-0593-x>
38. Rice, J.R.: The algorithm selection problem. *Advances in Computers*, vol. 15, pp. 65–118. Elsevier (1976), <https://www.sciencedirect.com/science/article/pii/S0065245808605203>
39. Rozinat, A., van der Aalst, W.: Conformance checking of processes based on monitoring real behavior. *Information Systems* 33(1), 64–95 (2008), <https://www.sciencedirect.com/science/article/pii/S030643790700049X>
40. Tavares, G.M., Barbon, S.: Analysis of language inspired trace representation for anomaly detection. In: Bellatreche, L., Bieliková, M., Boussaïd, O., Catania, B., Darmont, J., Demidova, E., Duchateau, F., Hall, M., Merčun, T., Novikov, B., Papatheodorou, C., Risse, T., Romero, O., Sautot, L., Talens, G., Wrembel, R., Žumer, M. (eds.) *ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium*. pp. 296–308. Springer International Publishing, Cham (2020)
41. Tavares, G.M., Barbon Jr., S.: Process mining encoding via meta-learning for an enhanced anomaly detection. In: Bellatreche, L., Dumas, M., Karras, P., Matulevičius, R., Awad, A., Weidlich, M., Ivanović, M., Hartig, O. (eds.) *New Trends in Database and Information Systems*. pp. 157–168. Springer International Publishing, Cham (2021)
42. Tavares, G.M., Barbon Junior, S., Damiani, E., Ceravolo, P.: Selecting optimal trace clustering pipelines with meta-learning. In: Xavier-Junior, J.C., Rios, R.A. (eds.) *Intelligent Systems*. pp. 150–164. Springer International Publishing, Cham (2022)
43. Tavares, G.M., Junior, S.B., Damiani, E.: Automating process discovery through meta-learning. In: Sellami, M., Ceravolo, P., Reijers, H.A., Gaaloul, W., Panetto, H. (eds.) *Cooperative Information Systems*. pp. 205–222. Springer International Publishing, Cham (2022)

44. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Trans. Knowl. Discov. Data* 13(2) (mar 2019), <https://doi.org/10.1145/3301300>
45. Tipping, M.E., Bishop, C.M.: Mixtures of Probabilistic Principal Component Analyzers. *Neural Computation* 11(2), 443–482 (02 1999)
46. van der Aalst, W., de Medeiros, A.: Process mining and security: Detecting anomalous process executions and checking process conformance. *Electronic Notes in Theoretical Computer Science* 121, 3–21 (2005), proceedings of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004)
47. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. *Artificial Intelligence Review* 18(2), 77–95 (Jun 2002), <https://doi.org/10.1023/A:1019956318069>

Gabriel Marques Tavares is a Ph.D. candidate in Computer Science at the University of Milan (UNIMI), Italy. He received his BSc and MSc at the Londrina State University (UEL) in 2019, while in 2014 he participated in an exchange program at the University of Michigan. Currently, his research activities focus on leveraging Process Mining analysis using Machine Learning methods. More specifically applying Meta-learning and Automated Machine Learning to unveil relationships between data characteristics and optimal pipelines. He has investigated themes such as automated Process Discovery, Trace Clustering, Event log encoding, Data Representation, Process Explainability, and Online Process Mining with particular attention to Concept Drift Detection.

Sylvio Barbon Junior is an Associate Professor at the Department of Engineering and Architecture at the University of Trieste (UNITS), Italy. Currently, he is part of the Machine Learning Lab. Previously, he was the leader of the research group that studies machine learning in the Computer Science Department at the State University of Londrina (UEL), Brazil. He received his BSc degree in Computer Science in 2005, MSc degree in Computational Physics from the University of São Paulo (2007), a degree in Computational Engineering in 2008, and a Ph.D. degree (2011) from IFSC/USP such as the MSc degree. In 2017, he was visiting researcher at the University of Modena and Reggio Emilia (Italy) working on multispectral analysis and machine learning. Still, in 2017, he was visiting researcher at Università Degli Studi Di Milano (Italy) focused on Data Stream and Process Mining. He is currently a professor in postgraduate and graduate programs. His research interests include Computer Vision, Pattern Recognition, and Machine Learning. Currently, focusing on Meta-Learning, Stream Mining, and Process Mining.

Received: January 10, 2022; Accepted: November 10, 2022.

A Framework for Privacy-aware and Secure Decentralized Data Storage*

Sidra Aslam^{1,2} and Michael Mrissa^{1,2}

¹ InnoRenew CoE, Livade 6, 6310 Izola, Slovenia
sidra.aslam@innorenew.eu

² University of Primorska, Faculty of Mathematics, Natural Sciences
and Information Technology, Glagoljaška ulica 8, 6000 Koper, Slovenia
michael.mrissa@innorenew.eu

Abstract. Blockchain technology gained popularity thanks to its decentralized and transparent features. However, it suffers from a lack of privacy as it stores data publicly and has difficulty to handle data updates due to its main feature known as immutability. In this paper, we propose a decentralized data storage and access framework that combines blockchain technology with Distributed Hash Table (DHT), a role-based access control model, and multiple encryption mechanisms. Our framework stores metadata and DHT keys on the blockchain, while encrypted data is managed on the DHT, which enables data owners to control their data. It allows authorized actors to store and read their data in a decentralized storage system. We design REST APIs to ensure interoperability over the Web. Concerning data updates, we propose a pointer system that allows data owners to access their update history, which solves the issue of data updates while preserving the benefits of using the blockchain. We illustrate our solution with a wood supply chain use case and propose a traceability algorithm that allows the actors of the wood supply chain to trace the data and verify product origin. Our framework design allows authorized users to access the data and protects data against linking, eavesdropping, spoofing, and modification attacks. Moreover, we provide a proof-of-concept implementation, security and privacy analysis, and evaluation for time consumption and scalability. The experimental results demonstrate the feasibility, security, privacy, and scalability of the proposed solution.

Keywords: Blockchain, Distributed Hash Table, Security, Privacy, Decentralized framework.

1. Introduction

With increasing the number of internet users, large amounts of data are being generated each day [18]. Cloud computing provides the facility to store, access, and share data with other users anytime. The main limitation of the cloud paradigm is its centralized storage design, which leads to a single point of failure issue. Cloud storage systems rely on Trusted Third Party (TTP) to collect and store users' privacy-sensitive data, which is more vulnerable to security and attacks. To address these challenges, blockchain has become popular as a decentralized and transparent data management facility [23], [42]

* This is an extended version of our previous paper [2].

that enables users to share and store information without any TTP. A blockchain is a peer-to-peer distributed ledger in which a list of records called blocks are linked with each other and secured using a cryptographic hash function [35]. It stores data on distributed nodes through a consensus mechanism that guarantees participant's trust by having the same copy of the data [34], [37].

However, blockchain allows anyone to read and write contents, which may raise data security issues [40], and does not handle privacy-sensitive data [21] by default. This is a limitation since data owners may not want to disclose their sensitive information (e.g. statistics about their business activities) on the blockchain. Scalability is also an issue, as the data is replicated on every peer, storing large quantities has a prohibitive cost. Besides this, immutability of blockchain, while an important feature, prevents data modifications.

In this paper, we propose a privacy-aware decentralized data storage and management framework that enables actors to write, read, delete, update, and access their transactions history. Our solution allows data owners to control and secure their data in a decentralized ledger. Building on previous work [2], our proposed framework is scalable enough to handle an increasing number of actors while performing data write, read, update, and delete operations. The main contributions of this paper are as follows:

- We propose a metadata extension based on existing research [1]. Our extension ensures privacy-aware data access and enables trust between actors by recording each actor's actions on data.
- We propose a pointer system to manage the history of values that are stored in the DHT for a single piece of data. It allows the data owner to maintain and access their transactions history in case of any updates in the pre-stored data.
- We propose a traceability algorithm that enables actors to trace their data and verify the product's origin in a decentralized platform.
- We design and evaluate our decentralized framework against linking, eavesdropping, spoofing, and modification attacks.
- We provide a critical comparison of the proposed solution with state-of-the-art decentralized solutions to show the research gap.
- We also provide implementation details with security and privacy analysis and performance evaluation of our framework over a wood supply chain scenario to demonstrate its feasibility.

This paper is structured as follows. Section 2 discusses the motivating scenario that highlights the research challenges. Section 3 provides some background knowledge together with an overview of existing decentralized solutions for data storage and their shortcomings. Section 4 provides the detailed discussion of our contribution with proposed algorithms. Section 5 shows the experimental results, analysis, and performance evaluation of our proposed framework. Finally, section 6 summarizes our results and gives guidelines for the future work.

2. Motivating Scenario and Research Problem

In this section, we first explain the wood supply chain scenario that motivates our work. We then describe the research problems that we address in this paper.

2.1. Motivating Scenario

Our scenario takes place in the context of the wood supply chain that motivates the need for decentralized solution and highlights our research problems. The wood supply chain includes the whole process from wood logs, production, transportation, and sell to the end customers. It enables the actors of the wood supply chain to verify the wood origin, transport, processing, and manufacturing. As depicted in Figure 1, we identified six actors that participate in the wood supply chain.

- **Forest manager**
The forest manager identifies the trees that are good to make furniture (e.g oak) and cuts them into logs.
- **Transporter**
The transporter loads wood logs from the forest and transports them to the sawmill.
- **Sawmill manager**
It processes the logs and stores them for a specific time duration.
- **Product assembler**
It divides logs into pieces for further processing.
- **Product seller owner**
The product seller owner sells the furniture to the end customer.
- **Customer**
The customer takes the wooden furniture and confirms the origin of the wood using the proposed traceability algorithm (see Section 4.4).

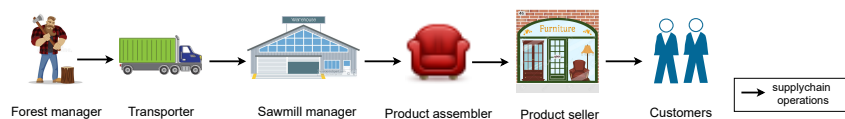


Fig. 1. Wood supply chain and its actors

This scenario highlights the need for decentralized data management, security, privacy, traceability, and data updates [36]. Frauds are common in the wood supply chain, for example, during transportation actors can replace high quality wood with low quality wood [30]. Therefore, all actors participating in the supply chain want to trace products to prevent frauds. To overcome this problem, Radio Frequency Identification (RFID) chips are used with the wood to manage wood traceability [31]. However, existing solutions involve centralized storage to maintain the record of RFID data, thus making single point of failure a major concern [26].

Therefore, blockchain, as a decentralized ledger technology that stores transactions in such a way that all participants can easily access them without requiring any TTP, comes as an interesting technology for solving the single point of failure issue. Each block of the blockchain keeps the hash of its previous block to make it impossible to modify the stored transactions thus ensuring immutability [25,13]. We can say that data cannot be modified once it has been recorded on the blockchain. However, our scenario highlights that actors

of the wood supply chain need to write, read, and update data about their product. As well, they do not want to have their business information publicly available due to security and privacy concerns. There is a need for a solution that overcomes the immutability feature of blockchain, to enable actors to perform update operation on recorded data. At the same time, the designed solution must protect data from unauthorized access and guarantee data access depending on the actor's permission. The identified requirements highlight our motivation to design decentralized data storage and management solution to ensure data access and updates, manage transactions history, security, privacy, data owner's control on their data, and product traceability in a single framework.

2.2. Research Problems

According to the wood supply chain scenario discussed above, using blockchain technology in supply chains requires taking into account the following research problems:

– Data modification

Our scenario highlights that actors want to update data at each point of the supply chain (e.g wood location changes). However, it is not possible to update data once it has been recorded on the blockchain, due to its immutability feature. The challenge is to work around the original blockchain design to enable data updates.

– Data security and privacy

Data stored on a blockchain is publicly accessible, highlighting the need for protection from unauthorized access. In other words, different actors shall be granted different access to specific data pieces according to their permissions. The challenge is to provide a decentralized solution that preserves privacy-sensitive data from unauthorized access to ensure data security and privacy.

To address those challenges, we rely on joint usage of blockchain and Distributed Hash Table (DHT), presented in the following to facilitate further understanding of the paper, together with an overview of existing work and its limitations.

3. Background Knowledge and Related Work

In this section, we introduce the basic concepts underpinning blockchain technology and Distributed Hash Table (DHT), we then explain their use in the context of decentralized data storage. We follow with a survey and analysis of existing decentralized data storage solutions. We compare our proposed solution with existing work and summarize the results in Table 1.

3.1. Basic Concepts: Blockchain and DHT

In 2008, blockchain technology [22] was introduced to the world and became popular due to its decentralized feature. The blockchain is a decentralized database that stores all the transactions that take place on the network. All participants on the network have the same copy of the transactions. Before adding each block to the blockchain, miners

accept and verify the transactions using a consensus algorithm such as proof of work. By using proof of work or similar mechanism [15], miners solve very difficult mathematical calculations that should be accepted by other miners on the network [3]. After verifying the correctness of transactions by other miners, a block is appended to the end of the chain [24]. Each block is comprised of a block version, timestamps, consensus signature, parent block hash, and many transactions. The parent block stores the hash of its previous block to form a blockchain that ensures the immutability of the stored data [32]. The hash is a unique value that ensures integrity of the entire blockchain from the initial block (known as genesis block) to the last.

A distributed hash table (DHT) is a decentralized data storage system that stores data as (key, value) pairs over a set of nodes that distribute the storage, possibly with some level of replication. As an example, a well-known DHT implementation is Kademlia [19]. Each node in the DHT maintains the keys it is responsible for and their corresponding values. A key is a unique identifier to its corresponding data value. Each key is generated by applying a hash function to the value. A DHT is based on two main tasks: PUT(key, value) is used to add new data, while GET(key) is used to retrieve the data, that is associated with the given key. A DHT node contains a routing table that maintains the identifier of its neighbor nodes. To find a (key, value) pair, a requesting node contacts the multiple nodes in the network until it reaches the destination node and finds the (key, value) pair. DHT has an advantage in terms of fault-tolerance because (key, value) pairs are replicated on multiple nodes in the network, that ensures data availability [43]. In addition, and as opposed to blockchain, it is scalable enough to manage large data volumes.

3.2. Blockchain and DHT-based data storage

There is a large amount of literature that combines DHT with blockchain to provide decentralized data storage. A framework to manage personal data is proposed in [43]. The solution stores encrypted data (with shared key) on DHT and its pointer on the blockchain. Both service and user can query the data. However, existing work supports one type of encryption. Most work use a shared symmetric key for data encryption/decryption, as in [43], to query the data. In contrast, our framework provides run-time flexibility, which provides various types of data encryption and decryption during execution depending on users' needs and application requirements. In [43], it is not clearly explained how symmetric keys are protected from unauthorized access and where they are stored. In our work, we encrypt symmetric key with the public key of the data owner, and store it on the DHT together with the data, so that later the data owner can access the data.

In [28], a distributed access control and data management framework is presented. The framework enables secure IoT (Internet of Things) data sharing by combining blockchain with off-chain storage (i.e DHT). Fine-grained access control permissions are stored on the blockchain and are publicly visible, which raises privacy issues. Also, it is not possible to update access control permissions due to public blockchain immutability nature. On the other hand, our proposed framework is flexible to update access control permissions. We also maintain data owner anonymity for sharing data.

In [1], the authors propose a decentralized data storage for PingER (Ping End-to-End Reporting) framework. The proposed framework stores metadata of the daily PingER files on a permissioned blockchain, while the original data is stored off-chain. However, their solution writes monitoring agent name and file locations on the permissioned blockchain,

which is immutable and shared with other participants on the network. In addition, this solution does not record the data modification history in case of any modification in the data. Our framework design relies on the PingER proposal for the metadata structure, however, we integrate privacy and security management to enable role-based access control and privacy protection. Our solution enables data owners to control and access their private data. We also provide a solution to manage the previous versions of data using pointers that enable authorized users to access their transaction history. In addition, our work includes proof of concept prototype as well as empirical performance evaluation, which is not the case in PingER.

Table 1. Our proposed framework comparison with existing work

Solutions	Decentralization	Data Privacy	Data Updates	Transaction History Support	Attacks Prevention
[43]	Yes	Yes	No	No	No
[28]	Yes	No	No	No	No
[1]	Yes	No	No	No	No
[8]	Yes	No	No	No	No
[16]	Partial	No	No	No	No
[5]	Yes	Yes	No	No	No
[38]	Yes	No	No	No	No
[11]	Partial	Yes	No	No	No
[41]	Partial	Yes	No	No	No
[7]	Partial	Yes	No	No	Yes
[27]	Yes	Yes	No	No	No
Our solution	Yes	Yes	Yes	Yes	Yes

The authors in [8], propose the LightChain framework, which is a permissionless blockchain that operates over participating peers of a skip graph DHT. The proposed framework enables all participating peers to access blocks and transactions by using a skip graph overlay. LightChain allows every peer to join the blockchain without any restrictions. However, blocks and transactions are addressable and accessible to everyone on the network. In contrast to the existing framework, our solution uses Role-based Access Control (RBAC) model that allows only authorized users to access blocks and transactions. We store metadata with a pointer on the blockchain, which enables other actors to keep track of data changes with the help of this metadata.

Table 1 presents a global overview of existing work with respect to the following features: decentralization, data privacy, data updates, transaction history support, and attacks prevention. The table shows that some existing solutions ensure decentralization, data privacy, and attacks prevention [43,7]. However, some solutions did not address data updates and transaction history support [28,1,8,27].

3.3. Other Decentralized Data Storage Solutions

An Ethereum-based blockchain platform is presented in [16]. The proposed solution allows companies partners to share data with each other. Original data are stored on off-

chain storage such as MySQL, while a hash sum of corresponding data is sent to the blockchain. However, MySQL database is not scalable as DHT to manage a large amount of data [12]. In addition, MySQL database becomes a single point of failure. In our solution, we use a DHT to store data as (key, value) pair, which can handle a large amount of data easily. In our framework, any authenticated user can efficiently retrieve the value with the help of a corresponding key. As well, our solution is fully decentralized and eliminates the risk of single point of failure.

In [5], the authors propose a framework called u-share. It is a blockchain-based framework to maintain the owner's data traceability while sharing data with their friends and family. The proposed framework is based on a software client to share the private keys with corresponding circle members, keeps a record of shared keys, and encrypt the data using the circle's public key before to share it. However, sharing private key raises security issues. Additionally, the existing framework relies on one type of encryption method. Compared to the existing u-share framework, our proposed solution allows actors to directly generate their public and private keys at run time and control of their private keys. Our solution allows data owners to directly encrypt, decrypt, and share their data with other actors by using different types of encryption methods.

The authors in [29] present a blockchain-based framework that enables users to share their data with other users. A smart contract is used to store data sharing policies that control users' access to the data, while users' private data is stored on the off-chain storage called multi-chain. However, policies stored on the smart contract are immutable. In contrast, our solution enables data owners to update access control permissions. In addition, we ensure data owner anonymity to share data.

In [38], a decentralized supply chain system to keep track of goods and recipe ingredients is presented. The proposed framework uses a smart contract to handle the exchange of goods on a distributed ledger. The main limitation of this solution is the immutability and availability of data to everyone, which could lead to privacy and data modification concerns. On the other hand, our solution stores encrypted data on DHT to ensures data privacy. In addition, our framework allows actors to update data at each point of the chain.

In [11], a blockchain-based food supply chain traceability through smart contract is presented. The proposed framework uses blockchain to store data hash while corresponding data are stored on IPFS (InterPlanetary File System) off-chain storage. IPFS is a peer-to-peer storage network where data stores on the peers of the network [17]. However, a manufacturer node server is used to handle all modules of the framework, which subjects to a single point of failure. On the other hand, our framework modules are fully decentralized and independent of any central orchestrator. For the sake of simplicity, we use a registry server to connect nodes to each other, however, decentralized discovery protocol can easily be used instead of registry server [9].

In [41], the authors propose a decentralized IoT data sharing solution using IOTA Tangle and IPFS technology. The proposed solution uses centralized data handling unit (such as a local server) to collect and encrypt the data using asymmetric encryption, which becomes a single point of failure. In contrast, our proposed solution manage and store data without any central party. The IPFS is used to upload the encrypted data, while the corresponding hash and metadata are managed on the IOTA Tangle. However, the IPFS network is immutable and stores files and its content permanently [14]. On the other hand, we use DHT to store the data and we extend it to allow data modification at any time.

In contrast, our solution allows going through the history of data values and supports querying it.

The authors in [33] propose a blockchain-based framework that maintains the traceability of the food supply chain. RFID technology is used to automatically identify objects through radio frequency signals. However, blockchain technology is not scalable to store a large amount of data. In contrast to this solution, we propose to only store metadata and pointer on the blockchain, while original data is stored on a DHT, which better supports storing large amounts of data. In addition, our framework supports data mutability, thanks to the DHT, whereas blockchain is immutable and shows more difficulty to handle large amounts of data.

In [4] the authors discuss the distributed cloud storage system called Storj. It is a trust-based storage system between host and customer. In this system, people sell their free storage hardware space and earn money. Customers encrypt (using AES256-CTR) their data before storing it on the network. Storj allows the data owner to control and access their data on the network. However, Storj is very costly and depends on a centralized architecture to conclude storage data and payments [7,10]. In contrast, our solution is fully decentralized architecture and avoids a single point of failure. In addition, Storj uses one type of encryption method to establish trust between customer and host [39]. As compared to this, our solution offers different types of encryption methods and enables trust in the decentralized system instead of participants on the network.

In [27] the authors discuss a decentralized data storage framework that combines Solid Pods with blockchain technology. Solid (Social Linked Data) relies on RDF (Resource Description Framework) and semantic web to manage data. Solid enables people to store their personal data in Pods (Personal online data stores) hosted at the location according to the people's wish. The proposed framework discusses the following two cases to ensure data confidentiality. The first is to store file hash on the blockchain while Solid Pods is used to store the data. Second, they use smart contract to store the data on a Blockchain whereas solid pods are used to store the software wallet (public and private key pair). User can access their data using the software wallet. However, Solid Pods itself does not ensure data verification and trust [6]. In addition, it does not support storing large amounts of data as DHT does [20]. In contrast, our framework allows to manage large amounts of data in a decentralized way due to the use of a DHT. Therefore, our approach to data storage is quite different as we do not adopt a user-based isolated storage but rather a globally decentralized storage that relies on the network peers to ensure security and privacy.

In a summary, most existing data storage solutions are subject to the single point of failure issue, data mutability or adopt different designs. In the following, we detail our framework and proposed algorithms in detail.

4. Contribution

In this paper, we propose a secure and privacy-aware decentralized framework to support data storage, authorized data access, data mutability, management of their update history, and traceability. This section starts with the metadata structure that is immutable record of data operations. Then, it describes the overview of our proposed framework and follows with the detail of its execution or sequence. After that, it details the proposed algorithms.

Each actor of the framework runs the same code that is structured into a set of components as depicted in Figure 3.

4.1. Metadata Structure

In [1], authors write metadata such as names and locations only once a day on the permissioned blockchain, which is immutable and they shared this information with all users on the network. In contrast, we store metadata of each actor's action (such as data write date and time) to maintain the actor's trust. This allows actors to keep track of the data.

We propose a privacy-aware metadata extension discussed in the paper [1], to handle privacy restrictions on the data. Therefore, our framework encrypts the actor's private information (e.g name and location) with encryption mechanisms (illustrated in Algorithm 2), and store this encrypted data on the DHT. Our solution also allows only authorize actors to update the product location in case if wood drives from one place to another place. We use a blockchain to store the metadata and DHT key of this encrypted data. Our proposed metadata structure contains the DHT key, previous pointer, data owner's id, date, time, and RFID_number as shown in Figure 2. The DHT key is a hash pointer that points to the data in the DHT. Previous pointer is a hash key of the previous version of the data, which enables data owners to access their transaction history. In our framework, each actor has unique data owner id which is used to make a data request and identify who is the owner of the corresponding data. Our solution records data and time of each operation (such as data write, read, update, and delete) that is performed on the data. RFID_number is a unique data id of the log, lumber and product which is used to trace the items in the chain.

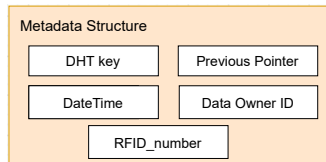


Fig. 2. Metadata structure on the blockchain

4.2. Architecture Overview

Our framework uses RESTful APIs to enable actors to communicate with other actors and support the framework functionalities.

Figure 3 depicts the execution workflow of the proposed framework and its components. In our framework, all actors are running the same `main` program and they call to `registry_server (/peers resource, method 'GET')` to retrieve the list of available actors (e.g peers) and connect with each other through APIs.

Let us illustrate the operation of our framework with the wood supply chain scenario developed earlier: an actor, for example a forest manager actor, starts the `main` program to store the number of logs and type of wood that he cuts. Then, he will call the `/peers`

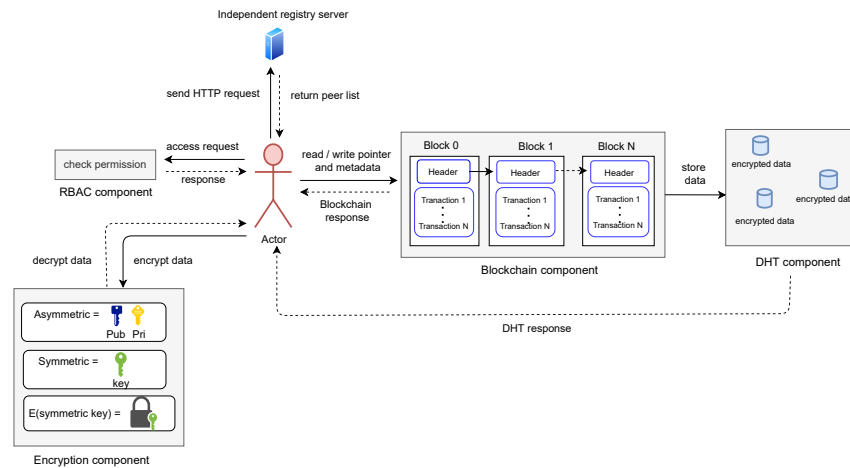


Fig. 3. Overview of the decentralized framework

resource of the `registry_server` with the 'POST' method to add its public key and Uniform Resource Locator (URL) to the list of connected peers or actors. After that, he will send a 'GET' request to the `/peers` resource to receive the information of available peers. Then, he will take a copy of the recent 40 transactions of the blockchain using `/chain` resource with a 'GET' method³.

In the proposed framework, the RBAC component called by the `main` component is responsible for checking the permission of the actor. It allows the only authorized actor to perform operations such as data write, read, delete, and update.

An authorized actor has a choice between multiple types of encryption techniques to secure their data in a decentralized ledger. Our `encryption_component` called by the `main` component generates keys (a public/private key pair, or a symmetric key) based on the encryption method chosen by the authorized actor and encrypts the data accordingly.

We store the encrypted data on the DHT component, while DHT key (a hash pointer of the data) and metadata are stored on the `blockchain` component. Later, an authorized actor can access their data using the DHT key stored on the `blockchain` component.

Accordingly, an authorized actor can create a new block using `/chain` resource with the method 'POST'. To read the data, an actor will call the resource `/chain/<id>` with 'GET' method. If an actor wants to update some part of the data, then it will call the `/chain/<id>` resource using 'PUT' method. Similarly, to delete the data, an authorized actor will make a 'DELETE' request to the `/chain/<id>` resource. An actor can access their public key using the resource `/public_key` with method 'GET'.

Figure 4 shows the swagger user interface that enables authorized actors to use the proposed APIs discussed above.

³ Please note that here we avoid downloading the whole blockchain due to performance issues, but only the most recent part, the rest being on-demand. This particular aspect of the work is out of the scope of this paper.

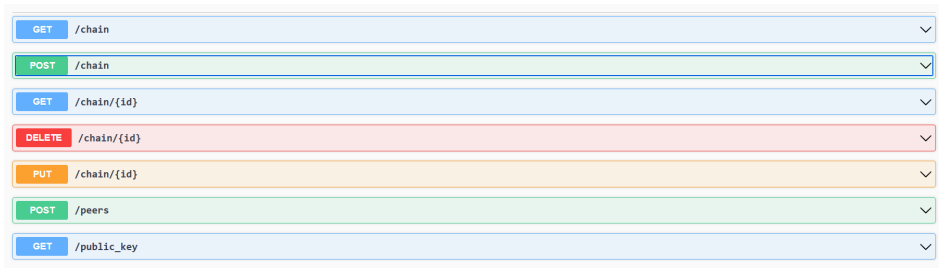


Fig. 4. Overview of the proposed API using Swagger

The overview of each actor’s actions (such as write, read, update, and delete) on the data is depicted in figure 5. The data represents in the figure 5 is stored on the DHT component, while corresponding metadata is managed on the blockchain component. Please see the detail of the metadata structure in section 4.1.

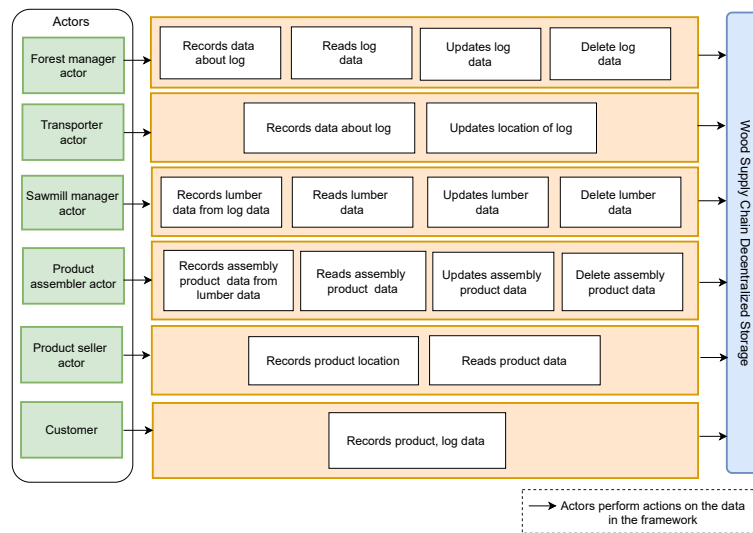


Fig. 5. High-level representation of actors actions on the data

4.3. Interaction via RESTful API

In this section, we detail the possible usage of our framework with a sequence diagram (Figure 6) that illustrates the interaction between an actor (e.g. a forest manager) and the framework using its RESTful API. We assume that every actor is already registered on the framework. An actor makes a 'POST' request to the /chain resource to write log data in the framework. Our solution assigns a unique data id (RFID_number) to the log that

enables authorized actors to trace the log in the chain. In the case of a successful response (HTTP code 201), it returns the links including the id in the response. Our framework stores the DHT key of this generated data in the metadata. Therefore, this DHT key points to the location of the log data on the DHT. The actor can use these links to perform further actions on the log data by sending another HTTP request as described in the links. To read the data, an actor would use the GET link that would call the `/chain/<id>` resource with method 'GET' to retrieve the representation of the log data. In the case of a successful response (HTTP code 200), our framework returns the representation of the log data. In case an actor wants to update their data, then they use the PUT link that makes a 'PUT' request to the (`/chain/<id>` resource). It will then write the new data against the same id. Then, a new metadata structure is created on the blockchain, and it contains the new DHT key of this updated data and the previous pointer of the old version of the data. Similarly, to delete the data, an actor may follow the DELETE link (`/chain/<id>` resource, method 'DELETE'). Our framework allows the authorized actor to delete specific data based on the id. After verifying the permission of the actor, it will delete the data. In this case, a new metadata structure is created on the blockchain that has a new DHT key with a NULL value.

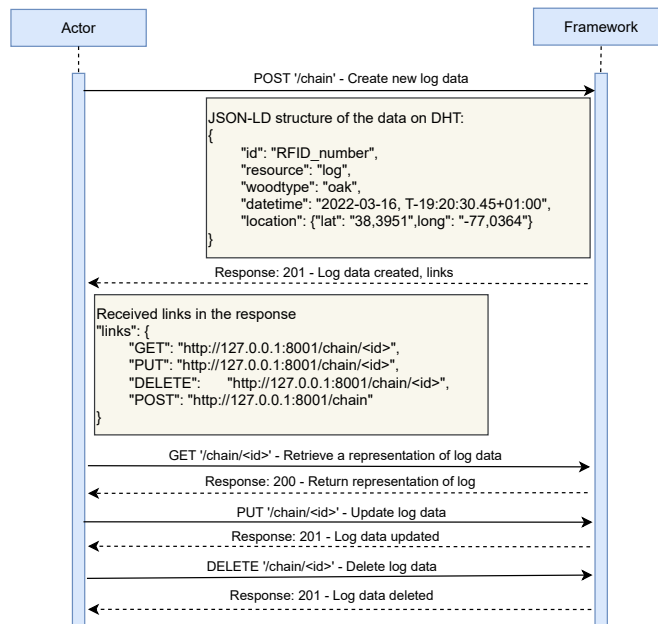


Fig. 6. Sequence diagram of possible actor interactions with the framework

4.4. Registration and Data Management

This section presents the proposed algorithms that support our solution including actor registration using designated REST APIs, data management on the decentralized storage and on the blockchain, and traceability algorithm to keep track of the data history.

– Actor Registration

Algorithm 1 describes the actor's connection or registration procedure to the proposed framework using our RESTful APIs. Once actor would successfully connect to the framework then they can perform different actions on the data such as write, read, update etc, and actors can also connect to other actors through HTTP requests. Each new actor needs to connect to the framework once to perform actions.

Firstly, the actor calls the `/peers` resource with 'GET' method to receive the available peer list (`pl`). After that, it calls the `/peers` resource 'POST' method to add its public key to the list of available peers and registers to the registry server. Then it sends a request to other peers to acknowledge the connected peer (`/peers` resource, 'POST' method). If the current actor is already in the list then it will be disconnected or removed from the peer list using the `/peers` resource with 'DELETE' method. Then it sends a request to other available peers to acknowledge the disconnected peer.

Algorithm 1 Actor registration algorithm

Input: `ca`: current actor

Output: `pl`: peer list, boolean value

- ▷ GET: HTTP verb GET request (constant)
- ▷ POST: HTTP verb POST request (constant)
- ▷ `pe`: endpoint of the peer (constant)
- ▷ `Req.Method`: identify request type (variable)
- ▷ `p`: peer in loop (variable)

```

1: if Req.Method == GET then
2:   return pl
3: end if
4: if Req.Method == POST then
5:   pl.Append(ca)
6:   for each p ∈ pl do
7:     RequestsPost(p(pe), ca)
8:   end for
9:   return true
10: end if
11: if Req.Method == DELETE then
12:   pl.Remove(ca)
13:   for each p ∈ pl do
14:     RequestsDelete(p(pe), ca)
15:   end for
16:   return true
17: end if

```

– Data Management on the DHT

The process to write or store the data including metadata and corresponding DHT key (a hash pointer of the encrypted data) is shown in Algorithm 2. Our proposed framework combine blockchain with a DHT in a way that allows authorized actors to write and update the data about their activities. For instance, if an actor has a role "data owner" and wants to store their log data such as:

```
{
  "id": "RFID_number",
  "resource": "log",
  "woodtype": "maple",
  "datetime": "2022-06-01, T-11:16:25.45+01:00",
  "location":
  {
    "lat": "14,2472",
    "long": "-43,2135"
  }
}
```

Then, `Authenticate(actor, role)` and `CheckPermission(actor, role, v)` verify that the current actor has the right permissions to store their data or not. The `CheckPermission` checks if the current actor has a role 'forest manager' then he is allowed to write, read, update, and delete their data in the decentralized platform.

After verifying the permission of the current actor, our framework provides different encryption methods (*em*) to encrypt the data before storing it on the decentralized ledger that ensures data security. An authorized actor is allowed to choose between *asymmetric em* and *symmetric em*. Asymmetric encryption is based on separate public and private keys. A public key is used to encrypt the data, while a corresponding private key is used to decrypt the data. In our motivating scenario, if a forest manager actor chooses *asymmetric em* then data will be encrypted with the data owner's public key, so later he can only access his data using his private key.

The authorized actor also has an option to choose *symmetric em* to encrypt the data, if he wants to enable other actors to read their data. A symmetric key is based on a single key to encrypt and decrypt the data. If the data owner chooses *symmetric em*, then our framework encrypts the data with the *symmetric* key and then this *symmetric* key will be encrypted with a data owner public key to protect the key from unauthorized actors.

Upon data read request, the data owner would encrypt this symmetric key using the requester's the public key to enable the authorized actors to read the data.

We store this encrypted symmetric key (*ek*) and encrypted data (*ed*) on the DHT. The *ed* stores on the DHT contains resource, woodtype, location (such as latitude and longitude) that shows the geographical location of the resource in the wood supply chain. Then, the function `FindLastTransaction` takes the data id such as (*r fid_number*) as input and returns previous pointer (*pp*) if it exists otherwise it returns 0. We store the metadata on the blockchain. The metadata includes DHT key (*dk*), *pp*, *datetime*, data owner id (*do id*), and data id (*r fid_number*).

Algorithm 2 Algorithm for the data write operation

Input: d: data, actor: current actor, role: role of the actor, v: HTTP verb POST, PUT, em: encryption method, pp: pointer of previous transaction when data is updated

Output: ed: encrypted data, encrypted symmetric key (ek) ▷ pk: public key of data owner (constant)

▷ doid: id of the data owner (constant)

▷ sk: symmetric key (variable)

▷ dht: variable to store the ed and ek

▷ dk: dht key points to the data in dht (variable)

▷ rfid_number: data id (variable)

▷ datetime: timestamp (variable)

▷ pp: previous pointer (variable)

```

1: if Authenticate(actor, role) then
2:   if CheckPermission(actor, role, v) then
3:     if em == true then                                ▷ if true we use asymmetric encryption)
4:       ed ← Encrypt(d, pk)
5:     else                                             ▷ if false we use symmetric encryption)
6:       encrypd ← Encrypt(d, sk)
7:       ek ← Encrypt(sk, pk)
8:       ed ← encrypd, ek
9:     end if
10:    dk ← Digest(ed)
11:    dht ← SetValue(ed)
12:    pp ← FindLastTransaction(rfid-number)
13:    AddTransaction(dk, pp, datetime, doid, rfid_number)
14:  end if
15: end if

```

– Data Management on the Blockchain

As an extension to the work in [1], we propose a metadata structure that manages the pointer and connects the different values attached to a specific piece of data to maintain its history. For example, a forest manager actor, as a data owner, would write a log information such as:

```

{
  "id": "RFID_number",
  "resource": "log",
  "woodtype": "maple",
  "datetime": "2022-05-03, T-10:12:21.45+01:00",
  "location":
  {
    "lat": "13,2351",
    "long": "-15,5142"
  }
}

```

In this case, the proposed solution stores the DHT key as a new pointer of the log data in the metadata. Later, the data owner can access the data using a data id (RFID_number

of the corresponding data). An actor can update some parts of the data against the same data id such as:

```
{
  "id": "RFID_number",
  "resource": "log",
  "woodtype": "maple",
  "datetime": "2022-08-06, T-14:16:23.45+01:00",
  "location":
  {
    "lat": "11,2256",
    "long": "-21,1525"
  }
}
```

Our solution allows the data owner to perform different operations (such as update, read and delete) on their data for the specific RFID_number. In case of data update, new metadata will be generated on the blockchain that includes a new DHT key of the updated data and the previous pointer that refers to the previous version of the data that is stored on the DHT (illustrated in Algorithm 2). Therefore, the DHT key of the previous version of the transaction becomes the previous pointer which is stored in the new metadata. The proposed metadata structure also stores the datetime of the updated data. This way if the data owner wants to see their transactions history, then the function `FindLastTransaction(did)` returns the recent version of the transaction against this data id as RFID_number containing the DHT key of new data and previous pointer of the updated data. This way an actor can access their update history. To read the data, an authorized actor can decrypt and access their data in the decentralized platform. In case, if data is encrypted with the data owner's public key then a data owner can use their private key to decrypt and read the data. If the data is encrypted with a symmetric key then the authorized actor first decrypts the symmetric key using their private key and then this decrypted symmetric key will be used to access the data that is stored on the DHT. Similarly, an authorized actor can delete their data against a specific RFID_number, then a new transaction is created on the blockchain that includes a new metadata structure. This metadata includes a new DHT key with a NULL value.

– Traceability

We propose an solution that maintains data id references to ensure traceability. It enables actors to verify the origin of the final product in the chain. Our solution assigns a unique data id (such as RFID_number) to the log, lumber, and product. We assume that, RFID chips are inserted into the logs and then into the lumbers and final products. The following code shows the log data in JSON format such as.

```
{
  "id": "RFID_number",
  "resource": "log",
  "woodtype": "maple",
  "datetime": "2022-05-10, T-13:10:20.45+01:00",
  "location":
  {
```

```

        "lat": "25,1324",
        "long": "-45,1326"
    }
}

```

A log produces different pieces of lumbers and each lumber has unique id as RFID_number. The following code shows the lumber data.

```

{
    "id": "RFID_number",
    "resource": "lumber",
    "datetime": "2022-05-13, T-14:12:23.45+01:00",
    "location":
    {
        "lat": "12,2425",
        "long": "-23,1526"
    },
    "log":
    {
        "id": "RFID_number"
    }
}

```

The data described above contains a reference id (RFID_number) of the log that was turned into lumbers. The different pieces of lumbers participate to build a final product such as wooden furniture. The following is a JSON representation of product data.

```

{
    "id": "RFID_number",
    "resource": "product",
    "datetime": "2022-06-02, T-16:14:26.45+01:00",
    "location":
    {
        "lat": "52,5323",
        "long": "-24,3316"
    },
    "lumber":
    {
        "id": "RFID_number"
    }
}

```

The product data represented above contains an id reference of lumber that was used to build it. This way an authorized actor can verify the origin of the wooden product and can identify where it comes from. The process to trace the data and verifies the product origin in the wood supply chain is shown in Algorithm 3. For instance, a customer buys a wooden product such as a bed and he wants to trace this product. Then, he can use the product id as a data id (such as RFID_number) to keep track of their origin. The proposed algorithm enables actors to trace the product's origin using the data id's references discussed above.

In Algorithm 3, the `did` is an `RFID_number` of the item in the wood supply chain, and data (e.g location) of the item changes for the same `did`. Therefore, we can have multiple transactions on the blockchain against this `did`. Whenever the location of the item would change then new metadata of the same `did` will be recorded on the blockchain, and the corresponding data is stored to the DHT. The `FindLastTransaction` function returns the last or recent transaction `t` of this `did`, which is a `RFID_number`. For instance, if we have `did` of the log then it finds the last transaction of this log.

This transaction `t` has the metadata that contains DHT key that points to the data recorded on the DHT. The function `CheckPermission` verifies if the current data requester is authorize to read the data or not depending on their role and HTTP verbs permission 'GET'. Then, the function `GetReferences` has the `t` as input and takes the `did` of the items. After that, it gets the previous references of this `did`. For example, if we have a input `did` as product id then it finds the previous references such as `RFID_number` of the lumbers. Then, it checks items (e.g lumbers) in the list and add items (e.g lumbers references) in the output list (`o`). Then, the `Traceability` function takes `i` such as lumber as input and call recursively to find out the log and add them in the list `o`. In case the list `o` is empty it is returned anyway, and it means that the log does not contain any previous reference.

Algorithm 3 Traceability algorithm

Input: `did`: data id (DHT key)

actor: requester actor, role: requester role, v: HTTP verb GET

Output: `o`: DHT keys of tracked items

▷ `l`: items list (variable)

```

1: l ← ∅
2: t ← FindLastTransaction(did)
3: if CheckPermission (actor, role, v) then
4:   l ← GetReferences (t)
5:   if l ≠ ∅ then
6:     o ← ∅
7:     for each i ∈ l do
8:       o.Append(i)
9:       o.Append(Traceability(i))
10:    end for
11:    return o
12:   end if
13: end if
14: return ∅

```

5. Results and Evaluation

This section presents the results and performance evaluation of the proposed decentralized data storage framework. The evaluation framework is discussed in Section 5.1. The

security and privacy analysis are presented in Section 5.2. Section 5.3 discusses the performance evaluation of our proposed framework.

5.1. Evaluation Framework

To implement and evaluate the performance of our framework, we used Python 3.7.0. We used a Python library⁴ to implement a blockchain to store the DHT key and metadata. We implemented a DHT using the Kademlia library⁵, which allows to store and get data linked with a given key on the peer-to-peer network. We used the cryptography RSA library to generate private/public keys and encrypt/decrypt the data. We conducted experiments and evaluated our framework on a 64-bit Windows operating system, Core i7 1.80 GHz processor, and 16 GB RAM.

5.2. Privacy and Security Analysis

The proposed solution supports data privacy and enables data owners to own and control their data in a decentralized platform. Our check permission method prevents unauthorized actors to perform operations on data such as data write, read, update, and delete. In addition, to protect privacy-sensitive data from unauthorized access, our framework provides multi layers of encryption to ensure privacy and security. The data stored on the DHT are encrypted before uploading. Even if an unauthorized actor gains access to the DHT nodes then they can only see the cipher texts and cannot achieve any information about the data. Moreover, in our solution, we used blockchain and DHT because of their decentralized and distributed design. This can solve the single-point failure issue, and ensures data replication and availability. We analyzed and evaluated the security of our framework under the following threats:

– Linking attack

A linking attack happens when the attacker tries to link various transactions or data with the corresponding public key. In our design, we use different encryption mechanisms to encrypt the data, such as the data owner's public key, symmetric key, and requester's public key. We generate public, private, and symmetric keys at run-time according to the encryption method chosen by the actor. To secure the symmetric key from unauthorized access, our framework encrypts the symmetric key with the data owner's public key and stores it on the DHT. This way only the authorized user is allowed to use this symmetric key to decrypt and access their data. For this reason, an attacker cannot link different transactions to the same public key, because our solution encrypts the data using different encryption mechanisms and public keys.

– Eavesdropping attack

In an eavesdropping attack, an attacker tries to listen to privacy-sensitive information in the network. To protect against this attack, we encrypt privacy-sensitive data with the requester's public key upon data read request. This way only authorized actors can access and read the data using their private key.

⁴ https://github.com/satwikkansal/python_blockchain_app/tree/ibm_blockchain_post

⁵ <https://github.com/bmuller/kademlia>

– Spoofing attack

A spoofing attack happens when a malicious actor uses the ID of another actor and tries to access the data. In our framework, a malicious user cannot spoof the ID of another actor because they could not spoof its private key. In our solution, each actor has a private key that is kept secret and not shared with others.

– Modification attack

A modification attack occurs when an attacker tries to change the data content. In our framework design, we allow data owners to encrypt the data using their public key and store the corresponding pointer on the blockchain. Our proposed metadata design keeps the track of data entry date and time to recognize the changes in the data. An attacker cannot modify the data because data can only be decrypted with a data owner's private key that is kept secret by the data owner.

5.3. Performance Evaluation

We evaluated the results according to time consumption and scalability with respect to the number of peers. We computed the time consumption of the proposed solution according to the following parameters: actor's check permission, data encryption/decryption using asymmetric or symmetric techniques, DHT access, and blockchain access. We observed time consumption while performing data write, update, read, delete, and traceability operations. Figure 7 and 8 show the time consumption of the different parts of our solution, respectively using symmetric encryption and asymmetric encryption.

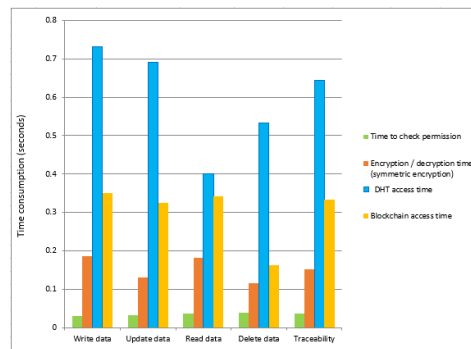


Fig. 7. Time consumption using symmetric encryption

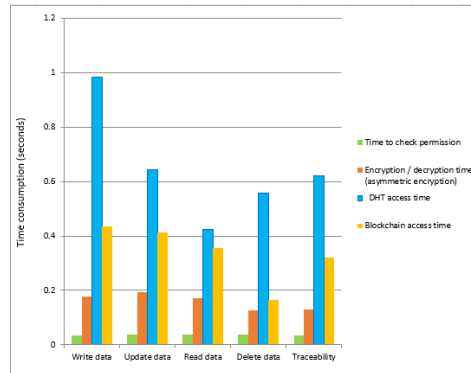


Fig. 8. Time consumption using asymmetric encryption

The general trend of our measurements shows that DHT access takes most of the time needed, followed by blockchain access, encryption/decryption and then permission check, which makes sense since the DHT deals with data storage and is therefore I/O-bound. We believe however that some low-level optimization is performed at this stage (see the scalability tests and discussion).

In general, the usage of symmetric or asymmetric encryption does not impact the solution much, except a slight increase of time consumption if asymmetric encryption is used. We make sense of these results by acknowledging the higher number of keys and costly computation that are needed when using asymmetric cryptography.

The most time-consuming operation is, without surprise, the write operation, since it requires the most from the system. Second comes the update operation which is similar to a write except it is already related to an existing piece of data. Third comes traceability, which does not modify the existing data but requires following the history of different pieces of data. Finally, the delete operation is less costly, and the read operation only consists in resolving the DHT pointer and if granted, fetching the data.

Moreover, we tested the scalability of our solution with a growing number of actors 1, 100, 200, 300, 400 and obtained a reasonable performance with 400 actors (please note that increasing the number of actors to more than 400 would lead to additional synchronization problem, which would slow down the speed and performance. These problems are out of the scope of this paper.) The HTTP requests will be only partially processed in parallel, since they share the CPU time, and we tested our prototype with a quad-core CPU. In our solution, actors are the same as blockchain nodes and DHT nodes. We tested our solution with a number of 400 actors which are considered as 400 nodes.

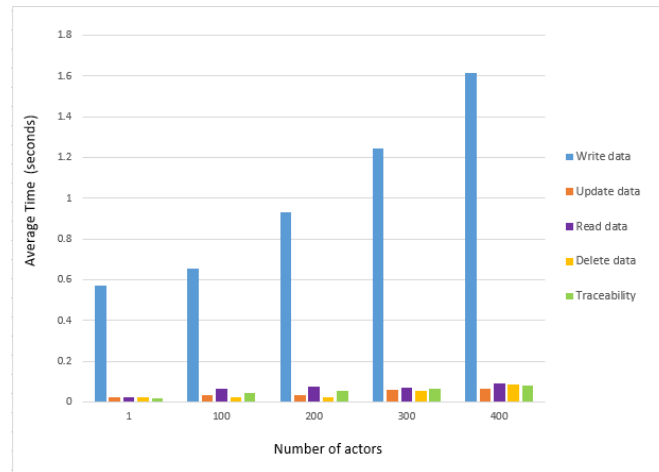


Fig. 9. Average time consumption under different number of actors

We calculated the average time consumption of our prototype with an increasing number of actors. The actor registration operation is performed only once for 1, 100, 200, 300, and 400 actor and the time costs is 0,0034 seconds, 0,0039 seconds, 0,0041 seconds, 0,0046 seconds, and 0,0049 seconds respectively. Therefore, we tested our prototype 100 times for all operations such as write data, update data, read data, delete data, and traceability. After that, we calculated the average time, Standard Deviation (SD), minimum (min), and maximum (max) values in seconds. Figure 9 depicts the average time consumption between a different number of actors, and detailed results statistics are summarized in Table 2..

As we can see from Figure 9 and Table 2, for the case of 1 actor, write data gives an average of 0,5712 seconds which is less than the average time of data write for 100, 200, 300, and 400 actors. The update data has an SD of 0,0211 seconds which is close to the SD of update data for the case of 200 actors. The data read provides an max value of 0,0456 seconds which is less than the may value of read data for the case of 100, 200, 300, and 400 actors. The delete data takes an average time of 0,0214 seconds which is close to the average time of delete data for 100 and 200 actors. The traceability data operation has a min value of 0,0112 seconds and a max value of 0,0312 seconds.

Table 2. Detailed results under different number of actors

Number of actors	Data operations	Average Time	St Deviation	Minimum	Maximum
1	Write data	0,5712	0,4321	0,4635	0,6564
	Update data	0,0224	0,0211	0,0221	0,0412
	Read data	0,0254	0,0113	0,0124	0,0456
	Delete data	0,0214	0,0212	0,0213	0,0434
	Traceability	0,0201	0,0101	0,0112	0,0312
100	Write data	0,6552	0,5352	0,5432	0,7681
	Update data	0,0346	0,0321	0,0334	0,0571
	Read data	0,0632	0,0512	0,0542	0,0724
	Delete data	0,0233	0,0221	0,0223	0,0342
	Traceability	0,0464	0,0413	0,0421	0,0641
200	Write data	0,9325	0,6215	0,6316	1,8622
	Update data	0,0356	0,0241	0,0256	0,0392
	Read data	0,0738	0,0635	0,0641	0,0956
	Delete data	0,0215	0,0153	0,0171	0,0516
	Traceability	0,0521	0,0439	0,0472	0,0695
300	Write data	1,2455	0,7529	0,7924	1,9372
	Update data	0,0573	0,0543	0,0561	0,0635
	Read data	0,0713	0,0537	0,0655	0,0836
	Delete data	0,0531	0,0457	0,0461	0,0734
	Traceability	0,0636	0,0531	0,0571	0,0913
400	Write data	1,6121	1,3163	1,3223	2,4692
	Update data	0,0626	0,0551	0,0569	0,0931
	Read data	0,0911	0,0815	0,0857	0,2419
	Delete data	0,0882	0,0731	0,0765	1,4271
	Traceability	0,0791	0,0682	0,0693	0,0975

For the case of 100 actors, the write operation gives an average of 0,6552 seconds which is slightly higher than the average time to write data with 1 actor. The update operation gives an SD time of 0,0321 seconds which is slightly higher than the SD to update data with 1 actor and 200 actors. The read operation has a SD of 0,0512 seconds which is slightly close to the SD of read data for 300 actors. The delete operation gives a min value of 0,0223 seconds which is close to the min value for 1 actor. The traceability algorithm has an average time of 0,0464 seconds which is less as compared to the average time for 200, 300, and 400 actors.

Similarly, with the number of 200 actors, the average time to write data is 0,9325 seconds which is slightly higher than the average time to write data for the number of 1 and 100 actors. The update operation provides an SD of 0,0241 seconds which is less than the SD of update data for the case of 100 actors. The read operation gives an average time of 0,0738 seconds which is slightly close to the average time to read data for the case of 300 actors. The delete operation has a min value of 0,0171 seconds which is less than the min value for 1, 100, 300, and 400 actors. The traceability data operation gives a max value of 0,0695 seconds which does not show much difference from the max value of 100 actors.

For the case of 300 actors, the write data operation gives an average of 1,2455 seconds which is slightly higher as compared to the average time to write data for 200 actors. The

update gives an SD value of 0,0543 seconds which is close to the SD for the number of 400 actors. The read data operation gives a max value of 0,0836 seconds which is less as compared to the max value to read data for the number of 200 and 400 actors. The delete operation provides an SD of 0,0457 seconds which is less than the SD for 400 actors. The traceability takes an average time of 0,0636 seconds which is higher than the average time for 1, 100, and 200 actors.

For the number of 400 actors, the average time to write data is 1,6121 seconds which is higher than the average time to write data for 1, 100, 200, and 300 actors. The update data operation gives an SD of 0,0551 seconds which is close to the SD value of update data for 300 actors. The read data provides a min value of 0,0857 seconds and a max value of 0,2419 seconds. The average time to delete data operation is slightly higher than the average time to update operation for 1, 100, 200, and 300 actors. The traceability provides a max value of 0,0975 seconds which is close to the max value for 300 actors.

We interpret the reasonable increase in time consumption despite the large increase in the number of actors as a consequence of the efficiency of DHT access, which is known to be logarithmic, combined with a number of low-level optimization from the Python language, together with operating system and hardware optimization mechanisms related to data management and process execution.

Overall, our experimental results demonstrates that the proposed solution is scalable and able to manage many actors at the same time. The results show that each operation take average time less than 1 minute, while increasing the number of actors, therefore, we can conclude that our solution is acceptable for the end user.

6. Conclusion

In this paper, we present a decentralized data storage and access framework that ensures data security, privacy, and mutability in wood supply chain scenario. The proposed framework integrates blockchain technology with DHT, a role-based access control model, and different types of encryption techniques. Our solution allows authorized actors to write, read, delete, update their data and manage transaction history on a decentralized system. The proposed traceability algorithm enables authorized actors to trace the product data in a decentralized ledger. We provided a critical comparative analysis of our work with existing solutions to show the research gap. The main limitations of existing solutions are a single point of failure, data mutability, and public availability of the data.

Our prototype design is flexible to expand and can be easily reused for different application domains such as medicine, agriculture, etc. We discussed the security and privacy analysis of our proposed solution and evaluate its performance in terms of time cost and scalability. The experimental results show that the proposed solution is scalable, secure, and achieves an acceptable time cost.

In future work, we plan to test our framework with different real-life use-cases and enhance data access with semantic annotation to identify data concepts that are stored and in turn exploit this information to drive the RBAC model. We believe the richness of description logic can contribute to better fine-grained access control and facilitate data management. Another step forward relates to the possibility to adapt semantically annotated data to specific local interpretation depending on the context of the query issuer - for example, converting data units between countries.

Acknowledgments. The authors gratefully acknowledge the European Commission for funding the InnoRenew project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Regional Development Fund). They also acknowledge the Slovenian Research Agency ARRS for funding the project J2-2504.

References

1. Ali, S., Wang, G., White, B., Cottrell, R.L.: A blockchain-based decentralized data storage and access framework for pinger. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). pp. 1303–1308. IEEE (2018)
2. Aslam, S., Mrissa, M.: A restful privacy-aware and mutable decentralized ledger. In: European Conference on Advances in Databases and Information Systems. pp. 193–204. Springer (2021)
3. Aslam, S., Tošič, A., Mrissa, M.: Secure and privacy-aware blockchain design: Requirements, challenges and solutions. *Journal of Cybersecurity and Privacy* 1(1), 164–194 (2021)
4. Benisi, N.Z., Aminian, M., Javadi, B.: Blockchain-based decentralized storage networks: A survey. *Journal of Network and Computer Applications* 162, 102656 (2020)
5. Chakravorty, A., Rong, C.: Ushare: user controlled social media based on blockchain. In: Proceedings of the 11th international conference on ubiquitous information management and communication. pp. 1–6 (2017)
6. Domingue, J., Third, A., Ramachandran, M.: The fair trade framework for assessing decentralised data solutions. In: Companion Proceedings of The 2019 World Wide Web Conference. pp. 866–882 (2019)
7. de Figueiredo, S., Madhusudan, A., Reniers, V., Nikova, S., Preneel, B.: Exploring the storj network: a security analysis. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing. pp. 257–264 (2021)
8. Hassanzadeh-Nazarabadi, Y., Küpçü, A., Özkasap, Ö.: Lightchain: A dht-based blockchain for resource constrained environments. arXiv preprint arXiv:1904.00375 (2019)
9. He, Q., Yan, J., Yang, Y., Kowalczyk, R., Jin, H.: A decentralized service discovery approach on peer-to-peer networks. *IEEE Transactions on Services Computing* 6(1), 64–75 (2011)
10. Hei, Y., Liu, Y., Li, D., Liu, J., Wu, Q.: Themis: An accountable blockchain-based p2p cloud storage scheme. *Peer-to-Peer Networking and Applications* 14(1), 225–239 (2021)
11. Huang, H., Zhou, X., Liu, J.: Food supply chain traceability scheme based on blockchain and epc technology. In: International Conference on Smart Blockchain. pp. 32–42. Springer (2019)
12. Khamphakdee, N., Benjamas, N., Saiyod, S.: Performance evaluation of big data technology on designing big network traffic data analysis system. In: 2016 Joint 8th International Conference on soft computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS). pp. 454–459. IEEE (2016)
13. Kumar, M.V., Iyengar, N.: A framework for blockchain technology in rice supply chain management. *Adv. Sci. Technol. Lett* 146, 125–130 (2017)
14. Legault, M.: A practitioner's view on distributed storage systems: Overview, challenges and potential solutions. *Technology Innovation Management Review* 11(6), 32–41 (2021)
15. Li, W., Andreina, S., Bohli, J.M., Karame, G.: Securing proof-of-stake blockchain protocols. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology, pp. 297–315. Springer (2017)
16. Longo, F., Nicoletti, L., Padovano, A., d'Atri, G., Forte, M.: Blockchain-enabled supply chain: An experimental study. *Computers & Industrial Engineering* 136, 57–69 (2019)
17. Lykousas, N., Koutsokostas, V., Casino, F., Patsakis, C.: The cynicism of modern cybercrime: Automating the analysis of surface web marketplaces. arXiv preprint arXiv:2105.11805 (2021)

18. Marr, B.: How much data do we create every day? the mind-blowing stats everyone should read. *forbes*. may, 21 2018 (2018)
19. Maymounkov, P., Mazieres, D.: Kademlia: A peer-to-peer information system based on the xor metric. In: *International Workshop on Peer-to-Peer Systems*. pp. 53–65. Springer (2002)
20. Mikroyannidis, A., Third, A., Domingue, J.: A case study on the decentralisation of life-long learning using blockchain technology. *Journal of Interactive Media in Education* 2020(1) (2020)
21. Moser, M.: Anonymity of bitcoin transactions. In: *Münster Bitcoin Conference (MBC)*, Münster, Germany (July 2013)
22. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* p. 21260 (2008)
23. Nakamoto, S., Bitcoin, A.: A peer-to-peer electronic cash system. Bitcoin.–URL: <https://bitcoin.org/bitcoin.pdf> 4 (2008)
24. Ølnes, S., Ubacht, J., Janssen, M.: Blockchain in government: Benefits and implications of distributed ledger technology for information sharing (2017)
25. Pazaitis, A., De Filippi, P., Kostakis, V.: Blockchain and value systems in the sharing economy: The illustrative case of backfeed. *Technological Forecasting and Social Change* 125, 105–115 (2017)
26. Podduturi, P.R., Maco, T., Ahmadi, P., Islam, K.: Rfid implementation in supply chain management. *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)* 12(2), 34–45 (2020)
27. Ramachandran, M., Chowdhury, N., Third, A., Domingue, J., Quick, K., Bachler, M.: Towards complete decentralised verification of data with confidentiality: Different ways to connect solid pods and blockchain. In: *Companion Proceedings of the Web Conference 2020*. pp. 645–649 (2020)
28. Shafagh, H., Burkhalter, L., Hithnawi, A., Duquennoy, S.: Towards blockchain-based auditable storage and sharing of iot data. In: *Proceedings of the 2017 on Cloud Computing Security Workshop*. pp. 45–50 (2017)
29. Shrestha, A.K., Vassileva, J., Deters, R.: A blockchain platform for user data sharing ensuring user control and incentives. *Frontiers in Blockchain* 3, 48 (2020)
30. da Silva, D.L., Corrêa, P.L.P., Najm, L.H.: Requirements analysis for a traceability system for management wood supply chain on amazon forest. In: *2010 Fifth International Conference on Digital Information Management (ICDIM)*. pp. 87–94. IEEE (2010)
31. Sirkka, A.: Modelling traceability in the forestry wood supply chain. In: *2008 IEEE 24th International Conference on Data Engineering Workshop*. pp. 104–105. IEEE (2008)
32. Swan, M.: Blockchain thinking: The brain as a decentralized autonomous corporation [commentary]. *IEEE Technology and Society Magazine* 34(4), 41–52 (2015)
33. Tian, F.: An agri-food supply chain traceability system for china based on rfid & blockchain technology. In: *2016 13th international conference on service systems and service management (ICSSSM)*. pp. 1–6. IEEE (2016)
34. Toyoda, K., Mathiopoulos, P.T., Sasase, I., Ohtsuki, T.: A novel blockchain-based product ownership management system (poms) for anti-counterfeits in the post supply chain. *IEEE access* 5, 17465–17477 (2017)
35. Tschorsch, F., Scheuermann, B.: Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials* 18(3), 2084–2123 (2016)
36. Tzoulis, I., Andreopoulou, Z.: Emerging traceability technologies as a tool for quality wood trade. *Procedia Technology* 8, 606–611 (2013)
37. Voronchenko, K.: Do you need a blockchain? Supervised by Ivo Kubjas 22 (2017)
38. Westerkamp, M., Victor, F., Küpper, A.: Blockchain-based supply chain traceability: Token recipes model manufacturing processes. In: *2018 IEEE International Conference on Internet of*

- Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 1595–1602. IEEE (2018)
39. Wilkinson, S., Boshevski, T., Brandoff, J., Buterin, V.: Storj a peer-to-peer cloud storage network. <https://www.storj.io/storj2014.pdf> (2014)
 40. Xu, L., Shah, N., Chen, L., Diallo, N., Gao, Z., Lu, Y., Shi, W.: Enabling the sharing economy: Privacy respecting contract based on public blockchain. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. pp. 15–21 (2017)
 41. Zheng, X., Lu, J., Sun, S., Kiritsis, D.: Decentralized industrial iot data management based on blockchain and ipfs. In: IFIP International Conference on Advances in Production Management Systems. pp. 222–229. Springer (2020)
 42. Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H.: Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* 14(4), 352–375 (2018)
 43. Zyskind, G., Nathan, O., et al.: Decentralizing privacy: Using blockchain to protect personal data. In: 2015 IEEE Security and Privacy Workshops. pp. 180–184. IEEE (2015)

Sidra Aslam has completed her Ph.D. in Computer Science from the University of Primorska, Slovenia in June 2022. She published research papers in scholarly journals and conference proceedings. She worked 3 years as an assistant researcher at the international company InnoRenew CoE, Izola, Slovenia, where she was involved in different European industrial projects. She worked as a teaching assistant and taught a master course at the University of Primorska, Koper, Slovenia. In 2021, she was awarded grant for Short Term Scientific Mission (STSM) and worked with the European company ‘Advanced Building and Urban Design (ABUD)’, Budapest, Hungary. In 2015, she received the best research paper award at the National Software Engineering Conference (NSEC), IEEE in Pakistan.

Michael Mrissa received his PhD in computer science from the University of Lyon, France, in 2007. His main areas of research are related to distributed systems and include service-oriented computing, semantic web, privacy, web of things. He has authored 80+ peer-reviewed publications in international conferences and journals, and he has been involved in several European, French and Slovenian research projects. He is currently researcher in the ICT research group of the InnoRenew CoE. He also holds a full professor position at the Faculty of Mathematics, Natural sciences and Information Technologies of the University of Primorska.

Received: January 10, 2022; Accepted: November 14, 2022.

Detecting and Analyzing Fine-Grained User Roles in Social Media*

Johannes Kastner and Peter M. Fischer

University of Augsburg
Universitätsstr. 6a, 86159 Augsburg, Germany
{johannes.kastner, peter.m.fischer}@uni-a.de

Abstract. While identifying specific user roles in social media -in particular bots or spammers- has seen significant progress, generic and all-encompassing user role classification remains elusive on the large data sets of today’s social media. Yet, such broad classifications enable a deeper understanding of user interactions and pave the way for longitudinal studies, capturing the evolution of users such as the rise of influencers.

Studies of generic roles have been performed predominantly in a small scale, establishing fundamental role definitions, but relying mostly on ad-hoc, data set-dependent rules that need to be carefully hand-tuned.

We build on those studies and provide a largely automated, scalable detection of a wide range of roles. Our approach clusters users hierarchically on salient, complementary features such as their actions, their ability to trigger reactions and their network positions. To associate these clusters with roles, we use supervised classifiers: trained on human experts on completely new media, but transferable on related data sets. Furthermore, we employ the combination of samples in order to improve scalability and allow probabilistic assignments of user roles.

Our evaluation on Twitter indicates that a) stable and reliable detection of a wide range of roles is possible b) the labeling transfers well as long as the fundamental properties don’t strongly change between data sets and c) the approaches scale well with little need for human intervention.

Keywords: Social Media, User Role Detection, Classification, Clustering, Supervised Learning Unsupervised Learning.

1. Introduction

As a significant share of personal and public life is shifting to social media platforms, they are growing in terms of user number and activity. The interaction of users can have a profound affect in both social media (eliciting reactions, spreading information) as well as in the real world (driving popular sentiment, affecting political decisions).

While the number of users is huge, their behavior and impact on others are clearly not uniform, thus motivating thorough studies. The need to counter malicious activities has driven many of those studies, providing tools to detect -among others- bots, bullies, spammers and fake news providers in large numbers and with little human intervention.

Yet, these are rather blunt, limited tools that do not provide a deeper understanding of the rich activities and varied user groups present in social media. Studies that do such

* This is an extended version of https://doi.org/10.1007/978-3-030-85082-1_23

wider and fine-grained investigations on user behavior are indeed performed, but typically require a significant amount of human involvement to organize the data and interpret the results. Thus, they are mostly done in an academic environment on limited sets of users representing coarse-grained roles.

Machine-aided identification of user roles in social media at scale and speed promises interesting insights on their prevalence and impact, allowing to capture different aspects of activity, social media usage, popularity and influence: Not every user that generates large numbers of tweets has malicious purposes but also could only be sharing relevant information to others using his/her network. Forwarding information may be driven by the desire to share relevant information or to endorse certain positions. Not every user that has a large number of followers is a star or influencer, as they may lack in activity. The same social media may be used for information dissemination, but also for conversations or restricted types of feedback.

Automatically recognizing such fine-grained roles provides another benefit, as user classifications can now be performed over longer periods of time. Such a stable recognition provides the means to explain how individual users and communities evolve over time.

We propose a method that combines unsupervised learning to discover fine-grained classes of users over a wide range of features with supervised learning - generalizing expert knowledge from manually labeled reference data to new data sets, mapping role candidates to well-known roles or identifying new roles.

The paper provides the following contributions:

- Our method covers both learning the structure of user groups as well as assigning suitable labels.
- A study on large, complementary data sets shows that both recognizing and transferring roles is feasible over longer time periods or topic variations.
- The classification hierarchy and the cluster metrics support (also iterative) human review, so that identification itself requires little human intervention.
- Sampling strategies provide means to scale the method to large data set as well as provide insights on the certainty and stability of role assignment.

The remainder of this extended paper is structured as follows: In Section 2 we discuss related work. We introduce our methodology in Section 3.2 and provide more details on structure discovery and labeling in Sections 4 and 5, respectively. After an extensive evaluation (Section 6), we conclude the paper.

2. Related Work

Clearly, identifying user roles has been one of the textbook examples of classifier algorithms, yet the application to social networks has been limited to particular aspects. Often, the studies focus on detecting specific roles or describing only a small number of coarse-grained classes. Considering the negative dynamics of many social networks, most researchers focus on identifying specific malicious users, example include: detection of bots [2] or spammers [14], identification of aggressors in the context of cyber bullying [1,11] or –of particular interest recently– discovery of instigators and spreaders of fake news [16,7]. In contrast, our goal is to comprehensively assign all users to roles. Multi-role approaches such as Varol et al. [18], Rocha et. al [6] and Lazaridou et. al [13] limit

themselves to identify a small number (often 3-5) of major, coarse-grained groups, roughly corresponding the upper levels of our detection hierarchy. Du et. al [5] provide a somewhat higher number of rules (still lower than ours), but only give generic descriptions. All of these previously mentioned methods are constrained on just detecting the structure by unsupervised learning: clustering via K-Means [13], EM [6] or via topic models [5], leaving the analysis entirely to human experts. In terms of classification, Varol et al. [18] fully rely on such human expertise, using similarity matrices and handcrafted rules. In contrast, qualitative works like Tinati et. al [17] or Java et. al [10] provide a comprehensive overview on fine-grained roles and their semantics, but consider only general rules on how to detect them. An interesting, complementary direction is the work on content communities/web forum, often exploring complex temporal models, e.g., [8]. It should be noted that all of these works (with the exception of [5] (Weibo, 12K users), [11] (Instagram, 18K users), and [8] (Stack Overflow)) solely rely on Twitter due to the limited availability of data from other services. A recent work by Hacker et al. [9] comes closest to our approach, while tackling the -more constrained- problem of user role identification in Enterprise Social Networks. Like our work, it follows a process-based approach involving and aiding human analysts in discovering and interpreting user roles. It applies a wide set of user features and employs clustering to identify user group candidates. As the authors themselves recognize, their problem is less challenging due to the smaller scale and better observability (allowing for more expressive metrics) and more well-defined and less context-dependent roles. Furthermore, we provide a more extensive process by incorporating a classifier to perform knowledge transfer of user role between data sets and employ a sample combination strategy for probabilistic roles assignment and better scalability.

While probabilistic clustering is well-established for centroid methods [4] and recent work presents probabilistic density-based methods with constraints (Lasek et al. [12]), hierarchical clustering is not covered well regarding probabilistic assignment.

3. Research Questions & Approach

Before introducing the main aspects of our approach we want to provide some basic assumptions and definitions:

In the scope of this work, we consider *social media* that allows users to publish content (which we call messages) and organize themselves in structures (networks, groups). These networks enable rich means of interaction on top of both content and structure, such as resharing or conversations. As a consequence, we do not consider media that is purely driven by opaque algorithms such as TikTok.

Users are all types of distinct entities that may visibly interact with the social media, including both humans and algorithms/bots.

A *data set* in our model is a set of messages by users stemming from a single social media, often corresponding to specific events or topics. These messages are recorded and extracted from a social network, currently mostly Twitter for due to its open nature.

As the related work only describes instances of user roles, but not the concept of a role itself, we use the following, basic definition:

A *user role* is a group of users that share similar feature values and are well separated from other groups. The features gather salient properties of users and allow a meaningful categorization, typically capturing behavior and position in the network/media. Groups

constitute roles if they are present in sufficient number within a data set and reoccur over multiple data sets.

3.1. Research Questions

Motivated by the introduction in Section 1, we phrase three questions in order to classify diverse user roles in large data sets:

1. To which extent can clusters of users be utilized to sensibly detect user roles in social media and build a classifier to (semi-)automatically label them?
2. Can this approach be applied individually over a wide variety of data sets, currently stemming from the same social media?
3. Can the knowledge on roles be transferred from a (set of) well-understood data set(s) to new data sets?

3.2. Approach

To answer these questions, we introduce our main approach using a high-level overview of our model, which can be seen in Fig. 1.

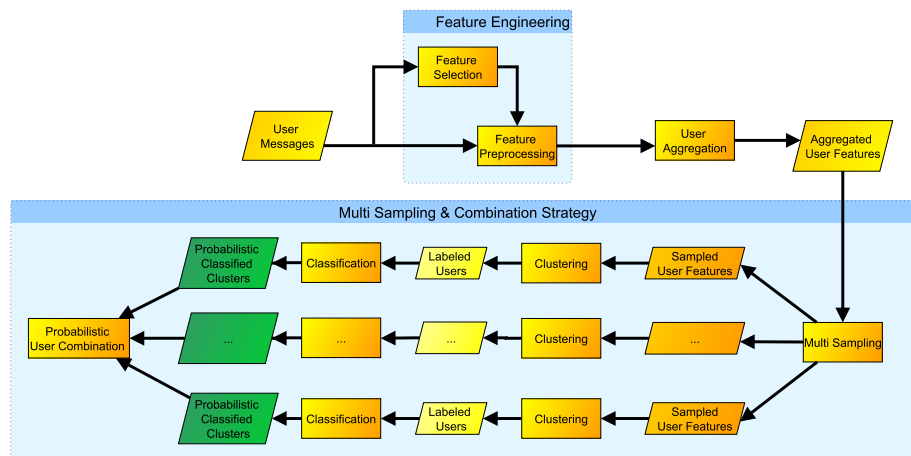


Fig. 1. Flowchart of the approach

Our process starts with a *Raw Data Set*, consisting of messages that were recorded from a social media. In the next step (*Feature Engineering*) we determine the relevant features to capture the various properties of users. As it is our goal to analyze large-scale data sets, it is essential that features are based on widely available data (e.g. not requiring the full social graph) and can be computed at scale, not requiring high complexity and runtime. Clearly, this needs to be repeated for each distinct social media, but - as our experience shows so far - only minor adaption is needed for data from the same social media.

After choosing the features, they need to be *preprocessed* to suit the requirements of clustering and classification methods, including but not necessary limited to outlier removal and normalization/standardization.

In order to solve the competing goals of scalability, minimal human effort as well explainability for fine-grained roles, we devised a multi-sampling strategy that allows us to apply precise and flexible, yet costly clustering methods such as hierarchical agglomerative clustering on large data sets. By gradually expanding the coverage of samples we can turn our analysis from an overall discovery of the general role structure in the data set to a complete assignment of all users to roles. Yet the most important benefit is enabling hard, hierarchical clustering and classifications methods to produce probabilistic assignment, capturing the uncertainties of role allocation for users on the fringes between groups.

Therefore, instead of clustering and classifying a data set once (which can be very costly and does not capture uncertainties well), we create representative samples with controllable overlap with our *Multi Sampling* strategy. These samples are clustered hierarchically, creating candidates for user roles that can be explained from the features in the clustering tree. With this *Multi Sampling* strategy we are able to enrich the hierarchical agglomerative clustering with aspects of probabilities, while commonly available method only allow for hard assignment.

The cluster analysis is followed by the *classification*, which delivers for each sample clusters of users with probabilities to given user roles from literature.

The competing labels from the different clusters and classifiers are *combined* to produce a *probabilistic role assignment*, so that we are able to clearly recognize the core users of clusters (same role assignment) as well as users which lay in between different clusters and thus user roles (different role assignment). The fact that some users do not get covered by the Multi Sampling strategy or occur only once is a tuneable, which is explained more in detail as part of our analysis in Section 6.

Since we have addressed different use cases in our questions, we have to distinguish between complementary scenarios, requiring different quantities of human involvement given the amount reference data: completely/partly unexplored data sets without or little training data vs well-established training data. This distinction is emphasized in the program flow chart in Fig. 2 that serves as a guidance through the following sections. While steps, such as the preprocessing of the raw data set, which includes normalization and standardization techniques, as well as the sampling and clustering of the data remain identical for both scenarios, the differences are as follows.

1) If only data sets such as a new social network or not yet comprehensive training data are available, we discover groups of similar users and their hierarchical relationship by clustering, thus providing candidates for user roles. The analyst will then assigning role labels to these groups to build manually new training data or enrich already available training data. He/she is aided by quality metrics, visualizations and dimensionality reduction like Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) to inspect the assigned labels. In turn, these manually provided labels form the input for a classifier that captures this knowledge and can be cross-validated on this data set.

2) If a sufficiently complete training data from the same social network with the same features is available for a classifier, this -possibly very tedious- labeling process can be cut short by providing candidate labels for the clusters in a new data set. Our training set (and additional manual labels) may be cross-validated to ensure the quality of the model.

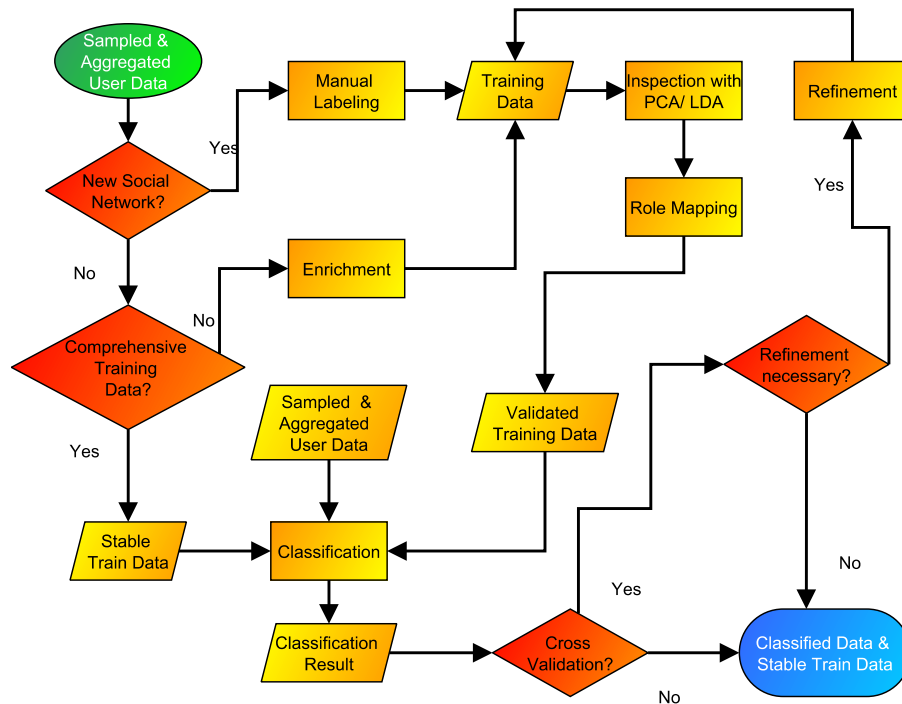


Fig. 2. Flowchart of the classification considering the scenarios

The user can evaluate these candidates either within the new data set or compare the roles across the data sets, as we show in our analysis. We also explored causes of mislabelings and provide methods to adapt them, yet a full exploration of options remains future work.

For each scenario we can go on with the combination strategy of the clustered and classified users and analyze the results considering different tunables in the steps of the multi sampling and combination strategy as well as the classification step which will be presented in detail in Section 6.

4. Feature Selection and Data Clustering

After introducing the main aspects and questioning of our approach, we focus now on the steps of the Feature Engineering and Data Clustering

4.1. Feature Engineering

In this work we aim to use features that cover significant and complementary aspects of users and are well established in the literature [6,13,1]. In addition, it should be feasible to compute in large scale so that data is commonly available and incur moderate cost

to compute. Likewise, we want to avoid a large number of features, as this hurts both algorithm performance and explainability.

Fig. 3 highlights the classes and instances of features: *static user properties* express (self-)description: most relevant is the *verified* status of a user, traditionally reserved for celebrities and influential users. *User activity* is characterized by the number of original tweets of each user (observed and “offtopic”), the activities on other tweets such as retweets and replies within the topic as well as mentions of other users. Basic *network position* features like the number of *followers* and *followees* of a user as well underpin the potential to exert influence. In turn, the user’s ability to actually elicit *reactions from the network* is captured by the *ratio of tweets* to lead to *replies* and *retweets* as well as the frequency of *being mentioned*.

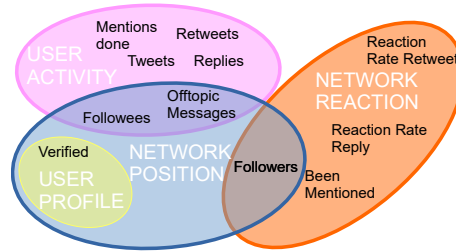


Fig. 3. User Feature Classification

We investigated a wide variety of additional features from these classes, but dropped them as they were correlated or had little discriminative power. We excluded complex network metrics such as centralities, spatio-temporal features [18] as well as content analyses [1,11]. Those suffer from data availability as well as cost and may be used for refinement or specialized sub-roles. Even partial social graphs are exceedingly hard to get from any social media (including Twitter), while our crawling strategy already provides a topic focus.

To investigate the correlation of the features described in the last paragraph, we depicted the correlation of pairwise features in a symmetric heat-map, as can be seen in Fig. 4. The bar on the right hand-side visualizes if pairwise features have a high negative correlation (deep blue), no correlation (white) or a high positive correlation (deep red). As most of the feature pairs have no correlation or only a weak positive or negative correlation, the features *followers* and *followees* are the most correlated features (as popular users show gains in either dimensions), yet changes in their ratio turned out to be a discriminative feature for specific groups, so we still considered both. Likewise, we keep some feature pairs with moderate positive correlation, e.g. *mentions done / tweets*, *offtopic messages / retweets* as well as *followers / offtopic messages*.

Given that many features in social media exhibit significant skew and value domain variation, we normalize each data set individually, so that the relative distribution differences and feature drifts are captured. More specifically, we reduced skewness using logarithmic transformation, followed by a Min-Max normalization to bring the values into

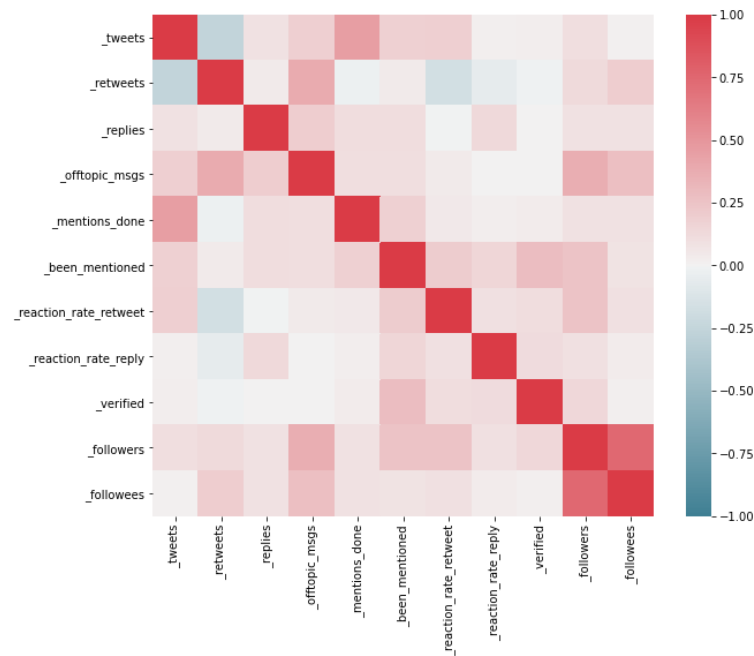


Fig. 4. Correlation Matrix for User Features

a range of 0 to 1. We also considered methods like inverse transformation, square/cube root and box-cox, but neither resulted in more balanced results.

Table 1 shows the properties of the *Olympics 2012* data set features before and after the normalization and standardization process.

As can be seen in Table 1 most of the features excluding *offtopic messages*, *reaction rate* for *retweets* and *replies* and the *verified* status contain strongly right skewed data, which can also be seen in the small median values up to the high 99th percentile and maximum. The normalization strategy is effective, leading to almost balanced skewness and median values.

4.2. User Group Clustering

To identify the structure and (sub)-groups among the user data, we evaluated a broad range of unsupervised learning approaches based on centroids (e.g., K-Means¹), density (like DBScan²) and probability distribution (e.g., EM³). Hierarchical clustering⁴ turned out to be most suitable: a) it can capture complex, irregular shapes without requiring a fixed number of clusters and b) the hierarchy serves as an (yet unlabeled) classification tree on

¹ <https://scikit-learn.org/stable/modules/clustering.html#k-means>

² <https://scikit-learn.org/stable/modules/clustering.html#dbscan>

³ <https://scikit-learn.org/stable/modules/mixture.html>

⁴ <https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html#module-scipy.cluster.hierarchy>

Table 1. Feature Statistics Olympics 2012 Twitter Data

Feature	Original Features				Normalized		
	Median	99%	Max	Skew	Median	99%	Skew
tweets	2	19.00	16621	543.97	0.11	0.31	0.72
retweets	1	13.00	3780	331.82	0.08	0.32	0.75
replies	0	3.00	759	204.19	0	0.21	2.52
offtopic	59	3790	>100K	13.28	0.35	0.71	-0.19
ment._done	0	8.00	7802	723.10	0	0.25	2.26
been_ment.	0	3.00	>140K	1017.85	0	0.12	6.73
react._retweet	0	0.75	1	2.55	0	0.75	2.55
react._reply	0	0.33	1	8.67	0	0.33	8.67
verified	0	0.00	1	22.62	0	0	22.62
followers	172	9520	>15M	229.46	0.31	0.55	-0.07
followees	231	2908	>600K	74.67	0.41	0.6	-0.97

which feature differences explain the user roles. Since we have a multi-dimensional data set geometric linkage methods, in particular Ward's worked best.

Yet, hierarchical methods are not without issues: On a technical side, they tend to incur high CPU and memory costs even for moderately large data sets due to their $O(n^2)$ scaling. On a conceptual side, almost all popular approaches tend to only support "hard" clustering, assigning a data point exclusively to a group. In reality, users may exhibit trait of multiple roles to various degrees, making a "soft", probabilistic assignment more meaningful.

To address both of these issues, we chose a sampling/ensemble-based approach when clustering the data: Clustering a small number of samples allows us to quickly discover the structure while drastically reducing the cost compared to clustering the whole data set. By incrementally drawing more samples, we see a linear cost increase (while allowing parallel execution) and provide a faithful representation of the data. With overlapping cluster results from several samples for the same user, we can choose to assign to a majority role or the probability for specific roles. Likewise, we can determine how stable the role recognition is. The number of samples becomes a tuneable, trading off the effort of computation (and labeling) with the coverage of users and the amount of support for the roles. If all users need to be covered, we may minimize the overlap or apply metric-based assignments.

4.3. Determining Cluster Count & Analysis

A key question when identifying user groups (and thus roles) by clustering is the actual number of such groups. While hierarchical clustering avoids the issue of having to provide a fixed number of clusters beforehand (as, e.g., K-Means or EM require), the classification tree (often represented as a dendrogram) produced by the clustering allows for a large range of cluster numbers - between 1 and the number of input values, as the cluster candidates are aggregated along the hierarchy.

Traditionally, this issue is tackled by computing quality metrics such as Davies-Bouldin, Silhouette and Calinski Harabasz (which are all internal cluster quality measures) for cluster candidates determining a useful point in this metric space, e.g., at diminishing returns using the elbow method. We followed the approach of [19] which relies on the distances

of the dendrogram as metric and refines the elbow with the acceleration of these global and local distances.

This approach yielded already useful, but not entirely satisfactory results: we could reliably determine the generalized, coarse-grained main groups, which correspond well to those in Fig. 5. For fine-grained roles we (often) did not get clear indications or (sometimes) groups that would not relate to user roles described in the literature.

We augmented this generic approach with a domain-specific methodology that is based on the insight that user roles often can be refined by clear differences on specific features, not just on general, global metrics. Intuitively, comparing boxplots (which show means, median and quantiles) in the manual labeling process led us to determine features whose differences explain the characteristics of subgroups. To formally express and discover these differences, we rely on statistical measures. In particular, we utilized effect sizes such as (pooled) Cohens d [15] to capture significant feature deviations. Cohens d is defined as the difference between the means of two sets divided through the standard deviation, while the pooled standard deviation [3] allows to deal with cluster candidates of different sizes, so smaller clusters with significant features can be detected reliably. Otherwise such smaller clusters that represent pronounced user roles such as *Star* or *Semi star* tend to get absorbed by bigger clusters. Furthermore, pooling is less sensitive to feature drift.

The refinement process is modeled using a Depth-First search covering the subtrees in the dendrogram forming the generalized roles. At a search step, the process compares (in pairwise fashion) the measures for each feature of the current cluster to those of its two direct descendants which are the refinement candidates. This search continues as long as there are significant effects, leading to a possible cutoff for refinement in this particular path. After the whole Depth-First Search we only have to cut off at the deepest distance in the dendrogram where we have found a clustering with considerable features.

The significance criterion remains a tuneable, but in most cases the results were best when finding at least 2 features with a large effect. When we investigate new data sets the criteria for overall significance may have to be adjusted, yet -on our experience- this rarely needed, as in almost all cases these values delivered useful clusterings.

Considering how clustering are used in the overall approach, no perfect fit for the cluster number is actually necessary. Instead we would like to slightly overestimate the number of clusters, avoiding an early cutoff that would lose possible user groups. The spurious groups will be merged either during the manual label assignment or by the trained classifier, as shown in the following section.

5. User Role Identification

While the hierarchical cluster structure identifies candidates, it does not provide the actual user roles. We now describe the (manual) assignment that also serves as the training data for a role classifier as well as the transfer of this user role knowledge to new data sets. This Section provides more details on the scenarios which were introduced in Section 3.2 as well as in Fig. 2.

5.1. (Manual) Role Assignment

Considering that neither a general consensus on types of user roles in social media nor precise definitions or models exist (see Section 2), we apply several complementary

methods to derive meaningful candidates. Starting from the cluster hierarchy (described in Section 4) that provides indications on the (approximate) number of clusters and their respective separation, but no meaning of those, we apply complementary approaches: 1) manually analyzing the overall structure of the clustering (dendrogram) and features of the individual cluster (boxplots) with the significant features allows us to match these clusters to fine-grained user roles from the literature [17,10], e.g., *Semi Star* or *Amplifier* (cp. Fig. 8). 2) dimensionality reduction such as PCA or LDA (cp. Fig. 7) aids this exploration process in several ways: the composition of the main components further highlights relevant features. The reduced number of dimension aids the computation cluster separation metrics and simplifies visual inspection. Likewise, it also helps with correlating user roles across data sets and exposes the drift/evolution.

These approaches yield an iterative strategy: Using the stopping heuristics, the number of clusters is narrowed to typically 15-30 candidates, though this value is clearly dependent on the specific data set. The structure of the dendrogram guides the manual mapping process in which we compare the feature distributions presented in boxplots. Certain heuristics support this work: 1) Specific classes of roles tend to manifest themselves further up the hierarchy, creating subtrees for those classes that could then be refined into more specific roles. 2) Some very distinct roles tend to show up in most data set, providing an “anchor” for the labeling. The refinement process is stopped once we do not gain additional, well-discernible or well-interpretable clusters. In some cases it may be useful to coarsen the roles again or combine several clusters into a single role.

We match these aspects to the role descriptions provided when possible (in particular on well-studied roles like *Star* with its large number of followers, almost always verified status and generally high impact despite relatively low activity), but also observed stable, recurring clusters that did not align well with the known roles descriptions, leading to role discovery. In our data set, we also found more *action triggering* user roles (cp. Fig. 5) such as *Idea Starters*, which are similar to *Semi Stars*, but gain popularity in the network by creating more content and triggering higher reactions in the network. Furthermore there are *Amplifiers*, which are well networked users pushing and spreading mostly (existing) trends and *Rising Stars*, which gain a large number of followers by activity in the network, receiving significant reactions in terms of retweets, but not yet at the level of *Stars* or *Semi-stars*, which fit into the intermediate user role group.

More intermediate user roles are *Spammers*, which have mostly a high activity in the network but are not as popular as the action triggering users. They are similar to *Average Users*, which can hardly be distinguished from the whole data set focusing on deviations of the statistical indicators of the features and stand in most cases one of the biggest groups in the user roles. *Daily Chatters* distinguish from the spammers because of their more moderate action in the network. In most cases they lay in between the *Spammers* and *Average Users*. Furthermore there is a role of the *Commentator*, which is similar to the *Daily Chatter*, but is more active in creating content, retweeting and especially in cases of reactions in the network by replying to content.

The last group of similar users we recognized in the data sets are passive users like *Forwarders*, who are better networked like average users, but mostly only forward content and thus receive only less reactions in the network. *Listener*, who mostly only consume, and thus have a weak connection in the network, share only less content and do not trigger other users. They are only underbid by *Loner*, who are mostly inactive in the network.

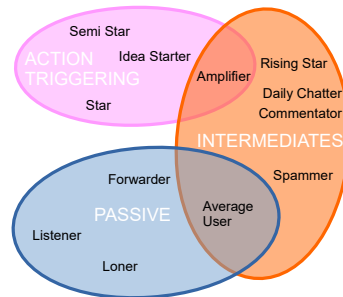


Fig. 5. User Roles

The same process can be applied across data sets, comparing the dendrograms, cluster metrics/descriptions and correlating the labeled roles. By doing so, we can track the user roles across the data sets and evaluate concept shifts and drifts among them, such as their frequency/probability of roles or their feature distributions. Typically, we observed around 10-15 class candidates that did show up in varying frequency over our data sets, sometime disappearing entirely.

While this manual labeling excels at capturing the specific knowledge of a domain expert and produces high-quality and well-described role clusters, it is a tedious process with very limited scalability that may suffer from reproducibility issues due to the subjective nature of human assessment.

5.2. Classification

The goal of classification is to transfer knowledge on user roles from existing data sets or other samples of the same data sets that have already be labeled. The multi-sampling approach as well as the previous clustering stage lead to some particularities that we describe in more detail.

The classifier consumes two types of input: On the one hand the training data, described in the previous subsection, is essential to capture the user role models. On the other hand the clustered user data, which needs to be classified so that each cluster is assigned to a user role label. As mentioned before in 3.2 we designed a Multi Sampling and Combination Strategy to provide scalability and role probabilities for each user a stable user roles based on the identification of significant features. In Fig. 6 the important parts of our Multi Sampling and Combination Strategy can be seen.

We start with user data that has been aggregated and engineered into representative features. The full set is split up in several representative, possibly overlapping samples that are clustered and form the second input for the classification. When applying manual labeling (as outlined in the section before), each user receives a role label depending on the cluster it belongs due. For those clusters that are not manually labeled, the classification process provides for each cluster, and thus for each user in the cluster, a probability vector to the given user roles. After each user in each sample has been clustered and classified, we can combine all probability vectors for each user into a single probability vector. While

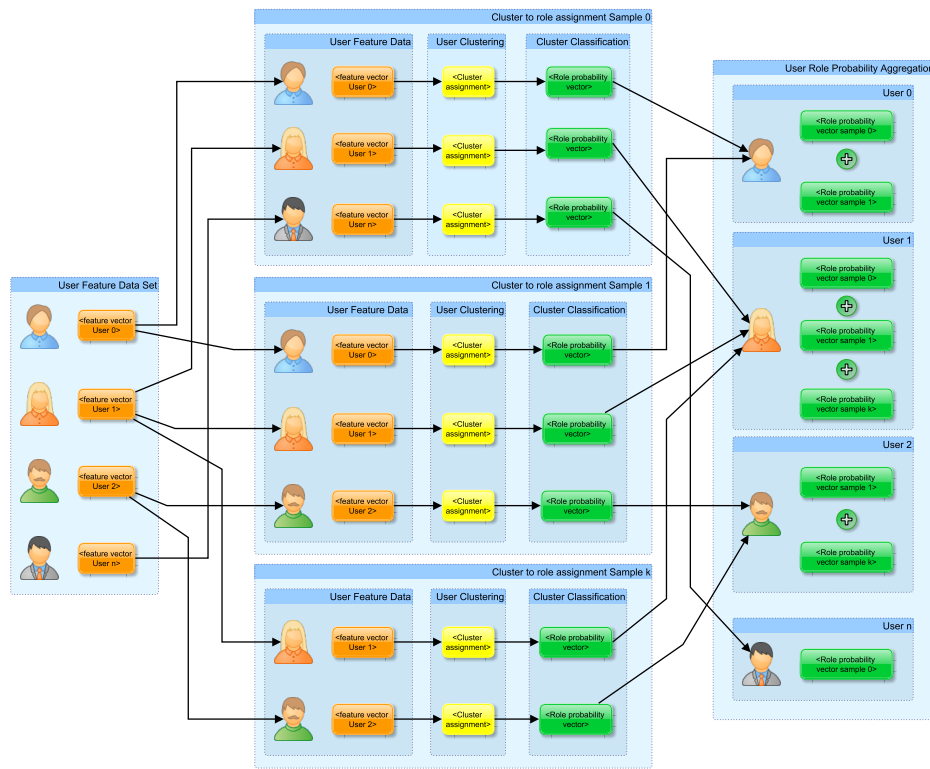


Fig. 6. Combination Strategy using classifier with training data available

it is feasible to only retain the user role with the highest probability at either the individual classification results or after the combination, this leads to suboptimal results, as the uncertainties for users that are at the fringe of two or more clusters and thus did not get a clear majority for one user role. Beyond that, the variances samples may lead to different clusters (and thus relative user behavior), further strengthening the need for a probabilistic classification to ensure accuracy for borderline users.

To overcome the issues mentioned at the end of Section 5.1, we utilized a classifier that was trained from several samples of one data set using the cluster means and determined the role labels on clusters in other data sets, expressing a n-class problem. For training, we took samples that showed the best cluster separation (supported by PCA, which can be seen in Fig. 7) to minimize the noise in the model and concatenated them. As initial experiments showed, the original number of dimensions in the data yielded better quality than reduced dimensionality.

The creation of training data was a time consuming iterative process consisting of a manual cluster analysis followed by a manual classification of each cluster to the user roles from literature. We picked several cluster centroids from several samples for each user role and adjusted the training data incrementally after using PCA and LDA. If we have a closer look at the training data in Fig. 7 the reduced dimensions redeem the complexity of

our given multidimensional cluster means. For each user role we have clearly separated training data as the dimension reduced projection in Fig. 7 shows for almost all roles. Since we have some user roles which lie close in between two user roles, e.g., *AVG User* vs. *Forwarder* vs. *Daily Chatter*, the process of creating training data is a very important key element which demands a manually consequent and precise procedure.

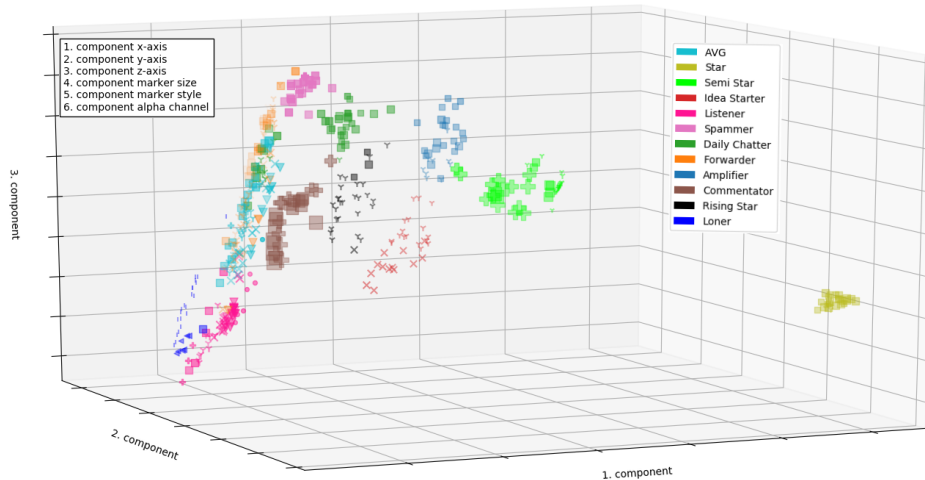


Fig. 7. Principal component analysis of clustered samples from *Olympics 2012* data set

We also investigated the tradeoff between classifying individual users instead of entire clusters. While the former may yield higher classification performance (no clustering is needed when working with a new data set), we observed that the inherent “noise” shown by individual users does not lend itself well to either training or classification. We therefore decided to represent a cluster by feature values computed from all its members. For this step, we considered the arithmetic mean as well as the median, each also with pooled Cohens d (relative to the entire sample) to boost separation. Our evaluation showed that means tended to provide better separation than median, while pooled Cohens d seemed to capture more temporal evolution than “pure” means.

As our (clustered) data sets are relatively small and skewed, yet we seek to express a large number of classes, we see little support for some classes. This more or less rules out deep learning. Instead, methods based on ensembles of decision trees, e.g., Gradient Boosted Decision Trees (GBM) or Extremely Randomized Trees (ET), multi-class support-vector machines (SVM) or k -nearest-neighbor (kNN) turned out to be most suitable. We utilized the Python implementations of scikit-learn for ET, SVM and kNN as well as XGBoost⁵ for GBM.

The setup to build training sets utilized repeated stratified cross validation with three splits (leave-one out, due to the small amount of data) and three repetitions (with different permutations to cater for possibly missing groups). We used F1-macro as a metric to

⁵ <https://xgboost.readthedocs.io/en/latest/>

compensate for class imbalance and prevent focus on either precision or recall and applied grid search to tune parameters. All classifiers learn and generalize well, leading to 94-95 percent score in validation and training set with no obviously stronger or weaker candidates.

When transferring the classification to new data sets, we compensated for mislabelings by varying training and prediction data (e.g. cluster number) or choosing more suitable training sets. Explicitly including drift models and relevance feedback from the user remain future work.

6. Evaluation

After explaining the idea, concepts and set-up of our approach in Section 3 we now present and discuss results of experiments on diverse data sets from Twitter. In our analysis we address the three questions outlined in Section 3.2. For each step we do not only show the technical results but also report our empirical observations.

6.1. Data Sets and Preparation

While our long-term goal is to recognize user roles over a variety of data from various social media, we focused in this initial analysis on data sets that are well-defined and contain a large number of users. As in most of the related work, we relied solely on Twitter, as its one for the few social media services in which data sets containing large numbers or longer periods of time are available.

In order to transfer knowledge on user role detection, we are looking at several classes (Table 2): major sports tend to be repetitive and predictable with a very large number of messages and users, covering significant periods of time. Different types of sports provide (albeit limited) thematic variance. These data sets are complemented by those of two major disasters which also tend to have a strong, yet very different topic focus and different interaction patterns. Finally, we applied our work on an instance of the Twitter sample stream to assess a data set without a strong topic focus.

Table 2. Overview on Data Sets

Data Set	Messages	Users	Time Period	Category
Olympic Games 2012	13.68M	2.27M	August 2012	sport event
Olympic Games 2014	14.58M	1.96M	February 2014	sport event
Olympic Games 2016	38.05M	4.76M	July/August 2016	sport event
FIFA World Cup 2014	109.00M	10.40M	June/July 2014	sport event
2015 Paris Attacks	6.77M	0.74M	November 2015	tragic incidence
NFL Superbowl LIV 2020	8.89M	0.89M	2. March 2020	sport event
2016 Berlin Truck Attack	0.66M	0.15M	19. December 2016	tragic incidence

Our data sets had each been recorded using the Twitter Streams API and Search API using commonly proposed hashtags. We only considered users that were active at least twice, as several metrics require aggregations. Generally speaking, the relative feature

distributions after normalization varied only slightly over time from 2012 until today, with minor changes: users tend to move slightly more into “reactive” behavior of forwarding than content generating or mentioning, while the *verified* status is now much more prevalent. Overall activity increased moderately, forwarding actions became more widespread.

6.2. Initial Data Set: 2012 Olympics

The first step focuses on a single data set (*Olympic Games 2012*) with uniform feature usage and role stability due to the relatively short period of time. Given those benign conditions, these analyses provide insight to which extent such as clustering, user roles detection and automated labeling are feasible, as stated in Q1 in Section 3.1.

Following the approach outlined in section 3.2, we created samples covering 5% to 10% of the data set and applied the hierarchical clustering we introduced in Section 4.2 afterwards using `scipy.cluster`. The latter sample size represents the maximum that could be clustered on the machines available on an 8-core partition of an AMD Epyc 7401. A small data set like *Berlin 2016* may still be clustered completely, yet a sample can be generated almost instantly, as can be seen in Table 3. For large data sets, full clustering is clearly impossible, while samples fit well. The cost is almost entirely consumed by creating the linkage matrix, so refinement/exploration steps are interactive in all variants. After clustering, we manually labeled clusterings of the samples to get a ground truth as training and test data as mentioned in Section 5.1. In real-life settings, this labeling and testing may be performed incrementally until a sufficiently good understanding of the data has been established.

Table 3. Runtime and memory of samples, full data sets and approximated(*)

	Oly12 5%	Oly12 10%	Oly12 100%	Berlin16 10%	Berlin16 100%
runtime	19 min	136 min	226 h*	10s	38 min
memory	94 GB	375 GB	375 TB*	1.2GB	184 GB

In particular after applying PCA (see Fig. 7), we can identify a number of well-separated clusters. Despite showing some minor variances, the dendrograms (see Fig. 8) over the set of samples exhibit a very similar overall structure that has become a part of our overall classification as on the leftmost column of Table 4: there are between 3 and 5 subtrees representing major groups, expressed by very distinctive feature values: The first major group (green) shows users that are able to *trigger strong reactions* (*retweets, replies, being mentioned*), the second (red) shows *passive users* with fairly weak positions in the network, while the group(s) in between show various degree of *moderate activity and impact*. Even further down the tree, (as shown on the boxplots), we see a strong motivation for fine-grained roles. While the cluster sizes are often small, there are salient feature differences (which we can detect using statistical tests like Cohens d) that explain the existence and semantics of this group. In the example one can see how *Semi Stars* and *Amplifiers* split on (among others) *retweet* activities and *reactions*. Overall, we determined 12 roles in the Olympics 2012 data set that are described in Table 4. Some characteristics are shown in the second column, in particular stronger deviations from the average as well

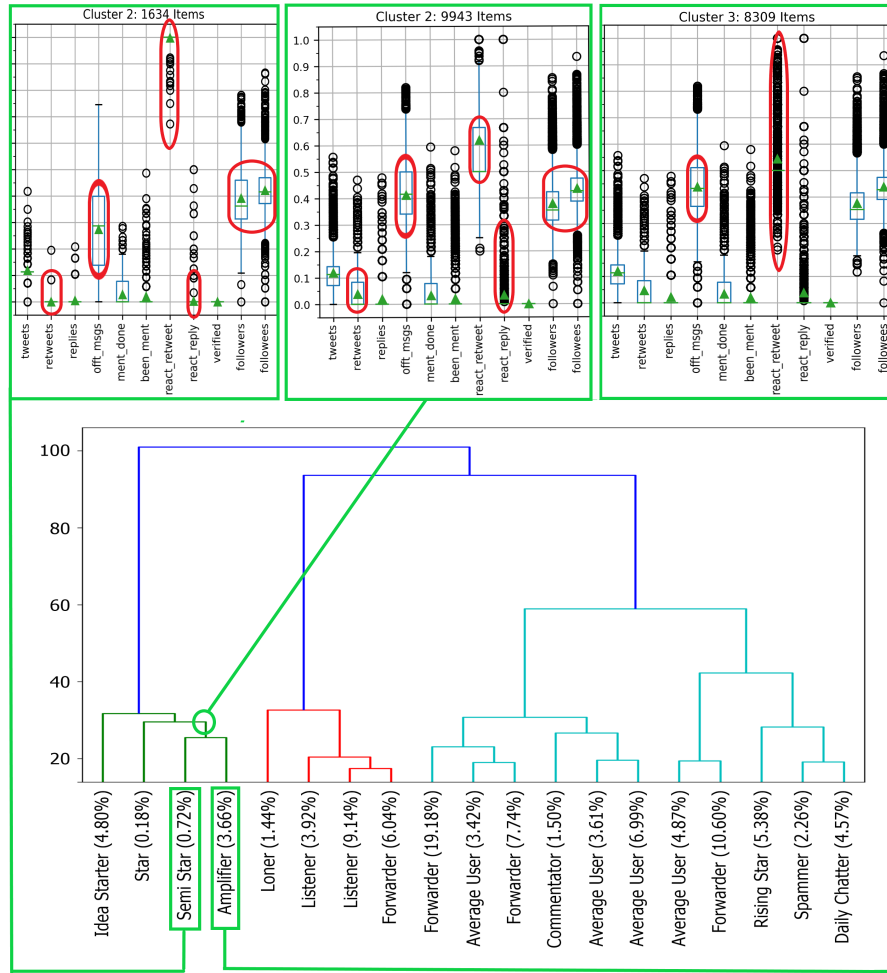


Fig. 8. Olympics 2012 10% sample Dendrogram with salient features

as (broadly) similar user groups. Note that these user roles are not strictly defined for all data sets, but a good starting point for the evaluation of all further data sets.

Continuing in our pipeline from Fig. 1 with the combination of the clustered and classified data sets and thus the users (see also Fig. 6), we now take a closer look at the coverage and certainty of roles, which can be seen in the subfigures of Fig. 9 for combinations of two different samples sizes (5% vs. 10%) for the *Olympic 2012* data set and the *Superbowl 2020* data set (10% vs. 20%).

When increasing the number of samples and thus the number of individually classified users the number of role assignments per user improves. As a result, the number of users without any role assignment (red bar) drops quickly, while the number of users with multiple, mostly consistent role assignment (green bar) grows rapidly. Furthermore, the number of users that only see a single role assignment (orange bar) becomes smaller,

Table 4. User roles and their characterization: \approx shows closeness to other roles, \downarrow/\uparrow feature deviation from close role/whole data set, $\searrow / \leftrightarrow / \nearrow$ changes over time

	Role	Characteristics	Freq./Trend
action triggering	Star	followers > followees, verified, \downarrow activity, \uparrow mentioned	0.2–0.8 \nearrow
	Semi Star	\approx Stars, \downarrow followers, mentioned, \uparrow react. (re)tweet, retweets, replies	0.2–1.4 \searrow
	Idea Starter	\approx Semi Star, \downarrow followers, \uparrow reactions	1–4 \leftrightarrow
	Amplifier	\approx Idea Starters, Semi Stars, \uparrow followers, followees	0.5–5 \searrow
intermediates	Rising Star	\approx Semi Star, Idea Starter, Amplifier \uparrow followers, (re)tweets, replies	1.5–5.5 \searrow
	Daily Chatter	\approx Average User, Spammer, \downarrow (re)tweets, offtopic	5–15 \leftrightarrow
	Commentator	\uparrow replies, offtopic, reations	0.3–2 \searrow
	Spammer	\uparrow (re)tweets, replies, offtopic \downarrow followers, followees, reactions	1–7 \leftrightarrow
passive	Average User	offtopic > tweets, retweets	8–30 \downarrow
	Forwarder	retweets > tweets, \uparrow offtopic, followers, followees. \downarrow reactions	25–65 \uparrow
	Listener	\downarrow (re)tweets, reactions	6–20 \nearrow
	Loner	$\downarrow\downarrow$ tweets, offtopic, followers	0–1.5 \searrow

enabling us to perform actual probabilistic assessments on the assignment certainty. In turn, the increasing “relative majority” part (yellow bar) gives insights on user that are not well identified - which is data set-dependent, but often includes *Spammer*, *Loners*, etc.. For most of these roles the percentage for the strongest role is between 40 and 50%, which is also a very persuasive value in the context of a 12-class classification problem. Also the distance to the second-best user role is quite high, which also substantiates the significance of user roles in our sampling and combination strategy. Further increasing the number of samples does not significantly decrease the share of those users, indicating that these are not artifacts of the sampling approach. Overall, the scaling works well, thus validating our approach.

When utilizing bigger, yet fewer samples (Fig. 9b and 9d) compared to the combination of smaller samples (Fig. 9a and Fig. 9c) for the same number of users, the quality of the results tends to be slightly better (in particular for the *Superbowl 2020* data set), yet at much higher resource requirements due to the quadratic complexity for clustering. As a consequence, it is typically better to utilize smaller, but more numerous samples.

We evaluated the clustered and labeled samples (in total 507 clusters) with the classifiers mentioned in Section 5 and achieved nearly perfect results with the classifiers, as the leftmost data points in Fig. 11b show. The confusion matrix between roles (Fig. 12a - 12d) confirms these results, as there are only very few mislabelings resp. misclassifications between *Average User*, *Daily Chatter* and *Listener*, respectively - which are also close to each other in feature space. The strong variance in the feature distribution present in the boxplots (Fig. 8) also shows why training and classifying individual users instead of clusters yields inferior results. Since we use a sampling and combination strategy, the

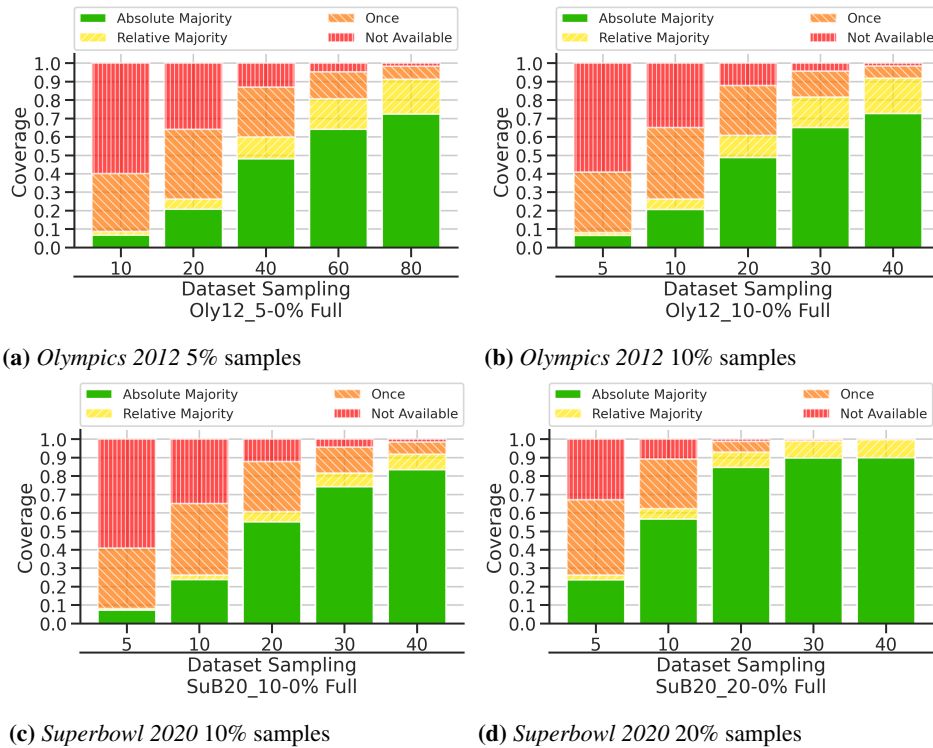


Fig. 9. Sampling and combination for several datasets

effect of mislabeling or misclassification is dampened by the fact that the first role is -in most cases- very dominant or -in cases of no majority- has a significant distance to the second-best role.

Overall, the results show that both clustering and classification work well. Expert knowledge is needed to interpret the dendrogram and assign roles, but already within a single data set, the knowledge can be transferred to additional samples and their clusters.

6.3. Multiple Individual Datasets

The second step analyzes several data sets individually to understand if the approach is more widely applicable (thus providing insights on Q2). Furthermore, this will show us of the same or similar roles are present in data sets varying in time and topic and how they evolve over time.

The 12 user roles identified on the *Olympics 2012* data set are also present and well-separated in the other data sets, though -as the rightmost column of Table 4 shows- the frequency (in percent) varies over data sets (and over time):

In the *Olympics 2014* (278 clusters) and *FIFA World Cup 2014* (193 clusters) data sets very few changes can be observed: *Average User* and especially *Loner* occur less frequently, while *Forwarder* and *Listener* occur more frequently.

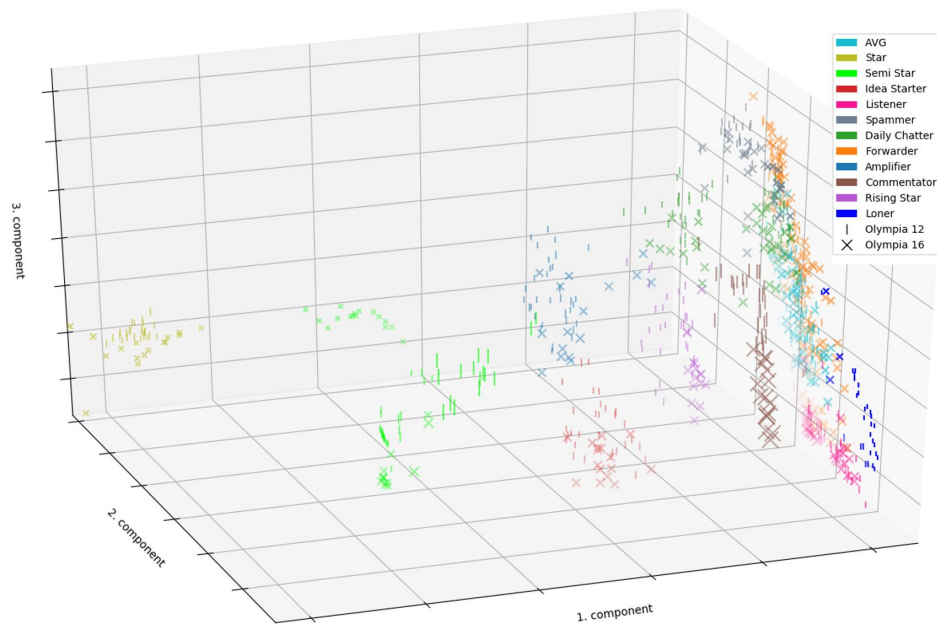


Fig. 10. PCA of clustered samples from *Olympics 2012* vs. *2016*

The first significant changes in terms of user roles frequency and features occurred in the *Olympics 2016* data set (355 clusters). The PCA in Fig. 10 -showing cluster centroids of our training data- provides a salient concept drift for many user roles between the *Olympic 2012* data set (Pipe symbol |) and the *Olympic 2016* data set (Crosses X): in particular, *Semi Stars* tend to also cover a space much closer to *Stars*, as the “verified” status was more freely distributed by Twitter. The already observed trend on the frequency of *Average Users* / *Loner* and *Forwarders* strengthens, as many users prefer to retweet more than creating their own content. This trend continues for the *Superbowl 2020* (345 clusters) data set, which is otherwise (despite the different sports and the time difference) rather similar to *Olympics 2016*.

The *2015 Paris Attacks* (160 clusters) data set covers a very different topic and distinct interactions (fewer *offtopic messages*, more *retweets*). Some user roles are not present (*Commentator*, *Loner*), yet most of the overall trends match the picture of the “sports events”: forwarding instead of content creation becomes more dominant (both as feature and as role), corresponding to the wider trend in all social media. In turn, “influencer” roles become pronounced, to the point where the *Semi Star* may have to split into two separate sub-roles.

The only exception where we could not apply our methodology was the random *Sample Stream*, as features based on topics lose their usefulness.

For Q2, we could show that the same features can be applied, leading to consistently recognizable user roles. We could observe how the distributions of roles shift over time and also correlate the roles of users over the boundaries of the data sets. Yet, at this step, the effort of labeling samples of each data set manually is a limiting factor.

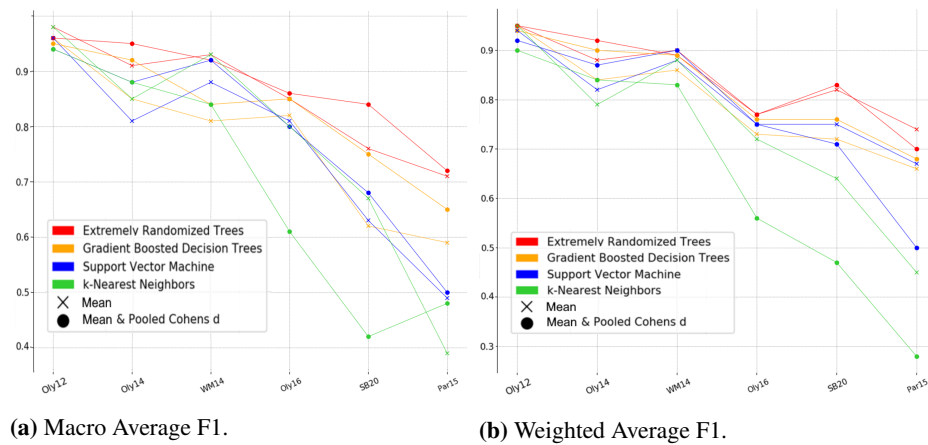


Fig. 11. Information retrieval measure F1 for classifiers

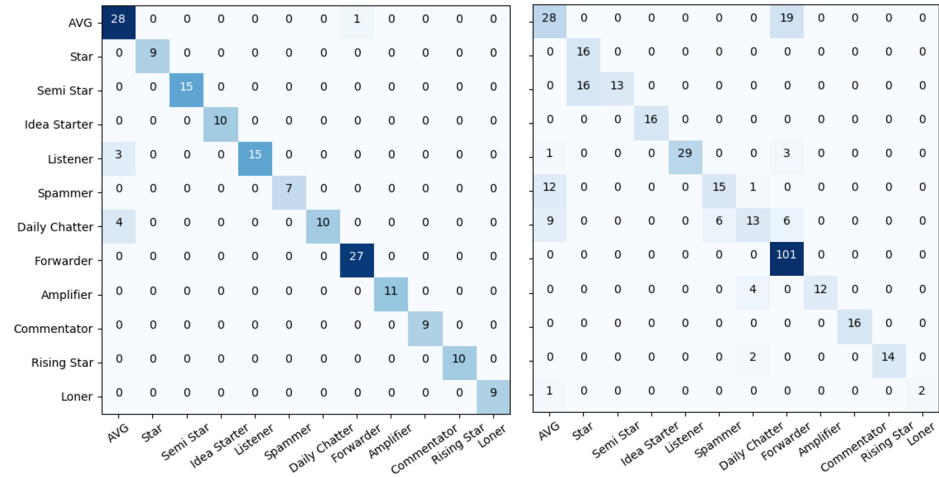
6.4. Applying Models on New Data Sets

In the third step, we classify new data sets with the models gathered from reference data to transfer role knowledge and assess the quality and effort involved, answering Q3. We further study the impact of variation and drift to understand the limitations.

Fig. 11a and 11b show the F1 scores when classifying the data set based on the *Olympics 2012* as the reference, as it provides the longest prediction period. While the weighted values in Fig. 11b depicts the quality of frequently represented user roles, the macro values in Fig. 11a support the overall performance of the classifiers. Overall, one can see a gradual degradation over time on the sport events, as the classification methods do not explicitly capture the drifts observed in the previous section, but still generalize the roles over time. Yet, the best methods achieve a 0.85 F1 score for “late” sport events. The *2015 Paris Attacks* data set sees the largest degradation, showing topic and interaction differences have a more profound impact than time. When comparing all these results to the slightly worse “macro” values, one can see that small groups are captured well, while larger clusters tend to be somewhat “blurry”.

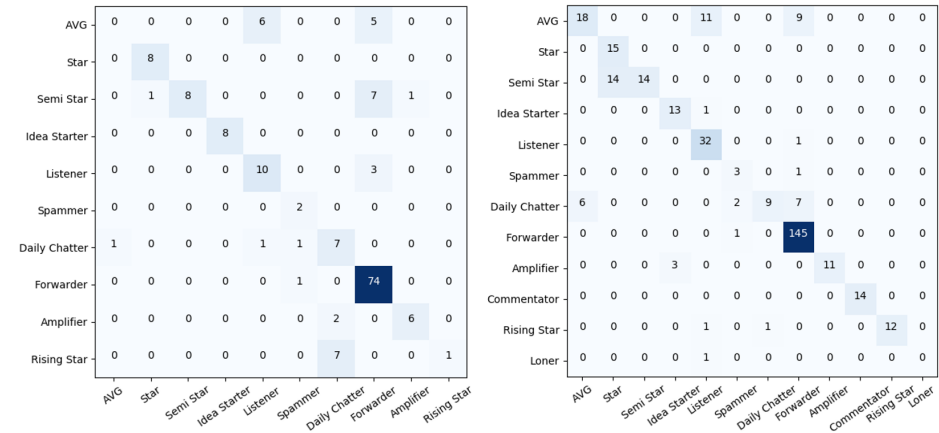
kNN and SVC keep up well for short time intervals, but tend to lose ground on longer distances. ET holds a small edge over GBM, while the latter stays still competitive and incurs much lower runtime cost. Both benefit from enriching the data sets with the pooled Cohens d values.

The confusion matrices for the *Olympics 2012*(Fig. 12a) *Olympics 2016*(Fig. 12b) and *Superbowl 2020* (Fig. 12d) data sets show how that roles that were either not well separated in the *Olympics 2012* data or drifted significantly are most affected. Yet, these misclassification often leads to adjacent roles, e.g., *Average Users* as *Listener* and *Forwarder*, *Daily Chatter* as *Forwarder* and *Average User* or *Semi Stars* as *Stars*. Especially the scores for Superbowl 2020 emphasize the drift to forwarding content as well as the rise of influencers from *Semi Stars* to *Stars*. Only in the *Paris 2015* data set (Fig. 12a) some more misclassifications are noticeable, due to the topical different training data. Thus the



(a) Olympics 2012

(b) Olympics 2016



(c) Paris 2015

(d) Superbowl 2020

Fig. 12. Confusion matrices of classifications

F1-scores actually understate the quality of the result, as they do not take the adjacency of roles into account.

We added the data set of the *2016 Berlin Truck Attack* (Christmas market) that was not evaluated in the previous stages and provides topic similarity to *2015 Paris Attacks*, while being close to the *Olympics 2016* in time. This data set provides a good opportunity to assess the impact of different training sets: in addition to baseline of the *Olympics 2012* and close sets (*Olympics 2016*, *2015 Paris Attacks*) and *Superbowl 2020* as a small, recent data set, we tested two combinations: *Olympics 2012* and *Superbowl 2020* for the full time range and *2015 Paris Attacks* with those two as mix of time range and topic proximity. As Table 5 shows, these combined data sets provide the best results, matching manual classification or producing misclassifications to close roles. *2015 Paris Attacks* by itself seems to be too small to provide a sufficiently general model, but is able to boost the full time range model.

The experiments show that a transfer of labeling knowledge is effective with certain limitations: large topic differences or very long time differentials diminish the usefulness, yet a good choice of reference data can mitigate this effect.

Table 5. Classification of several data sets

Classifier	Oly12	Oly16	Par15	SB 20	Oly12 + SB20	Oly12 + SB20 + Par15
XGB	0.58	0.59	0.51	0.70	0.78	0.92
ET	0.74	0.63	0.56	0.73	0.77	0.82

7. Conclusion & Future Work

In this paper we proposed a method on how to determine and label user roles in large-scale social media data sets. This method combines unsupervised learning (more specifically, hierarchical clustering) to discover the classes of users over a wide range of features and supervised learning - generalizing the knowledge from manually labeled smaller data sets.

Our analysis on a range of large data sets from Twitter show that well-separated roles can consistently be recognized and transferred. The labeling achieves high accuracy not only within the same data set, but also on new data sets from different event types and/or years apart. The resource requirements of such analyses are modest, bringing them in range of commodity hardware.

For future work, we see a number of interesting directions: As the quality of classification begins to deteriorate over longer time frames, we plan to incorporate evolution into both clustering and classification, considering both temporal models (for long-term studies of snapshots) and stream clustering (for short-term, continuous analyses). They may also pave the way for longitudinal studies of users groups and user mobility among groups. Likewise, adapting our model to cope with topically non-related or even topically unconstrained data sets poses a new set of challenges. Initial experiments show that the method should generally work, but significant work still needs to be done. In either case,

testing our method on a wider range of data sets from Twitter or even other social networks would be highly interesting.

References

1. Chatzakou, D., Kourtellis, N., Blackburn, J., Cristofaro, E.D., Stringhini, G., Vakali, A.: Mean Birds: Detecting Aggression and Bullying on Twitter. *WebSci* (2017)
2. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Who is Tweeting on Twitter: Human, Bot, or Cyborg? In: *ACSAC* (2010) (2010)
3. Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences*. Erlbaum, Hillsdale, NJ [u.a.], 2. ed. edn. (1988), literaturverz. S. 553 - 558
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1), 1–38 (1977), <http://www.jstor.org/stable/2984875>
5. Du, F., Liu, Y., Liu, X., Sun, J., Jiang, Y.: User Role Analysis in Online Social Networks Based on Dirichlet Process Mixture Models. In: *2016 International Conference on Advanced Cloud and Big Data (CBD)*. pp. 172–177 (2016)
6. Edgar, R., Alexandre, P.F., Caladoa, P., Sofia-Pinto, H.: User Profiling on Twitter. *Semantic Web Journal* (2011)
7. Espinosa, M., Centeno, R., Rodrigo, A.: Analyzing User Profiles for Detection of Fake News Spreaders on Twitter - Notebook for PAN at CLEF 2020 (09 2020)
8. Fu, C.: Tracking User-role Evolution via Topic Modeling in Community Question Answering. *Information Processing & Management* 56(6), 102075 (2019)
9. Hacker, J., Riemer, K.: Identification of User Roles in Enterprise Social Networks: Method Development and Application. *Business & Information Systems Engineering* 63 (08 2021)
10. Java, A., Song, X., Finin, T., Tseng, B.: Why We Twitter: Understanding Microblogging Usage and Communities. In: *WebKDD/SNA-KDD* (2007) (2007)
11. Kao, H.T., Yan, S., Huang, D., Bartley, N., Hosseinmardi, H., Ferrara, E.: Understanding Cyberbullying on Instagram and Ask.Fm via Social Role Detection. In: *WWW '19 Companion* (2019)
12. Lasek, P., Gryz, J.: Density-based Clustering with Constraints. *Computer Science and Information Systems* 16, 7–7 (01 2019)
13. Lazaridou, E., Ntalla, A., Novak, J.: Behavioural Role Analysis for Multi-faceted Communication Campaigns in Twitter. In: *WebSci* (2016) (2016)
14. Li, H., et al.: Bimodal Distribution and Co-Bursting in Review Spam Detection. In: *WWW* (2017) (2017)
15. Sawilowsky, S.: New Effect Size Rules of Thumb. *Journal of Modern Applied Statistical Methods* 8, 597–599 (11 2009)
16. Shu, K., Zhou, X., Wang, S., Zafarani, R., Liu, H.: The Role of User Profiles for Fake News Detection. In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. p. 436–439. *ASONAM '19*, Association for Computing Machinery, New York, NY, USA (2019)
17. Tinati, R., Carr, L., Hall, W., Bentwood, J.: Identifying Communicator Roles in Twitter. *WWW '12, rel MSND Workshop* (2012)
18. Varol, O., Ferrara, E., Ogan, C.L., Menczer, F., Flammini, A.: Evolution of Online User Behavior during a Social Upheaval. *WebSci* (2014)
19. Zambelli, A.: A Data-Driven Approach to Estimating the Number of Clusters in Hierarchical Clustering. *F1000Research* 5 (08 2016)

Johannes Kastner is a researcher and PhD Student at the University of Augsburg (Germany). He received his Masters Degree in Computer Science from the University of Augsburg (Germany) in 2016. His research interests include Clustering, Classification and User Role Detection in general.

Peter M. Fischer holds the chair for Databases and Information Systems at the University of Augsburg (Germany) as a Full Professor since 2017. He received his Diploma/Masters Degree in Computer Science from the Technical University (TU) Munich (Germany) in 2002. He worked as a researcher at TU Munich, University of Heidelberg (Germany) and ETH Zürich (Switzerland), where he received his PhD in 2006. After his work as a Postdoc and Senior Researcher at ETH Zürich he became an Assistant Professor for Web Science at the University of Freiburg (Germany). His research interests include the analysis of social streams and graphs, scalable database systems for temporal data, adaption and recommendation of information as well as provenance and assurance of data and operations.

Received: January 10, 2022; Accepted: August 05, 2022.

