Com SIS

Computer Science and Information Systems

Volume 20, Number 1, January 2023

Contents

Papers

Special Section: Engineering of Computer Based Systems

Special Section: Management of Digital EcoSystems

Special Section: Parallel and Distributed Computing and Applications

ISSN: 2406-1018 (Online)

Computer Science and Information Systems

Vol 20, No 1, January 2023

Com SIS

Computer Science and Information Systems

Published by ComSIS Consortium

Volume 20, Number 1
January 2023

ComSIS is an international journal published by the ComSIS Consortium

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 1, 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.mpn.gov.rs/

Computer Science and Information Systems

AIMS AND SCOPE

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

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

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

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2021 two-year impact factor 1.170,
- 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

Computer Science and Information Systems

Volume 20, Number 1, January 2023

CONTENTS

Papers

Special Section: Engineering of Computer Based Systems

Special Section: Management of Digital EcoSystems

Special Section: Parallel and Distributed Computing and Applications

Computer Science and Information Systems 20(1): i–iii https://doi.org/10.2298/CSIS230100iI

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

Starting 2023, this first issue of Volume 20 of Computer Science and Information Systems features 13 regular articles and three special sections: "Engineering of Computer Based Systems" (5 articles), "Management of Digital EcoSystems" (4 articles) and "Parallel and Distributed Computing and Applications" (3 articles). As is already customary, we are thankful for the hard work and enthusiasm of our authors, reviewers, and guest editors, without whom the current issue and the publication of the journal itself would not be possible.

This issue marks a milestone in the publication of our journal – due to general lack of demand we are switching to publishing in electronic form only. Not without sadness we express special gratitude to our long-time partner, printing house Sigra Star, for being with us from the start and providing high-quality and timely printing services for the better part of the last 20 years.

The first regular article, "Homomorphic Encryption Based Privacy-Aware Intelligent Forwarding Mechanism for NDN-VANET" by Xian Guo et al. tackles security and privacy issues faced by machine-learning solutions for intelligent forwarding strategies in vehicular ad-hoc networks (VANET). The article proposed PABRFD, a privacy-aware extension of the BRFD smart receiver forwarding decision solution for named data VANETs (NDN-VANET). PABRFD achieves this by using homomorphic encryption (HE) and a secure Bayesian classifier to resolve the security and privacy issues of information exchanged among vehicle nodes.

In the second regular article, "Cloud-Based ERP Construction Process Framework in the Customer's Perspective," Seung-Hee Kim et al. provide a theoretical foundation for standardized research on cloud enterprise resource planing (ERP) construction methods, as well as a practical guideline. The article provides a detailed overview, comparison of cloud and on-premise ERP, and classification of process frameworks for implementing cloud ERP into infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), content-as-a-service (CaaS), and software-as-a-service (SaaS).

Qasim Khalid et al., in "SBEO: Smart Building Evacuation Ontology" propose a reusable ontology for indoor spaces, based on three different data models: user, building, and context. The ontology provides a common representation of indoor routing and navigation, describes users' characteristics and preferences, grouping of individuals and their role in a specific context, hazards, and emergency evacuation. Evaluation demonstrates that SBEO is flexible, comparable to other ontologies in the field, and that it successfully addresses the information needs of context-aware route recommendation systems for emergency evacuation in indoor spaces.

The article "" by Xin Xu et al. considers the influence of sentimental intensity to improve extraction accuracy in the task of emotion-cause pair extraction (ECPE) from textual input. This is achieved through the proposed SEE-3D model based 3D convolutional

neural networks and sentiment analysis, which combines clustering of emotion clauses, application of a pre-trained sentiment analysis model to compute emotional similarity, and fusion of similar features.

In "Solving the P-Second Center Problem with Variable Neighborhood Search," Dalibor Ristić et al. tackle a variant of the well-known and highly studied problem pertaining to the identification of p of the potential n center locations in such a way as to minimize the maximum distance between the users and the closest center (p-center problem). The variant, p-second center problem, minimizes the maximum sum of the distances from the users to the closest and the second closest centers. The solution is found using the proposed variable neighborhood search algorithm.

"TS-GCN: Aspect-level Sentiment Classification Model," by Shunxiang Zhang et al., addresses the problem of aspect-level sentiment classification (ASC), which refers to determining sentiment polarity of aspect words in the text. The proposed model, TS-GCN (truncated history attention and selective transformation network-graph convolutional networks) combined BERT and BiLSTM text feature extraction models, selective transformation networks for predicting implicit words, and graph convolutional networks for sentiment classification.

Hua-Yi Lin, in "Secure Cloud Internet of Vehicles Based on Blockchain and Data Transmission Scheme of Map/Reduce" addresses the issues surrounding personal information security in vehicle-to-vehicle transmission of information in open environments. The study combines blockchain to ensure the security of vehicle-based information transmission, elliptic curve Diffie–Hellman (ECDH) key exchange protocol, as well as a secure conference key mechanism with direct user confirmation combined with the back-end cloud platform Map/Reduce.

"Reinforcement Learning-based Adaptation and Scheduling Methods for Multi-source DASH" authored by Nghia T. Nguyen et al. studies video streaming from multiple sources in the dynamic adaptive streaming over HTTP (DASH) framework. The article proposed proposes two algorithms for streaming from multiple sources based on the reinforcement learning (RL) paradigm: RL-based adaptation with greedy scheduling (RLAGS) and RL-based adaptation and scheduling (RLAS). The efficiency of the proposed algorithms is demonstrated through extensive simulations with real trace data.

In "Analyzing Feature Importance for a Predictive Undergraduate Student Dropout Model," Alberto Jiménez-Macias et al. extend a previous study that proposed a predictive model to identify students at risk of dropout from the beginning of their university degree by analysing feature importance for dropout segmented by faculty, degree program, and semester in the different predictive models, as well as proposing a dropout model based on faculty characteristics. Results suggest that variables related to grade point average (GPA), socioeconomic factors and pass rate of courses have a more significant impact on the model than other factors.

Gozde Karatas Baydogmus, in "Solution for TSP/mTSP with an Improved Parallel Clustering and Elitist ACO," design a low-cost and optimized algorithm for the traveling salesman problem (TSP) by using GPU parallelization, machine learning, artificial intelligence approaches. This is achieved in three stages: clustering the points in the given dataset with K-means clustering, finding the shortest path using the ant colony approach in each of the clusters, and connecting each cluster at the closest point to the other.

In their article "Sternum Age Estimation with Dual Channel Fusion CNN Mode," Fuat Türk et al. address the problem of adult age determination through sternum multidetector computed tomography (MDCT) images using artificial intelligence algorithms. The authors propose a dual-channel convolutional neural network (CNN) architecture, which is able to predict the age groups defined as 20–35, 35–50, 51–65, and over 65 with 73% accuracy over sternum MDCT images.

Finally, "Self-Service Kits to Scale Knowledge to Autonomous Teams – Concept, Application and Limitations" authored by Alexander Poth et al. propose a self-service kit (SSK) approach that fosters team autonomy while enabling successful knowledge spread and sharing throughout a large organization. The methodology is presented and instantiated in an enterprise context which faces the challenges of handling similar topics and reinventing the wheel, at the same time needing to distill practices to make them shareable.

Computer Science and Information Systems 20(1): v–vi https://doi.org/10.2298/CSIS230100vD

Guest Editorial – Engineering of Computer Based Systems

Miodrag Djukic and Miroslav Popovic

University of Novi Sad, Faculty of Technical Sciences
Trg D. Obradovića 6, Novi Sad, Serbia
{miodrag.djukic, miroslav.popovic}@uns.ac.rs

This special section includes extended versions of selected papers from the 7th Conference on the Engineering of Computer Based Systems (ECBS 2021), organized by the University of Novi Sad, Faculty of Technical Sciences, in-cooperation with the ACM, ACM SIGAPP, and ACM SIGOPS, at the University of Novi Sad, Serbia, on May 26-27, 2021. There were 23 accepted papers in the conference, and 5 of them were selected for this special issue. All these papers were carefully revised, extended, improved, and judged acceptable for publication in this special section. Each paper has undergone a review process of two rounds; also, it has been reviewed by two referees at least. The aim of this special issue is to present some new directions and research results in the area of engineering of computer based systems.

The first paper "Multi-constrained Network Occupancy Optimization" is authored by Amar Halilovic, Nedim Zaimovic, Tiberiu Seceleanu and Hamid Feyzmahdavian. In this paper, the authors present an approach for network occupancy minimization by optimizing the packing process while satisfying multiple constraints. They formulate the minimization problem as a bin packing problem, and we implement a modification of the Best-Fit Decreasing algorithm to find the optimal solution.

The second paper "Formalization and Verification of Kafka Messaging Mechanism Using CSP" is authored by Junya Xu, Jiaqi Yin, Huibiao Zhu and Lili Xiao. In this paper, authors firstly apply the process algebra CSP and the model checking tool PAT to analyze Kafka messaging. Secondly, to further analyze the security of Kafka, they add the intruder model and the authentication protocol Kerberos model, and compare the verification results of Kafka with and without Kerberos.

The third paper "Complete Formal Verification of the PSTM Transaction Scheduler" is authored by Miroslav Popovic, Marko Popovic, Branislav Kordic and Huibiao Zhu. In this paper, authors propose a method for complete formal verification of trustworthy software, which jointly uses formal verification and formal model testing. As an example, they test the CSP model of PSTM transaction scheduler, correct and extend the CSP model, and analyze the algorithms' performance based on PAT results.

The fourth paper "Supporting 5G Service Orchestration with Formal Verification" is authored by Peter Backeman, Ashalatha Kunnappilly and Cristina Seceleanu. In this paper, authors propose a novel framework for modeling and verifying 5G orchestration, considering simultaneous access and admission of requests and virtual network function scheduling and routing. By combining modeling in user friendly UML, with UPPAAL model checking and SMTs based model finding, their framework supports both modeling and formal verification of service orchestration.

The fifth paper "Blockchain-based model for tracking compliance with security requirements" is authored by Jelena Marjanović, Nikola Dalčeković, and Goran Sladić. In

this paper, authors consider a decentralized, tamper-proof system that will provide trustworthy visibility of the SDL metrics over a certain period, to any authorized auditing party. They provide a model for creating a blockchain-based approach that allows inclusion of auditors through a consortium decision while responding to SDL use cases defined by this paper.

We gratefully acknowledge all the hard work and enthusiasm of authors and reviewers, without whom the special section would not have been possible.

Computer Science and Information Systems 20(1): vii https://doi.org/10.2298/CSIS230100viiB

Guest Editorial – Management of Digital EcoSystems

Djamal BENSLIMANE[1], Zakaria MAAMAR[2], and Ladjel BELLATRECHE[3]

[1] Claude Bernard Lyon 1 University, Lyon, France djamal.benslimane@univ-lyon1.fr
[2] University of Doha for Science and Technology, Doha, State of Qatar
[3] LIAS/ISAE-ENSMA – Poitiers University, Poitiers, France bellatreche@ensma.fr

This volume contains the revised and extended versions of papers presented at the 13^{th} International Conference on ManagEment of Digital EcoSystems (MEDES'2021), which was held virtually in Hammamet, Tunisia during the period of November 1 to 3, 2021. MEDES Conference is a platform for academics, scientists, and industry partners who get together to discuss the latest developments and challenges related to digital ecosystems in terms of resource management, data privacy, operation continuity, to mention just some. The selected papers have been reviewed by a panel of experts providing constructive feedback to their authors.

The first paper, *The Application of Machine Learning Techniques in Prediction of Quality of Life Features for Cancer Patients*, considers that training predictive Quality of Life (QoL) models in the medical field poses many challenges due to data privacy and lack of of patient data. It then analyzes classification and regression machine learning models to predict QoL indicators for breast and prostate cancer in centralized and federated learning settings. The experimental evaluation shows that long-term periods centralized models provide better predictions. It also shows that federated models perform well only for the short-term predictions.

The second paper, *Internet of Things and Agent-based System to Improve Water Use efficiency in collective irrigation*, describes an effective intelligent irrigation system based on smart sensors and multi-agents. Smart sensors collect data whereas agents take care of supervision, planning, and prediction. A real-time irrigation decision is proposed and is based on a predicted soil moisture estimated.

The third paper, *Combining Offline and On-the-fly Disambiguation to Perform Semantic-aware XML Querying*, presents a fully automated XSemSearch system for XML keyword search. Using semantic concepts of a knowledge base, both XML documents and keyword queries are transformed into semantic representations. The proposed solution exploits two distinct disambiguation strategies: offline context-based XML document disambiguation strategy and online global keyword query disambiguation strategy. Three alternative query processing algorithms to evaluate query processing time and quality are also provided.

The fourth paper, *Data-centric UML Profile for Agroecology Applications: Agricultural Autonomous Robots Monitoring Case Study*, deals with the lack of conceptual models for Internet of Things data, and proposes a UML profile that takes into account both the representation of data gathered from different kinds of devices and non-functional requirements. The feasibility and integration of the proposed UML profile in complex systems are discussed through a theoretical quality assessment and an implementation in the agroecology case study for the monitoring of autonomous agricultural robots.

We thank all reviewers for their nice and hard work and hope that readers will enjoy the content of this special issue inspiring them for more research.

Computer Science and Information Systems 20(1): ix https://doi.org/10.2298/CSIS230100ixS

Guest Editorial – Parallel and Distributed Computing and Applications

Hong Shen[1], Hui Tian[2], and Yingpeng Sang[1]

[1] School of Data and Computer Science, Sun Yat-Sen University, China,
shenh3@mail.sysu.edu.cn, sangyp@mail.sysu.edu.cn
[2] School of Information and Communication Technology, Griffith University, Australia
hui.tian@griffith.edu.au

Computational systems that can perform multiple operations or tasks simultaneously have been under many years of development and evolvement, adapting to the requirements on energy efficiency and conservation in global economy. The objective of this special collection is to publish and overview recent trends in the interdisciplinary area of parallel and distributed computing, applications and technologies. This special collection include the following 3 papers, covering topics of data locality optimization in Spark computing environment, neural network accelerators with optical computing and communication, human action recognition based on skeleton features.

The first paper "Optimizing Data Locality by Executor Allocation in Spark Computing Environment" improves the data locality in Spark computing by executor allocation for reduce stage. They firstly calculate the network distance matrix of executors and formulate an optimal executor allocation problem to minimize the total communication distance. Then, when the network distance between executors satisfies the triangular inequality, an approximate algorithm is proposed; and when the network distance between executors does not satisfy the triangular inequality, a greedy algorithm is proposed.

The second paper "Efficient Neural Network Accelerators With Optical Computing and Communication" presents a comprehensive review for the efficient photonic computing and communication in electronic Artificial Neural Networks accelerators. The related photonic devices are investigated in terms of the application in ANNs acceleration, and a classification of existing solutions is proposed that are categorized into optical computing acceleration and optical communication acceleration according to photonic effects and photonic architectures. They also discuss the challenges for these photonic neural network acceleration approaches to highlight the most promising future research opportunities in this field.

The third paper "Human Action Recognition based on Skeleton Features" proposes a novel feature descriptor, named as ExGist, to describe the skeleton information of human bone joints for human action recognition. The joint coordinates are extracted using OpenPose and the thermodynamic diagram, and ExGist is used for feature extraction. The advantage of ExGist is that it can effectively characterize the local and global features of skeleton information. By comparing the performance on different classifiers, ExGist achieves better results with an accuracy rate of 89.2%.

Computer Science and Information Systems 20(1):1–24 https://doi.org/10.2298/CSIS220210051G

Homomorphic Encryption Based Privacy-Aware Intelligent Forwarding Mechanism for NDN-VANET

Xian Guo[1], Baobao Wang, Yongbo Jiang, Di Zhang, and Laicheng Cao

School of Computer and Communication,
Lanzhou University of Technology, Gansu 730050, China
iamxg@163.com
{1244737514, 670342320, 463923303, 28140795}@qq.com

Abstract. Machine learning has been widely used for intelligent forwarding strategy in Vehicular Ad-Hoc Networks (VANET). However, machine learning has serious security and privacy issues. BRFD is a smart Receiver Forwarding Decision solution based on Bayesian theory for Named Data Vehicular Ad-Hoc Networks (NDN-VANET). In BRFD, every vehicle that received an interest packet is required to make a forwarding decision according to the collected network status information. And then decides whether it will forward the received interest packet or not. Therefore, the privacy information of a vehicle can be revealed to other vehicles during information exchange of the network status. In this paper, a Privacy-Aware intelligent forwarding solution PABRFD is proposed by integrating Homomorphic Encryption (HE) into the improved BRFD. In PABRFD, a secure Bayesian classifier is used to resolve the security and privacy issues of information exchanged among vehicle nodes. We informally prove that this new scheme can satisfy security requirements and we implement our solution based on HE standard libraries CKKS and BFV. The experimental results show that PABRFD can satisfy our expected performance requirements.

Keywords: VANET, Bayesian decision theory, BRFD, homomorphic encryption.

1. Introduction

VANET has the characteristics of high-speed movement of vehicle nodes and frequent changes of network topology, which will cause frequent disconnection of the link between vehicle nodes [1, 2]. To ensure the reliability and stability of network connection, the routing protocol is a key to affect the performance of the VANET. In recent years, many intelligent solutions based on machine learning have been proposed [3-9]. Literature [10] made a comprehensive research on the security and management challenges for applying machine learning in VANET. To improve the VANET performance, the Bayesian classification algorithm has been widely used in predicting vehicle behavior [11-15]. In [16], we proposed a Bayesian-based Receiver Forwarding

[1] Corresponding author

Decision (BRFD) scheme to solve the broadcast storm in Named Data Vehicular Ad Hoc Networks (NDN-VANET).

Although machine learning has been widely used for VANET and other environments [17], the security and privacy problems in machine learning have been a focus of academia and business. A comprehensive investigation of privacy and security issues in machine learning is made in the literature [18]. Machine learning consists of two stages: the training stage and the testing stage. The poisoning attack[19, 20] is the best-known attack method in the training stage, and the attacks aiming at the testing stage include the membership inference attack[21], the evasion attack [22], and the model extraction attack[23, 24]. In this paper, we study security and privacy issues in the testing stage of BRFD.

The privacy-preserving machine learning (PPML) is firstly proposed by Lindell et al. [25]. The PPML allows two participants to extract the joint dataset without revealing their privacy. The early researches of PPML mostly used Yao's garbled circuit protocol [26], which has a large computational and communicating overhead. At present, the privacy protection technologies to achieve PPML have three broad categories, which are based on differential privacy (DP) [27-31], secure multi-party computation (SMPC) [32-38], and homomorphic encryption (HE) [39-44]. Adding noise to the sensitive data is a key method of differential privacy to achieve privacy protection. However, the increasing noise may lead to an accuracy decrease of machine learning. The schemes based on secure multi-party computation require multiple information interactions between participants, which is not suitable for the NDN-VANET with fast topology changes. Homomorphic encryption supports calculation on ciphertext, which can ensure the classification accuracy of the Bayesian model. Therefore, this paper adopts homomorphic encryption mechanism to solve security and privacy issues in BRFD we proposed in [16].

In BRFD, the vehicles exchange network status information in plaintext to make forwarding decision and no cryptographic mechanism is used in BRFD. The vehicle's privacy information such as location, speed, and so forth can be revealed to other vehicles. Aiming at the security and privacy issues caused by information exchange of network status used by machine learning in BRFD, a Privacy-Aware intelligent forwarding solution PABRFD is proposed by integrating HE [45] into BRFD in this paper. In PABRFD, HE is integrated into the Bayesian-based forwarding decision to protect the network status information of the vehicle. In addition, we improve the calculation method of the Bayesian probability values in BRFD and enhance the efficiency of the classification protocol by removing the complicated Gaussian formula calculation. We implement our novel scheme based on the CKKS library and BFV library [46] and make a performance comparison. We also informally analyze the security attributes of the PABRFD.

The remainder of this paper is structured as follows. The related works are reviewed in Section 2. The theoretical knowledges related to PABRFD are introduced in Section 3. The detailed PABRFD is described in Section 4. The experimental results of the PABRFD scheme are analyzed in Section 5. Finally, the conclusion and future work are introduced in Section 6.

2. Related Works

At present, researchers have proposed many privacy-preserving solutions based on HE aiming at machine learning for various applications. In this section, we review some solutions for VANET. In the vehicle-to-everything (V2X) communication system, in order to realize intelligent communication, vehicles and infrastructure equipment need to exchange data regularly. Therefore, the confidentiality and integrity of data need to be protected in an unverified and untrusted environment. Ulybyshev et al. [39] proposes a HE-based secure data exchange mechanism to protect the communication privacy between vehicles. The solution provides an access control scheme based on roles and attributes, which can detect and prevent data leakage caused by internal users. In addition, the authors propose a search method based on HE, which can query a vehicle's record stored on an untrusted cloud server based on ciphertext. The authors prove that their solution can protect vehicle and its owner's sensitive information against curiosity or malicious attacks.

Kong et al. [40] proposes a HE-based VANET secure data sharing scheme to protect a vehicle's private data. Each vehicle node is required to build a comprehensive data report and send the data report to RSU for secure data aggregation. Finally, the aggregated results will are sent and stored in a traffic management agency. After receiving a data query request, the RSU will share the aggregated result with the vehicle node.

In VANET, a reputation system often is used to judge whether a vehicle agrees to communicate with the target vehicle or not, according to the feedback information of other vehicles. So the feedback information plays a crucial role in the trust evaluation of neighbor nodes. In [41], a privacy-preserving vehicle feedback (PPVF) scheme is proposed based on HE and the data aggregation technique for VANET with cloud assistant system. The cloud service provider obtains the parameters related to vehicle in the vehicle feedback information, which is used for reputation calculation without revealing the private information of the vehicle that provides the feed information. Theoretical analysis and simulation experiment show that PPVF can achieve privacy protection for the feedback vehicle and PPVF has acceptable computational accuracy and communication consumption.

As machine learning-based routing algorithm is widely used in VANET, routing scheme also faces various security threats. An opportunistic routing protocol (ePRIVO) for vehicular delay-tolerant networks (VDTN) based on HE is proposed in [42]. The ePRIVO can protect some sensitive information during a routing decision of a vehicle. The ePRIVO models VDTN as a time-varying neighboring graph, and the graph's edge corresponds to the neighboring relationship between vehicles. In the ePRIVO, vehicles use HE to calculate the graph's similarity and secretly compare route metrics. Furthermore, their experimental results and analysis show that the accuracy of the ePRIVO is about 29% higher than other related routing protocols for privacy-preserving.

Alamer et al. [43] propose a privacy-preserving bidding framework VCC for VANET based on HE to protect the private interaction between a vehicle and a cloud server. An incentive mechanism is used in the bidding framework to encourage the interaction between the cloud server and the vehicle. The cloud server selects a participation vehicle to complete a task in cooperation. The selected vehicle will receive a certain

reward after the task is completed. Moreover, this mechanism ensures the authenticity of all participants and provides an allocation rule that enables the VCC framework to select the best resources for the task. In addition, due to using the HE technology, VCC and RSU can run an effective bidding process without acquiring the sensitive information of a vehicle.

A decentralized privacy-preserving deep learning model (DPDL) is proposed by integrating deep learning, blockchain, and FHE into VANET in [44]. DPDL can effectively reduce network communication overhead and congestion delay by decomposing computing tasks from a centralized cloud service to edge computing (EC) nodes. Blockchain is used to establish a secure and reliable data communication mechanism between RSU and EC nodes. In addition, the DPDL model provides a privacy-preserving data analysis scheme for VANET, and the fully homomorphic encryption (FHE) is used to encrypt the traffic data on each EC node and input it to the local DPDL model, thereby it can effectively protect the privacy and trustworthiness of the vehicle. Using of Blockchain can provide a reliable distributed update mechanism for the DPDL model, and the parameters of each local DPDL model are stored in the blockchain to share with other distributed models. In this solution, all distributed models can update their models in a reliable and asynchronous manner.

3. Preliminaries

3.1. Bayesian Theory Foundation

Bayesian Classification Algorithm

Bayesian classification algorithm [47], which is widely used for sample classification, is based on the Bayes theorem. The Naive Bayesian classification algorithm is one of the Bayesian classification algorithms. The Naive Bayes classification algorithm assumes that each attribute value is independent with the others and does not affect the classification results. The idea of the Naive Bayes classification algorithm is shown as follows.

Let $X = \{x_1, x_2, \cdots, x_m\}$ is an item to be classified, each $x_i (i = 1,2,\cdots, m)$ is a feature value of X. Given a set of categories $Y = \{y_1, y_2, \cdots, y_n\}$ where each $y_i (i = 1,2,\cdots, n)$ represents a category. If the posterior probability $P(y_k|X) = max\{P(y_1|X), P(y_2|X), \cdots, P(y_n|X)\} P(y_i|X) (i = 1,2,\cdots, n)$, X belongs to the *k*-th classification.

The posterior probability of each category $P(y_i|X)$ is shown as follows.

$$P(y_i|X) = \frac{P(X|y_i)P(y_i)}{P(X)}. \quad (1)$$

$P(X|y_i)$ is the prior probability of X, $P(y_i)$ is the probability of each category.

Bayesian Classifier

The Bayesian classifier [48] is a simple probabilistic classifier based on a Naive Bayesian algorithm. In this classifier, the model w consists of several probabilities: $\{P(y_i)\}_{i=1}^{n}$ is the probability of each category y_i, and $\{\{P(x_j|y_i)\}_{i=1}^{k}\}_{j=1}^{m}$ is the prior probability (When the X belongs to the i-th category y_i, the j-th feature value of X is x_j. m is the dimension of the X, k is the total number of categories). The classifier selects the category of the highest posterior probability as the final decision result, and the decision result is denoted as k_0.

$$\begin{aligned} k_0 &= P(y_i|X) \\ &= \underset{i\in[k]}{argmax} \frac{P(X|y_i)P(y_i)}{P(X)} \\ &= \underset{i\in[k]}{argmax} P(X|y_i)P(y_i). \end{aligned} \tag{2}$$

In the above formulation, removing the denominator $P(X)$ does not affect the final result due to the characteristics of the *argmax* function. In the Naive Bayes classifier, the m feature values of X are independent of each other, k_0 is shown as follows.

$$k_0 = \underset{i\in[k]}{argmax} P(y_i) \prod_{j=1}^{m} P(x_j|y_i) . \tag{3}$$

3.2. BRFD

Due to the characteristics of faster computing speed and higher classification accuracy, the Bayesian classification algorithm is widely used in VANET to improve network performance by predicting vehicle's behavior. In [16], we proposed a scheme called BRFD based on the Naive Bayes classifier to mitigate the broadcast storm problem incurred by interest packets in NDN-VANET. The BRFD mainly consists of three stages: the HELLO interaction, the Naive Bayesian decision, and the back-off forwarding.

HELLO Interaction

In BRFD, the special interest packets with the HELLO tag are used to regularly exchange the network status information $C = \{(x_i, y_i), speed_i, dis_i, num_i, D_i\}$ between neighboring nodes. In C, (x_i, y_i) denotes the vehicle location, $speed_i$ denotes the vehicle speed, dis_i denotes the distance, num_i denotes the number of neighbor vehicles, and D_i denotes the Bayesian Decision result. When the neighbor vehicle receives the HELLO packet, it will store the vehicle status information in its Decision Neighbor List (DNL). DNL is shown as follows.

$$DNL = \begin{pmatrix} x_1 & y_1 & speed_1 & dis_1 & num_1 & D_1 \\ x_2 & y_2 & speed_2 & dis_2 & num_2 & D_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n & y_n & speed_n & dis_n & num_n & D_n \end{pmatrix} \quad (4)$$

Naive Bayesian Decision

In BRFD, after a vehicle *a* receives an interest packet from the neighboring vehicle *s*, firstly, the vehicle *a* reads the network status information $C = \{(x_i, y_i), speed_i, dis_i, num_i, D_i\}$ of the vehicle *s* to obtain the Bayesian decision condition $\{dis(s,a), speed_a, dis_a, num_a\}$, and then calculates the forwarding probability of $P(F|C)$ and the non-forwarding probability $P(\bar{F}|C)$ of the received interest packet according to the DNL. Finally, the vehicle decides whether it will enter the back-off forwarding process or not by comparing the value of the $P(F|C)$ and the $P(\bar{F}|C)$.

Back-off Forwarding

The BRFD is a scheme based on the receiver-forwarding decision. Each vehicle does not know whether other vehicles also will forward the received interest packet. Therefore, a conflict may be incurred by the same copies of an interest packet due to multiple vehicle nodes forward the same interest packet. To solve this problem, a back-off forwarding mechanism is adopted. Each node will set a back-off delay according to its forwarding probability calculated in the Naive Bayesian decision stage.

3.3. Homomorphic Encryption

HE[49] is a cryptographic technique based on a certain mathematical problem. The detailed description of HE is as follows. Let p_1 and p_2 be two plaintexts in the plaintext space M, *PK* is the public key and *SK* is the private key respectively. *Enc* and *Dec* are two algorithms for encryption and decryption respectively. "⊙" represents operation on ciphertexts, "·" represents operation on plaintexts. The HE operation satisfies the following equation.

$$p_1 \cdot p_2 = Dec_{SK}(Enc_{PK}(p_1) \odot Enc_{PK}(p_2)) . \quad (5)$$

HE is one of the common privacy protection technologies. Due to the computation on the two ciphertexts is equal to the direct computation on the two plaintexts as shown in equation (5), it is widely used for privacy-preserving in machine learning applications. At present, HE can be divided into Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), and Fully Homomorphic Encryption (FHE). PHE only allows one type of operation but does not limit the number of operations. The classic cases of PHE are EI-Gamal[50], Paillier[51] and RSA[52]. SHE allows multiple types of operations, but the number of operations is limited. The classic

case of SHE is BGN[53]. FHE does not limit the type and number of operations. The classic case of FHE is Gentry[45].

The SEAL library provides two fully homomorphic encryption schemes: the BFV scheme and the CKKS scheme [54]. The BFV scheme supports integers operation, and the CKKS scheme supports floating-point numbers operation. There are a lot of floating-point numbers in the process of Bayesian classification probability calculation. So, when the BFV scheme is used, it is necessary to convert the eigenvalues in the motion data to integers by an expansion factor. This process will lead to a loss of data accuracy, further reducing the model classification accuracy. So, we adopt the CKKS scheme to implement our PABRFD scheme in this paper.

CKKS allows the addition and multiplication of encrypted real or complex numbers. It is a common scheme for privacy-preserving in machine learning applications. CKKS includes six basic algorithms: Key generation algorithm (*KeyGen*), Encode algorithm (*Ecd*), Decode algorithm (*Dcd*), Encryption algorithm (*Enc*), Decryption algorithm (*Dec*), and ciphertext calculation algorithm (*Eval*).

The CKKS is designed based on RLWE. The plaintext space of RLWE is a polynomial ring $\mathbb{R} = \mathbb{Z}[X]/\Phi_M(X)$ (Here, $\Phi_M(X)$ is the M-th cyclotomic polynomial of degree $N = \Phi(M)$, M is a positive integer). However, the plaintext space of CKKS is a complex vector space $\mathbb{C}^{\Phi(M)/2}$. Therefore, it is necessary to find the mapping relationship between the two. CKKS defines the canonical embedding mapping σ: $\mathbb{R} \to \mathbb{H}$ and the natural projection π: $\mathbb{H} \to \mathbb{C}^{\Phi(M)/2}$, both π and σ are full mappings, and their inverse mappings are $\pi^{-1} and\ \sigma^{-1}$. Let $\mathbb{H} = \left\{(z_j)_{j \in \mathbb{Z}_M^*} : z_{-j} = \overline{z_j}, \forall j \in \mathbb{Z}_M^*\right\} \subseteq \mathbb{C}^{\Phi(M)}$, Let T be a multiplicative subgroup of $\mathbb{Z}_M^*$ satisfying $\mathbb{Z}_M^*/T = \{\pm 1\}$. For a vector $z \in \mathbb{C}^{\Phi(M)/2}$, the encoding procedure first expands it into the vector $\pi^{-1}(z) \in \mathbb{H}$, and then computes its discretization to $\sigma(\mathbb{R})$ after multiplying a scaling factor Δ. At last, the corresponding integral polynomial is $m(X) = \lfloor \Delta \cdot \pi^{-1}(z) \rceil_{\sigma(\mathbb{R})}$. The decoding procedure of CKKS is the inverse process of the encoding algorithm, so it will not be introduced in detail.

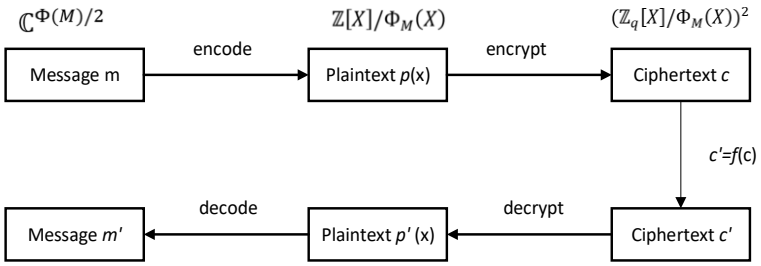


Fig.1. Overview of CKKS Scheme

An overview of the CKKS scheme is shown as follows Fig.1: A message m that used to perform a specific computation is first encoded into plaintext polynomial $p(X)=Ecd(m)$ and then is encrypted by using the public key. Once a message m is encrypted into a ciphertext $c=Enc(p(X))$, the CKKS scheme provides addition and multiplication operations. A combination of homomorphic operations is denoted as f. Decrypting c' with the secret key will obtain $p'(X)=Dec(c')$ and then decoding to $m'=Dcd(p'(X))$. More detailed techniques are mentioned in [54].

3.4. Argmax Protocol

The *argmax* function is usually used to obtain the subscript of the maximum value in the set. In [55], the authors combine the *argmax* protocol with HE to obtain the subscript of the maximum value in the Bayesian classification results. Assuming a participant *A* holds a set of encrypted probability values encrypted with the counterpart's public key *pk*, and the other participant *B* has a decryption key *sk*. For convenience, $[[x]]$ denotes $Enc_{pk}(x)$. Let $[[a]]$ and $[[b]]$ denote two ciphertexts to be compared. Let $\mathcal{G}$ denotes a set of a linear polynomial from $g(x)=Cx$ where *C* is a positive integer. Since $g \in \mathcal{G}$ is a linear polynomial with positive coefficients, we obtain $g(a) - g(b) \geq 0$ when $a \geq b$ and $g(a)-g(b) < 0$ otherwise.

When comparing two ciphertext values, *A* randomly chooses $g \in \mathcal{G}$ and computes $[\![h]\!]=[\![g(a)]\!] \ominus [\![g(b)]\!]=g([\![a]\!]) \ominus g([\![b]\!])$. Then *A* sends $[\![h]\!]$ to *B* who decrypts it to obtain *h*. if $h \geq 0$, *B* updates $index$ and generates a ciphertext $[\![d]\!]=[\![1]\!]$, which is an encryption of vector whose values are all 1. Otherwise, *B* will not update $index$ and generate $[\![d]\!]=[\![0]\!]$, which is an encryption of vector whose values are all 0. *B* sends $[\![d]\!]$ to *A*, who then computes $([[d]] \otimes [[a]]) \oplus ((1 \ominus [[d]]) \otimes [[b]])$. The resulting ciphertext is the higher value and will be used in the next comparison. The comparison is repeated until all values have been compared. Once all comparisons are made, *B* sends the *index* to *A*, who reverses the permutation to obtain the actual *index i*.

4. Our Solution PABRFD

In this section, we present our new solution PABRFD. To explain our PABRFD, we firstly describe a simplified NDN-VANET network model as shown in Fig. 2. In this network model, the vehicle nodes are roughly divided into two categories: The vehicle node *R* and the neighbor vehicle nodes of *R* (They are denoted by a set $S = \{S_1, S_2, \cdots, S_n\}$). Here, we assume that the vehicle node *R* received an interest packet from some adjacent vehicle node S_i and then it needs to decide whether it should forward the received interest packet or not according to our PABRFD solution. The neighboring vehicles of *R* in S will periodically provide network status information to *R* by a special HELLO packet and *R* will store these information in DNL according to BRFD [16]. The network status information is important data that is used to make an intelligent forward decision by the vehicle *R*.

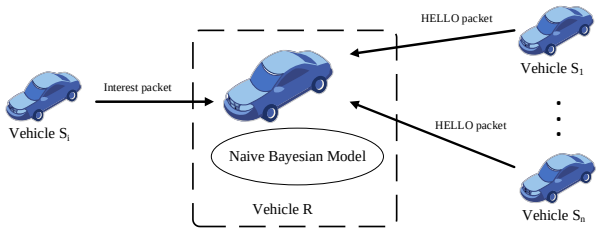


Fig.2. The Simplified Network Model

4.1. Security Requirements

To ensure secure and privacy exchange of network status information, we assume that the vehicle nodes in the set *S* are semi-honest, so the PABRFD scheme should satisfy the following security requirements:

1. **System correctness**: A vehicle node that received an interest packet can correctly make a forward decision.
2. **Data integrity**: It is required that the content in a HELLO packet cannot be modified when it is transmitted in NDN-VANET.
3. **Data confidentiality**: Some sensitive information such as the vehicle location and speed and so forth, will not be revealed when the HELLO packet is transmitted in NDN-VANET.

4.2. System Building Blocks

To achieve these security requirements mentioned in section 4.1. We propose a novel scheme PABRFD based on BRFD [16] by integrating security mechanisms such as HE into BRFD in this paper and by improving BRFD. This PABRFD consists of the following modular.

Key Generation

In PABRFD, all vehicle nodes need to initialize their own public key, private key, and evaluation key. We assume that λ is a security level parameter, and L represents the upper limit of the ciphertext length. Choose a large integer $p>0$ as the base and q_0 as the modulus, let $q_l = p^l \cdot q_0, 0 < l \leq L$, l is the ciphertext depth. For a real number $\sigma>0$, $Dg(\sigma^2)$ extracts a vector in $\mathbb{Z}^N$ by drawing its coefficient from the discrete Gaussian distribution of variance σ^2. For a real number $0\leq\rho\leq1$, the distribution $ZO(\rho)$ draws each entry in the vector from $\{0,-1,1\}^N$, where the probability of selecting 1 and -1 is $\rho/2$, and the probability of selecting 0 is $1-\rho$. For a positive integer h, $HWT(h)$ is the set signed N-dimensional binary vector set $\{0,1,-1\}^N$ whose Hamming weight is exactly h.

Key generation algorithm $Keygen(1^{\lambda})\text{->}(pk, sk, evk)$: Generate a secret key sk, a public key pk for encryption, and an evaluation key evk.

According to the security parameters λ and q_L, choose a power-of-two $M = M(\lambda, q_L)$ for cyclotomic polynomial, an integer $h = (\lambda, q_L)$, integer $P = P(\lambda, q_L)$, real number $\sigma = \sigma(\lambda, q_L)$.

$$s \leftarrow HWT(h), a \leftarrow \mathbb{R}_{q_L}, e \leftarrow Dg(\sigma^2). \tag{6}$$

Let the private key $sk \leftarrow (1, s)$, the public key $pk \leftarrow (b, a) \in \mathbb{R}_{q_L}^2$, where $b \leftarrow -a \cdot s + e \ (mod\ q_L)$, $\mathbb{R}_{q_L}$ represents a polynomial ring. Sample $a' \leftarrow \mathbb{R}_{p \cdot q_L}, e' \leftarrow Dg(\sigma^2)$, and then the evaluation key evk is

$$evk \leftarrow (b', a') \in \mathbb{R}_{P \cdot q_L}^2. \tag{7}$$

where $b' \leftarrow -a' \cdot s + e' + P \cdot s^2 (mod\ P \cdot q_L)$.

Bayesian Model Construction

The Bayesian model consists of the prior probability of each feature attribute and the class probability of each category. We assume that $P(F_i)(i = 1,2)$ denotes Interest Forwarding Event, F_1 means that the vehicle node R will forward the received interest packet, and F_2 means that the vehicle node R won't forward the interest packet.

Let N_{F_1} be the number of interest packets forwarded in the training set. Let N_{F_2} be the number of interest packets not forwarded in the training set, then the forwarding and non-forwarding probabilities are shown as follows.

$$P(F_i) = \frac{N_{F_i}}{N_{F_1} + N_{F_2}} \ (i = 1,2) \ . \tag{8}$$

Let the $N_{(dis^2|F_i)}$ is the number of the *dis* in the training set that the decision result is $F_i(i = 1,2)$. Let the $N_{(num|F_i)}$ is the number of the *num* in the training set that the decision result is $F_i(i = 1,2)$. Let the $N_{(speed|F_i)}$ is the number of the *speed* in the training set that the decision result is $F_i(i = 1,2)$. Then the prior probability of each feature value is shown as follows.

$$\begin{aligned} P(dis^2|F_i) &= \frac{N_{(dis^2|F_i)}}{N_{F_i}} \ . \\ P(\text{num}|F_i) &= \frac{N(num|F_i)}{N_{F_i}} \ . \\ P(speed|F_i) &= \frac{N(speed|F_i)}{N_{F_i}} \ . \end{aligned} \tag{9}$$

The vehicle node R stores the probability of classification $F_i(i = 1,2)$ and the prior probability of each feature value into a vector of length t+1(t is the number of feature values, t=3 in this scheme), $W_{F_i}[j] = P(C_j|F_i)$,$W_{F_i}[t+1] = P(F_i)(i = 1,2,j = 1,2,3)$, $C = \{dis^2, speed, num\}$, C_j represents the j-th element in C. In model $W = \{w_1, w_2\}$, w_1 and w_2 represent forwarding and non-forwarding categories respectively, and the model matrix W is shown as follows.

$$Model\ W = \begin{bmatrix} P(C_1|F_1) & P(C_2|F_1) & P(C_3|F_1) & P(F_1) \\ P(C_2|F_2) & P(C_2|F_2) & P(C_3|F_2) & P(F_2) \end{bmatrix} \tag{10}$$

4.3. Implement of PABRFD

In this section, we present the implementation details of the PABRFD scheme. PABRFD can be divided into the following four stages: System Initialization stage, Network Status Information Exchange stage, Naive Bayes Decision stage, and Secure Argmax. A detail flow chart of the PABRFD is described in figure 3 :

System Initialization

In this system initialization process, we firstly initialize cryptographic materials as described in section 4.2.1. And then we will prepare the Bayesian model for PABRFD according to section 4.2.2. In our novel scheme, we use SUMO traffic simulation software [56] to generate vehicle motion data. The simulation area of the network road is set to 200m*200m. The number of vehicle nodes in the network is 10-60. The speed of the vehicle is between 0-100 mph. In BRFD [16], *dis*, *speed*, and *num* are regarded as continuous values and assume that these continuous values obey a Gaussian distribution of mean μ and variance σ^2. The conditional probability of each feature value is calculated by the following formula.

$$g(x,\mu,\sigma) = \frac{1}{\sqrt{2\Pi}\sigma} e^{-(\frac{(x-\mu)^2}{2\sigma^2})}. \quad (11)$$

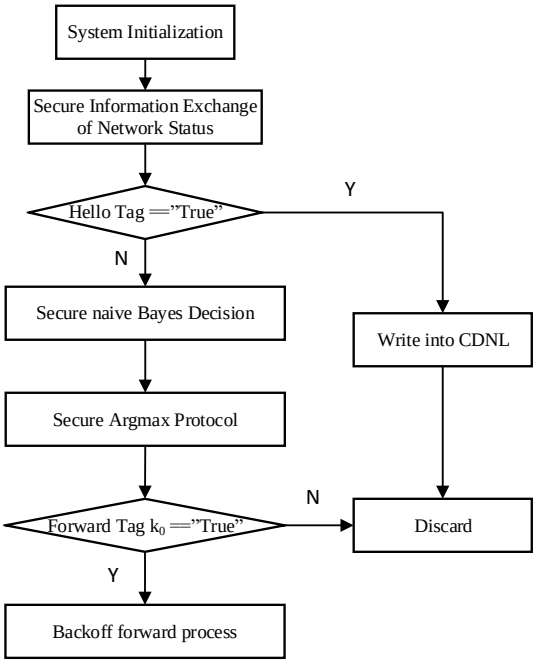


Fig.3. PABRFD flow chart

In PABRFD, to improve the computational efficiency of HE, we choose to discretize *dis*, *speed*, and *num*. The value of *dis* is between 0 and $200\sqrt{2}$. In our simulation experiment, we use the square of *dis* as a feature value to train the model. Therefore, the value of dis^2 is between 0 and 80000, we divide it into 400 intervals, and replace the entire interval with the middle value of each interval. The set of dis^2 after division is $\{100,300,\cdots,79900\}$. In the same way, the value of *speed* is between 0 and 100, we divide it into 10 intervals, and the set of *speed* after division is $\{5,15,\cdots,95\}$. The *num* is between 0-60, we divide it into 10 intervals, and the set of *num* after division is

$\{0,3,9,15,\cdots,57\}$ (When the number of neighboring nodes is 0, the node cannot forward interest packets, so 0 is a special value and needs to be divided separately).

The algorithm of the Bayesian model training stage is shown in Algorithm 1:

Secure Information Exchange of Network Status

In PABRFD, all nodes in the network need to periodically send their network status information to their neighbor nodes. We assume that some nodes $S_i(i = 1,2,\cdots)$ need to send its network status information. It will firstly encode the network status information $C = \{(x_i, y_i), speed_i, dis_i, num_i, D_i\}$ into polynomial $m(C)$ and then encrypts $m(C)$, the encryption process does as follows:

Sample $v \leftarrow ZO(0.5)$, $e_0, e_1 = Dg(\sigma^2)$.

Final output:

$$c \leftarrow v \cdot pk_{S_i} + (m(C) + e_0, e_1)(mod\ q_l)\ . \tag{12}$$

And then the node S_i writes c into an interest packet with a HELLO tag and broadcasts the interest packet to its neighboring nodes. The vehicle node that received the HELLO packet will update Ciphertext Decision Neighbor List (CDNL) according to the information carried in this HELLO packet.

Algorithm 1: Bayesian model training algorithm

Input: Vehicle motion datasets $C = \{(x_i, y_i), speed_i, dis_i, num_i, D_i\}$
Output: Model *W*
//Data discretization
$dis^2 : (0,80000) \rightarrow \{100,300,\cdots,79900\}$
$speed : [0,100] \rightarrow \{5,15,\cdots,95\}$
$num : [0,60] \rightarrow \{0,3,9,15,\cdots,57\}$
//Calculate the class probability and the prior probability of each feature attribute
for *i*=1 to 2:
$P(F_i) = \frac{N_{F_i}}{N_{F_1} + N_{F_2}}$. //Calculated the class probability
$P(dis^2|F_i) = \frac{N_{(dis^2|F_i)}}{N_{F_i}}$.
$P(\text{num}|F_i) = \frac{N(num|F_i)}{N_{F_i}}$. // Calculate the prior probability
$P(speed|F_i) = \frac{N(speed|F_i)}{N_{F_i}}$.
end for
//Build the model matrix *W*
for *i*=1 to 2:
for *j* = 1 to 3:
$W_{F_i}[j] = P(C_j|F_i)$ // Write the prior probability
end for
$W_{F_i}[j] = P(F_i)$ // Write the class probability
end for
Return model *W*

Secure Bayesian Decision

We assume that a vehicle node *R* receives an interest packet from some vehicle node $S_i(i = 1,2,\cdots)$that want to request some interested data on the network, the vehicle node *R* will extract the encrypted network status information $\{([[x_s]],[[y_s]]),[[speed_s]],[[num_s]],[[D_s]]\}$ from CDNL. And then the node *R* will use homomorphic encryption to calculate the values used in the secure Bayesian Decision. Firstly, it calculates the square of the distance [[dis^2(*S*,*R*)]], and then constructs the item $[[C]] = \{[[dis^2(S,R)]], speed_R, num_R\}$ to be classified. For each class F_i(*i*=1,2), the node *R* matches [[*C*]] with the Naive Bayesian model *W*={w_1, w_2} and calculates the posterior probability [[$P(F_i|C)$]].The [[dis^2(*S*,*R*)]] is shown as follows.

$$\begin{aligned}[[dis^2(S,R)]] &= ([[x_S]] - x_R)^2 + ([[y_S]] - y_R)^2 \\ &= [[x_S]]^2 + [[y_S]]^2 - 2 \cdot x_R \cdot [[x_S]] - 2 \cdot y_R \cdot [[y_S]] + x_R^2 + y_R^2 .\end{aligned} \quad (13)$$

The posterior probability $P(F_i|C)$ can be calculated by the following formula:

$$[[P(F_i|C)]] = \frac{[[P(C|F_i)]] \cdot P(F_i)}{P(C)} (i = 1,2) . \quad (14)$$

In the above formula, $P(F_i)$ is the class probability, which has been obtained in the Bayesian training stage. Due to the characteristics of the *argmax* protocol (Details are shown in 3.1.2), $P(C)$ does not affect the final result, so it is not necessary to calculate. We assume that the feature values of the network status information are independent, so $P(C|F_i)$ can be calculated by the following formula:

$$[[P(C|F_i)]] = [[P(([[dis^2(S,R))]]|F_i)]]P(speed_R|F_i)P(num_R|F_i) . \quad (15)$$

The three conditional probabilities can be calculated by the formula (9) introduced in the Bayesian model training stage. Bring the calculation result of the formula (15) into the formula (14) to obtain the posterior probability [[$P(F_i|C)$]].

The detailed algorithm is shown as follows.

Algorithm 2: Secure naive Bayes Decision

Input: Items to be classified $[[C]] = \{[[dis^2(S,R)]], speed_R, num_R\}$, natural projection π, Naive Bayesian model *W*={w_1, w_2}

Output: The ciphertext sequence $\{[[P_{\pi(i)}]]\}$

//calculates the forwarding and non-forwarding probability [[$P(F_i|$C)]](i=1,2)

for *i*=1 to 2 do

temp ←[[C]] ⊗*W*, //Match every probability

[[$P(F_i|C)$]]←add(*temp*)

end for

The ciphertext sequence is

$$\{[[P_{\pi(i)}]]\} = [[P(Fi|C)]] = P(F_i)\prod_{j=1}^{t}[[P(C_j|F_i)]]\}_{i\in[1,2],j\in[1,t]}$$

Secure Argmax Protocol

After calculating the classification probability $\{[[P_{\pi(i)}]]\}(i = 1,2)$, the vehicle node R needs to interact with its neighboring vehicle nodes in S to obtain the final classification result k_0. That is to say, we need to compute the maximum value in these classification probabilities $\{[[P_{\pi(i)}]]\}(i = 1,2)$.

The *argmax* protocol is executed between the vehicle node S_i and the vehicle R, and the final forwarding decision result k_0 is output.

As mentioned in sections before, the interaction process between the vehicle node R and some neighboring node S_i is shown in figure 4.

The detailed algorithm is shown as follows.

Algorithm 3: Secure argmax protocol

R's input: Encrypted category probability $\{[[P_{\pi(i)}]]\}_{i\in[1,2]}$, Polynomial set $\mathcal{G}$

S's input: private key SK_S, and public key PK_S

Output: Subscript of maximum probability k_0

S:

index ←1

R:

$[[max]] \leftarrow [[P_{\pi(1)}]]$

for *i*=1 to 2 do

R:

Random sampling $g \in \mathcal{G}$

$[[temp]] \leftarrow g([[P_{\pi(i)}]]) \ominus g([[max]])$

send [[*temp*]] to *S*

S:

$temp \leftarrow Dec_{SK_S}([[temp]])$

if *temp*≥0 : *r*←1,*index*←*i*

if *temp*<0: *r*←0

$[[r]] \leftarrow Enc_{PK_S}(r)$

send [[*r*]] to *R*

R:

$[[max]] \leftarrow ([[r]] \otimes [[P_{\pi(i)}]]) \oplus ((1 \ominus [[r]]) \otimes [[max]])$

end for

S:

send *index* to *R*

R:

$k_0 = \pi^{-1}(index)$

Output category F_{k_0} is the final classification result

5. Security and Performance Analysis

5.1. Security Analysis

We informally prove that PABRFD can satisfy the security requirements proposed in 4.1.

Proposition 1. System correctness: Any vehicle node that received an interest packet can correctly make the forward decision.

***Proof*:** We assume that vehicle nodes are semi-honest in PABRFD, that is to say, a node will also follow the requirements of a protocol even if it is controlled by an attacker. In addition, PABRFD only introduces cryptographic mechanism to protect privacy in contrast with BRFD. So, the correctness of BRFD can ensure the correctness of PABRFD.

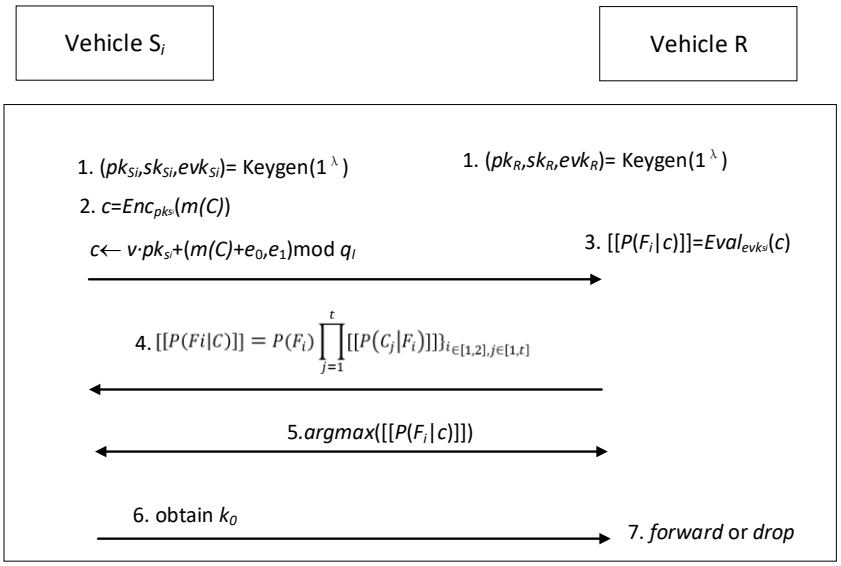


Fig.4. The interaction process between S_i and R

Proposition 2. Data integrity: Any malicious vehicle node can't modify data in the HELLO packet when it is transmitted in NDN-VANET.

***Proof*:** In PABRFD, HE is used to protect privacy in a HELLO packet when a vehicle node exchanges network status information with other vehicles. That is to say, a vehicle node will encrypt network status information such as location, speed, and so forth in the HELLO packet. Therefore, HE actually is a public key encryption scheme. So, using of HE in PABRFD can guarantee data integrity in the HELLO packet. Of course, data in the HELLO packet can't be modified by other nodes that they are malicious in semi-honest.

Proposition 3. Data confidentiality: Some sensitive information such as the vehicle location and speed and so forth, will not be revealed when the HELLO packet is transmitted in NDN-VANET.

***Proof*:** the proof is similar to proposition 2. The information in the HELLO packet is encrypted by using HE when it is transmitted in NDN-VANET or stored in CNDL. Therefore, the vehicle node without the corresponding decryption key cannot obtain the sensitive information of other vehicle nodes.

5.2. Performance Analysis

In this section, we evaluate PABRFD and two relative schemes such as BRFD and BFV-BRFD in terms of computational and communication efficiency. The BFV-BRFD that we specially designed to compare the performance with PABRFD is a scheme based on BRFD.

Experimental data and environment

The experimental data comes from SUMO [56] traffic simulation software. We use the ndnSIM [57] platform to evaluate and analyze our scheme PABRFD and the relative schemes. The simulation area of the network road is set to 200m*200m. The number of the vehicle nodes is 10-60, and the speed of the vehicle is set between 0-100 mph. The duration of each experiment is set to 100s, and the experiment results are average of 10 experiments we do.

Classification performance

As shown in the Table 1 below, we conduct experimental tests from different dimensions to evaluate the performances of BRFD, PABRFD, and BFV-BRFD schemes. We use three common metrics such as *Precision*, *Recall*, and *Accuracy* as follows.

$$\begin{aligned} Precision &= \frac{TP}{TP+FP}. \\ Recall &= \frac{TP}{TP+FN}. \\ Precision &= \frac{TP+TN}{TP+TN+FP+FN}. \end{aligned} \tag{16}$$

Here, parameters *TP*, *TN*, *FP*, and *FN* are explained as follows.

TP(*True Positive*): The true class of the sample is positive, and the predicted result is positive.

TN(*True Negative*): The true class of the sample is negative, and the predicted result is negative.

FP(*False Positive*): The true class of the sample is negative, and the predicted result is positive.

FN(*False Negative*): The true class of the sample is positive and the predicted result is negative.

Our experimental results in Table 1 show that the *Precision*, *Recall*, and *Accuracy* of the PABRFD and BFV-BRFD schemes are lower than those of the BRFD scheme without cryptographic mechanism. Integrating the encryption mechanism into BRFD in the PABRFD and BFV-BRFD schemes incurs the decline of the classification performance. In addition, in the PABRFD and BFV-BRFD schemes, the motion data is discretized to improve the computational efficiency of Bayesian classification probability. However, the discretized motion data also results in a decrease in the classification performance. It is worth mentioned that the classification performance of the PABRFD and BFV-BRFD schemes is still within an acceptable range. In addition, the classification performance of the PABRFD scheme is higher than that of the BFV-BRFD scheme, because the motion data needs to be converted into integers in the BFV-BRFD scheme. So, loss of the accuracy of the motion data also results in a decrease in the classification performance.

Table 1. Classification performance

scheme	*Accuracy*	*Precision*	*Recall*
BRFD	0.97	0.98	0.99
PABRFD	0.96	0.97	0.99
BFV-BRFD	0.93	0.94	0.98

Table 2. Time consumption comparison of homomorphic encryption

scheme	poly_modulus_degree	encryption (ms)	decrypt (ms)	addition (ms)	multiplication (ms)
BFV-BRFD	2048	287	18	11	80
	4096	1423	78	23	421
	8192	4810	241	71	1213
PABRFD	2048	314	16	7	65
	4096	1513	72	22	208
	8192	4818	233	70	841

Communication overhead

The computational time of the HE algorithm is highly related to the degree of polynomial modulus (poly_modulus_degree) and the modulus of ciphertext (coeff_modulus). The larger the poly_modulus_degree is, the security higher of the scheme is, but the ciphertext complex also increases, which will decrease the computational efficiency of the homomorphic operation. Therefore, in our experiments, we test the classification performance by using different polynomial modulus degrees:

2048, 4096, and 8192. As we can see in the Table 2 below, the performance of the PABRFD scheme is better than that of BFV-BRFD, especially in terms of multiplication operation time. As the poly_modulus_degree increases, the time consumption of the PABRFD scheme increases slowly.

In addition, to further evaluate and compare the performance of PABRFD, BRFD, and BFV-BRFD, we compare the computational performance of the three schemes in performing a Bayesian decision. As shown in Table 3, due to the introduction of the homomorphic encryption mechanism, the time consumption in running a Bayesian decision in the PABRFD and BFV-BRFD significantly increases. At the same time, as the poly_modulus_degree grows, the time consumption also increases exponentially and the result is similar in the PABRFD and BFV-BRFD. At the same time, the model *Accuracy* of the BFV-BRFD scheme is lower than that of BRFD. The BFV-BRFD scheme needs to convert floating-point numbers to integers in the Bayesian decision, which loses the *Accuracy* of the motion data. So, it is difficult for the BFV-BRFD scheme to achieve the accuracy of BRFD.

Table 3. Time consumption comparison in a Bayesian decision

Scheme	poly_modulus_degree	Average Time(ms)	Whether to achieve the *Accuracy* of BRFD
BRFD	/	3.8	/
BFV-BRFD	4096	614	no
	8192	1411	no
PABRFD	4096	519	yes
	8192	1267	yes

Like the work [16], we also introduce the following metrics such as *IPSD*, *NFIP*, and *NSIP* to evaluate the PABRFD scheme: *IPSD* indicates an interest packet satisfaction delay. That is to say, it is a time interval from a node sends an interest packet to the node receives a data packet related to the sent interest packet. *IPSD* is an important indicator to evaluate the NDN performance. *NFIP* notes a total number of forwarding interest packets in a simulation period. And *NSIP* indicates a total number of satisfied interest packets in the simulation period. That is to say, all of the nodes that sent these interest packets got the intended content.

Firstly, we compare the *IPSD* of PABRFD, BFV-BRFD, and BRFD. In NDN-VANET, high-speed movement of vehicles results in intermittent wireless links between adjacent nodes. So, the vehicle nodes should forward the received interest packets with minimal delay. As shown in Figure 5, after integrating the homomorphic encryption mechanism, the *IPSD* of the PABRFD scheme and the BFV-BRFD scheme is much higher than that of the BRFD scheme.

Secondly, we compare the *NFIP* of PABRFD, BFV-BRFD, and BRFD schemes. In the NDN-VANET network, *NFIP* represents the total number of Interests forwarded during the simulation, which is an important indicator for evaluating broadcast suppression. The lower the *NFIP*, it has the fewer redundant interest packets, and the better network performance. The *BRFD* scheme adopts a receiver-forwarding decision scheme based on Bayesian to suppress broadcast storms. As shown in Figure 6, the *NIFP* in the PABRFD is 5%-10% higher than that in the BFV-BRFD. Because the

classification accuracy of the BAPRFD scheme is higher than that of the BFV-BRFD scheme, the *NFIP* of the PABRFD scheme is closer to that of the BRFD scheme.

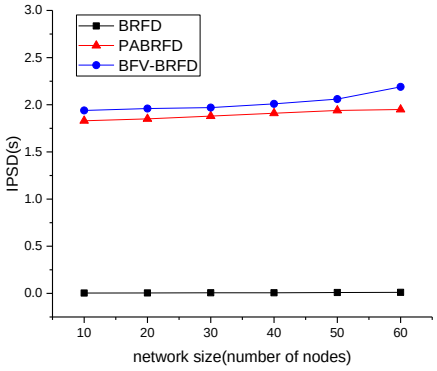


Fig. 5. IPSD

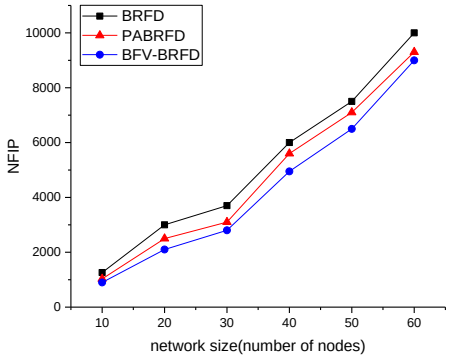


Fig. 6. NFIP

Finally, we analyze the *NSIP* of PABRFD, BFV-BRFD, and BRFD schemes during the simulation. *NSIP* represents the total number of Interest packets satisfied during the simulation, and *NSIP* also reflects the accuracy of the forwarding decision mechanism, which is an important evaluation index in NDN. As shown in Figure 7, the PABRFD and BFV-BRFD scheme reduces *NIPS*, mainly due to the decline of *NFIP*.

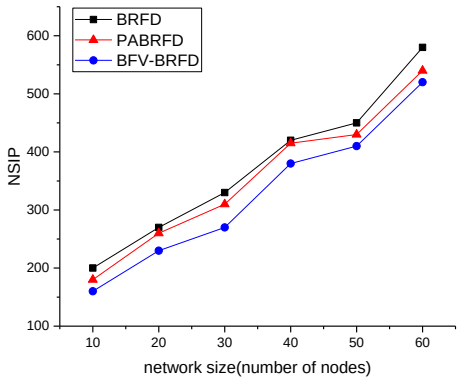


Fig. 7. NSIP

6. The conclusion and future work

In this paper, a Privacy-Aware intelligent forwarding solution PABRFD is proposed by integrating HE into BRFD. In PABRFD, a secure Bayesian classifier based on HE is used to protect the security and privacy of vehicle nodes. In PABRFD, First of all, the vehicle motion data in NDN-VANET is discretized to avoid the complex Gaussian calculation in the BRFD scheme, and the calculation efficiency of Bayesian classification probability is improved, which also provides a convenience for a secure and efficient naive Bayesian classifier. Secondly, we propose a HE-based secure network status exchange and storage mechanism suitable for the NDN-VANET environment, which can protect the private information of vehicles in NDN-VANET. Finally, we informally analysis security attributes that we hope to achieve. And we also implement the PABRFD and compare it's performance with related schemes BRFD and BFV-BRFD. Our experimental results show that our novel solution can satisfy our expected requirements.

However, the PABRFD scheme has limitations, and HE increases the time consumption of the PABRFD. Therefore, finding a solution with lower time consumption and better performance is one of our main research works in the future. At the same time, the security and privacy issues in the Bayesian training phase are also one of our future research works.

Acknowledgments: This work is supported by NSFC No. 61461027; Gansu province science and technology plan project under grant No. 20JR5RA467; Innovation Promotion Education Fund of Ministry of Education No. 2018A05003; Graduate Fine-designed Course of Lanzhou University of Technology.

References

1. M. S. Sheikh and J. Liang. A comprehensive survey on VANET security services in traffic management system. Wireless Communications and Mobile Computing. vol. 2019, (2019).
2. M. A. Hossain, R. M. Noor, K.-L. A. Yau, S. R. Azzuhri, M. R. Z'aba, and I. Ahmedy. Comprehensive survey of machine learning approaches in cognitive radio-based vehicular ad hoc networks. IEEE Access. vol. 8, 78054-78108. (2020).
3. L. Zhao, Y. Li, C. Meng, C. Gong, and X. Tang. A SVM based routing scheme in VANETs. in 2016 16th International Symposium on Communications and Information Technologies (ISCIT), IEEE, 380-383. (2016).
4. K. Roscher, T. Nitsche, and R. Knorr. Know thy neighbor-a data-driven approach to neighborhood estimation in vanets. in 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), IEEE, 1-5. (2017).
5. H. Bangui, M. Ge, and B. Buhnova. A hybrid machine learning model for intrusion detection in VANET. Computing. vol. 104, No. 3, 503-531. (2022).
6. R. Bibi, Y. Saeed, A. Zeb, T. M. Ghazal, T. Rahman, R. A. Said, et al. Edge AI-based automated detection and classification of road anomalies in VANET using deep learning. Computational intelligence and neuroscience. vol. 2021, (2021).
7. S. K. Singh, J. Cha, T. W. Kim, and J. H. Park. Machine learning based distributed big data analysis framework for next generation web in IoT. Computer Science and Information Systems. vol. 18, No. 2, 597-618. (2021).
8. C. Zhang, X. Zhao, M. Cai, D. Wang, and L. Cao. A new model for predicting the attributes of suspects. Computer Science and Information Systems. vol. 17, No. 3, 705-715. (2020).
9. N. Y. Yen, H.-Y. Jeong, K. Madani, and F. I. Massetto. Guest editorial: Emerging services in the next-generation web: Human meets artificial intelligence. Computer Science and Information Systems. vol. 18, No. 2, 1-6. (2021).
10. L. Liang, H. Ye, and G. Y. Li. Toward intelligent vehicular networks: A machine learning framework. IEEE Internet of Things Journal. vol. 6, No. 1, 124-135. (2018).
11. S. Ftaimi and T. Mazri. A comparative study of Machine learning algorithms for VANET networks. in Proceedings of the 3rd International Conference on Networking, Information Systems & Security, ACM, 1-8. (2020).
12. S. Khatri, H. Vachhani, S. Shah, J. Bhatia, M. Chaturvedi, S. Tanwar, et al. Machine learning models and techniques for VANET based traffic management: Implementation issues and challenges. Peer-to-Peer Networking and Applications. vol. 14, No. 3, 1778-1805. (2021).
13. T. Liu, S. Shi, and X. Gu. Naive Bayes Classifier Based Driving Habit Prediction Scheme for VANET Stable Clustering. Mobile Networks and Applications. vol. 25, No. 5, 1708-1714. (2020).
14. A. Mehmood, A. Khanan, A. H. H. Mohamed, S. Mahfooz, H. Song, and S. Abdullah. ANTSC: An intelligent Naïve Bayesian probabilistic estimation practice for traffic flow to form stable clustering in VANET. IEEE Access. vol. 6, 4452-4461. (2017).
15. S. A. Karuppusamy, S. Umasangeetha, and N. Nandhagopal. Study on Intelligent Naive Bayesian Probabilistic Estimation Practice for Traffic Flow to Form Stable Clustering In VANET. International Journal Of Information and Computing Science, ISSN. vol. 6, No. 2, (2019).
16. X. Guo, Y. Chen, L. Cao, D. Zhang, and Y. Jiang. A receiver-forwarding decision scheme based on Bayesian for NDN-VANET. China Communications. vol. 17, No. 8, 106-120. (2020).
17. M.-Y. Chen, J. d. J. Rubio, and A. K. Sangaiah. Guest editorial-Pattern recognition, optimization, neural computing and applications in smart city. Computer Science and Information Systems. vol. 18, No. 4, 3-4. (2021).

18. T. Zuowen and Z. Lianfu. A review of research on privacy protection in machine learning(In Chinese). Journal of Software. vol. 31, No. 7, 2127-2156. (2020).
19. S. Alfeld, X. Zhu, and P. Barford. Data poisoning attacks against autoregressive models. in Proceedings of the AAAI Conference on Artificial Intelligence, AAAI. (2016).
20. I. M. Ahmed and M. Y. Kashmoola. Threats on Machine Learning Technique by Data Poisoning Attack: A Survey. in International Conference on Advances in Cyber Security, Springer, 586-600. (2021).
21. Z. Zhang, C. Yan, and B. A. Malin. Membership inference attacks against synthetic health data. Journal of biomedical informatics. vol. 125, No. 6, 63-81. (2022).
22. C. C. Wei Lifei, Zhang Lei, and Li Simeng. Security issues and privacy protection of machine learning. Computer Research and Development. vol. 57, No. 10, 126-148. (2020).
23. M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning. in 2019 ieee symposium on security and privacy, IEEE, 739-753. (2019).
24. K. Yoshida, T. Kubota, M. Shiozaki, and T. Fujino. Model-extraction attack against FPGA-DNN accelerator utilizing correlation electromagnetic analysis. in 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), IEEE, 318-318. (2019).
25. Y. Lindell and B. Pinkas. Privacy preserving data mining. in Annual International Cryptology Conference, Springer, 36-54. (2000).
26. A. C.-C. Yao. How to generate and exchange secrets. in 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), IEEE, 162-167. (1986).
27. T. Zhang and Q. Zhu. Distributed privacy-preserving collaborative intrusion detection systems for VANETs. IEEE Transactions on Signal and Information Processing over Networks. vol. 4, No. 1, 148-161. (2018).
28. G. Raja, S. Anbalagan, G. Vijayaraghavan, S. Theerthagiri, S. V. Suryanarayan, and X.-W. Wu. SP-CIDS: Secure and Private Collaborative IDS for VANETs. IEEE Transactions on Intelligent Transportation Systems. vol. 22, No. 7, 4385-4393. (2020).
29. X. Li, H. Zhang, Y. Ren, S. Ma, B. Luo, J. Weng, et al. PAPU: Pseudonym Swap With Provable Unlinkability Based on Differential Privacy in VANETs. IEEE Internet of Things Journal. vol. 7, No. 12, 11789-11802. (2020).
30. X. Chen, T. Zhang, S. Shen, T. Zhu, and P. Xiong. An optimized differential privacy scheme with reinforcement learning in VANET. Computers & Security. vol. 110, No. 25, 1025-1056. (2021).
31. G. Raja, S. Anbalagan, G. Vijayaraghavan, P. Dhanasekaran, Y. D. Al-Otaibi, and A. K. Bashir. Energy-efficient end-to-end security for software-defined vehicular networks. IEEE Transactions on Industrial Informatics. vol. 17, No. 8, 5730-5737. (2020).
32. H. Kaur, N. Kumar, and S. Batra. ClaMPP: A cloud-based multi-party privacy preserving classification scheme for distributed applications. The Journal of Supercomputing. vol. 75, No. 6, 3046-3075. (2019).
33. T. Li, L. Lin, and S. Gong. AutoMPC: Efficient multi-party computation for secure and privacy-preserving cooperative control of connected autonomous vehicles. in SafeAI@ AAAI, CEUR Workshop Proceedings, 1-4. (2019).
34. Y. Wu, X. Wang, W. Susilo, G. Yang, Z. L. Jiang, S.-M. Yiu, et al. Generic server-aided secure multi-party computation in cloud computing. Computer Standards & Interfaces. vol. 79, No. 21, 112-130. (2022).
35. S. Sayyad. Privacy Preserving Deep Learning Using Secure Multiparty Computation. in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 139-142. (2020).
36. X. Ma, F. Zhang, X. Chen, and J. Shen. Privacy preserving multi-party computation delegation for deep learning in cloud computing. Information Sciences. vol. 459, No. 2, 103-116. (2018).

37. H. Li, J. Chen, L. Wang, Q. Pei, and H. Yue. Privacy-preserving Data Aggregation for Big Data in Financial Institutions. in IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 978-983. (2020).
38. J. Zhou, S. Chen, K.-K. R. Choo, Z. Cao, and X. Dong. EPNS: Efficient Privacy Preserving Intelligent Traffic Navigation from Multiparty Delegated Computation in Cloud-Assisted VANETs. IEEE Transactions on Mobile Computing. vol. 12, No. 3, 11-25. (2021).
39. D. Ulybyshev, A. O. Alsalem, B. Bhargava, S. Savvides, G. Mani, and L. B. Othmane. Secure data communication in autonomous v2x systems. in 2018 IEEE International Congress on Internet of Things (ICIOT), IEEE, 156-163. (2018).
40. Q. Kong, R. Lu, M. Ma, and H. Bao. A privacy-preserving sensory data sharing scheme in Internet of Vehicles. Future Generation Computer Systems. vol. 92, No. 2, 644-655. (2019).
41. H. Cheng, M. Shojafar, M. Alazab, R. Tafazolli, and Y. Liu. PPVF: privacy-preserving protocol for vehicle feedback in cloud-assisted VANET. IEEE Transactions on Intelligent Transportation Systems. vol. 6, No. 12, 1-13. (2021).
42. N. Magaia, C. Borrego, P. R. Pereira, and M. Correia. ePRIVO: An enhanced privacy-preserving opportunistic routing protocol for vehicular delay-tolerant networks. IEEE Transactions on Vehicular Technology. vol. 67, No. 11, 11154-11168. (2018).
43. A. Alamer, Y. Deng, and X. Lin. A privacy-preserving and truthful tendering framework for vehicle cloud computing. in 2017 IEEE International Conference on Communications (ICC), IEEE, 1-7. (2017).
44. H. Sasaki and N. Kamiyama. Summary Cache of IoT Data Using ICN. in 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 707-710. (2021).
45. C. Gentry. Fully homomorphic encryption using ideal lattices. in Proceedings of the forty-first annual ACM symposium on Theory of computing, ACM, 169-178. (2009).
46. Kimlaine. Microsoft SEAL. Available: https://github.com/microsoft/SEAL
47. D. Lowd and P. Domingos. Naive Bayes models for probability estimation. in Proceedings of the 22nd international conference on Machine learning, ACM, 529-536. (2005).
48. K. M. Leung. Naive bayesian classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering. vol. 2007, 123-156. (2007).
49. A. Acar, H. Aksu, A. S. Uluagac, and M. Conti. A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys (CSUR). vol. 51, No. 4, 1-35. (2018).
50. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory. vol. 31, No. 4, 469-472. (1985).
51. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. in International conference on the theory and applications of cryptographic techniques, Springer, 223-238. (1999).
52. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. vol. 21, No. 2, 120-126. (1978).
53. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. in Theory of cryptography conference, Springer, 325-341. (2005).
54. J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. in International Conference on the Theory and Application of Cryptology and Information Security, Springer, 409-437. (2017).
55. Y. Yasumura, Y. Ishimaki, and H. Yamana. Secure Naïve Bayes classification protocol over encrypted data using fully homomorphic encryption. in Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services, ACM, 45-54. (2019).
56. M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO–simulation of urban mobility: an overview. in Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation, ThinkMind, 23-28. (2011).

57. S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang. A new version of the NDN simulator for NS-3. Univ. of California. 1-8. (2015).

Xian Guo is an associate professor of School of Computer and Communication, Lanzhou University of Technology. He is a visiting scholar at University of Memphis. He received MS and PhD in Lanzhou University of Technology, China, in 2008 and 2011, respectively, and BS in Nothwest Normal University. His current research interests include network and information security, cryptographic, and blockchain. E-mail: iamxg@ 163.com.

Baobao Wang is currently a master student at Computer and Communication School of Lanzhou University of Technology. He received his Bachelor degree from North China Institute of Science and Technology, China, in 2019, and started his master studying in 2019. His research interests are machine learning, Information-Centric Networking etc., Secure Multiparty Computation.

Yongbo Jiang is a lecturer of School of Computer and Communication, Lanzhou University of Technology. He received MS and PhD in Xidian University, China, in 2008 and 2013, respectively. His current research interests include network and information security, and Information-Centric Networking etc.

Di Zhang is an associate professor of School of Computer and Communication, Lanzhou University of Technology. He received MS and PhD in Communication University of China, China, in 2013 and 2016, respectively. His current research interests include network and information security, and blockchain etc.

Laicheng Cao is a professor of School of Computer and Communication, Lanzhou University of Technology. He received MS in Lanzhou University, China, in 2004. His current research interests include network and information security, and cryptography etc.

Received: February 10, 2022; Accepted: July 02, 2022.

Computer Science and Information Systems 20(1):25–50 https://doi.org/10.2298/CSIS211230045K

Cloud-Based ERP Construction Process Framework in the Customer's Perspective

Hyeong-Soo Kim[1], Deok-Soo Oh[1], and Seung-Hee Kim[2]

[1] Master's Student, Department of IT Convergence Software Engineering, Korea University of Technology and Education, 1600 Chungjeol-ro, Dongnam-gu, Cheonan-si, Chungcheongnam-do (31253) Republic of Korea
{kawa95, ts0070}@ koreatech.ac.kr
[2] Faculty, Department of IT Convergence Software Engineering, Korea University of Technology and Education, 1600 Chungjeol-ro, Dongnam-gu, Cheonan-si, Chungcheongnam-do (31253) Republic of Korea
sh.kim@koreatech.ac.kr

Abstract. Process frameworks for the implementation of cloud enterprise resource planning (ERP) were derived and each process was examined through detailed comparisons with on-premise ERP construction processes, using process engineering characteristics. The process frameworks for implementing cloud ERP are classified into infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), content-as-a-service (CaaS), and software-as-a-service (SaaS), depending on the construction type, and are defined based on 6 derived processes, 21 activities, and numerous specific tasks. The process engineering characteristics of the final proposed process framework were further analyzed and examined in comparison to on-premise ERP construction processes with respect to differences and similarities. This study provides a theoretical foundation of standardized research on cloud ERP construction methods. As a practical guideline for stakeholders, it can be used in practice as a process tailoring tool, providing information on specific activities and tasks for each construction phase, contributing to the construction and spread of reliable cloud-based ERP systems from the customer's perspective.

Keywords: Cloud ERP, ERP comparison, SaaS ERP, CaaS, PaaS, IaaS

1. Introduction

Enterprise resource planning (ERP) is a system for the integrated management of business processes of a company and refers to a commercial software package [1] that supports management in an integrated manner, in real-time. ERP usually integrates main business activities and increases the value of the business process operation performed by all business functions [2,3]. The integration of business processes through ERP ensures information integrity, and falls into the software category of corporate organizational data management [4]. An ERP system can facilitate information flow by automating activities between all business functions within the organization by combining internal and external management across the organization [5].

ERP software can be installed locally on computer hardware built in-house by the company and on user personal computers, depending on the construction type [6]. Regardless of the installation, most ERP software are classified into on-premise ERP and cloud ERP. The former refers to a traditional ERP system construction method where the maintenance of servers or software, such as manual upgrade and update, are performed by the owner company, and the latter refers to a system hosted by the cloud vendor that provides all services [4]. In recent years, most experts identify the cloud ERP as being the future of business technology [7]. In a 2021 ERP report [8], 53.1 % selected cloud ERP, and according to Forbes [6], cloud computing spending has grown at 4.5 times the rate of IT spending since 2009. In the 2018 ERP report of Panorama Consulting [9], 85 % of ERP systems adopted are SaaS or cloud-based ERP systems. In the "Best ERP Software Vendor Companies Comparison 2021" by SelectHub [10], the top 10 ERP leaders among 125 products are Oracle JD Edwards EnterpriseOne, SAP business ByDesign (ByD), sageX3, SYSPRO, Microsoft Dynamics 365 ERP, Infor Syteline, Oracle Suite, EPICOR Kinetic, and IFS Applications from top to bottom in ranking.

Cloud computing is typically divided into the following models according to the construction type. The model is classified as IaaS when only the computer infrastructure is leased, as PaaS when the vendor hosts all infrastructure and programming tools for the implementation of web-based application programs, and as SaaS when the costs are paid for the software hosted by the vendor [5]. In terms of deployment models, three-quarters of the organizations that chose cloud ERP are using the SaaS model [8]. Cloud computing ERP is a hosting service provided through the Internet [5], whereas, the ERP system in the SaaS model resides in the cloud and provides computing functions to run the ERP system.

SaaS Model refers to the application hosted as a service, and the user can access the application through Web-based software in the browser without installing or maintaining any software [26]. Therefore, cloud-based ERP construction and SaaS ERP have different ranges of meaning. Saas ERP is considered a service within a SaaS model. By contrast, cloud-based ERP refers to a cloud computing-based ERP service, among which IaaS, PaaS, and SaaS service models are included. Herein, this approach is referred to as cloud ERP or cloud-based ERP.

Drawbacks to cloud ERP have been pointed out, such as difficulties in changing and conducting ongoing training for the processes [8], security [8,12,13], confidentiality [12], network reliability and integration problems [12], an increased risk of data loss [8], and ambiguity in the performance of cloud services and data processing [13].

On-premise ERP systems are hosted by organizations who handle their own infrastructure, operating systems and software, and database services and hardware [27]. However, such an approach makes it difficult to access information remotely [26], and the system data are not in real-time, which indicates an inadequate reliability [28]. In addition, the organization must install and operate all hardware and software, which requires intricate work [29]. However, cloud ERP systems have a variety of advantages, such as a fast construction, low initial cost, rapid upgrades and updates, the ability to handle changes and growth, and the ability to back up and restore data [30].

In general, on-premise ERP and cloud ERP have a commonality in that they both require tasks and tests for integration with other linked systems [8], but there are very clear differences in the construction methods and maintenance of fundamental

technology, TCO calculation, implementation / distribution / update / maintenance of solution applications with respect to task changes, scalability, security, performance, and technical support. Despite these differences, most cloud ERP providers use conventional on-premise ERP construction methods to carry out vendor-oriented cloud ERP construction projects. Vendor-specific construction processes, pricing policies, billing methods, solution flexibility, and support programs increase the difficulty for the customer who wants to select and implement the optimal cloud ERP solution through a strategic approach [14]. Therefore, the development of a standardized cloud ERP construction process framework from the customer's perspective is required for application to all cloud ERP selection methods, while increasing the reliability and scalability to overcome the limitations of on-premise ERP systems, such as system integration issues within an organization and the cost constraints [15, 16, 17].

In this study, a cloud-based ERP construction process framework from the customer's perspective is proposed by gathering and classifying the construction processes used by each commercial cloud ERP vendor, currently having a high market share. Furthermore, process engineering characteristics of the proposed model are examined through comparisons with those of the on-premise ERP. The results of this study can be used as basic data for the development of a standardized cloud ERP construction methodology to provide guidelines at a practical level for customer's perspective cloud ERP construction.

2. Related Work

2.1. Comparison Between Cloud-based ERP and Traditional On-Premise ERP

In reviewing previous studies related to ERP, the focus is on comparing traditional on-premise ERP and cloud ERP, which is a new method of implementing ERP. Table 1 shows the results. However, despite a variety of comparative studies on cost, usability, maintenance, scalability, implementation, security, mobility, and quality, studies on specific comparisons or methods of construction processes are lacking. Table 1 lists comparative studies conducted on cloud ERP and on-premise ERP. In the comparison results (which include the factors, criteria, and high and low grades), this paper cites the results of studies by various researchers without modification, or comparison factors found in the papers comparing cloud ERP and on-premise ERP, and the main content is presented in the comparison results.

2.2. Spiral Model of software development

The primary functions of a software process model are to determine the order of the stages involved in software development and evolution and to establish the transition criteria for progressing from one stage to the next [70]. A spiral model is one type of software development process model, and as its main feature, it adopts a risk-oriented

approach to the software processes. As shown in Fig. 1, the spiral model expands a system by repeatedly cycling through 4 phases. In Phase 1, the objectives, methods, and constraints are determined. In Phase 2, the risk factors are analyzed and resolved. In Phase 3, the software is developed and evaluated. Finally, in Phase 4, plans for the next phase are generated. The radial dimension shown in Fig. 1 represents the cumulative cost incurred in accomplishing the steps to date; in addition, the angular dimension represents the progress made in completing each cycle of the spiral [70]. The main advantages of this model are an improved software quality and the flexibility to respond to changes owing to the nonlinear and iterative nature of the development. However, when systems are developed incrementally, it can lead to high costs and failure if each cycle is not managed well or if a risk analysis is not properly conducted. As such, this model is suitable for projects in which problems with the technology and performance are anticipated.

Table 1. Comparison between cloud-based ERP and traditional on-premise ERP through preliminary studies

Factor	Criteria	Cloud ERP	On-Premise ERP
1. Cost	Upfront investment [12],[13],[14],[15],[16],[17],[18], [19],[20],[24],[36],[37],[38],[39], [40],[41],[42],[43],[44],[45]	Not High	High
	License [12], [13], [14], [17], [32],[36],[46],[47]	Low, handled by the provider, services in pay-per-use mode	High, user license required
	Energy [13]	Low	High
	Maintenance [12], [13], [14], [15], [16], [17], [21], [22], [32], [36], [44], [46], [48], [49], [50], [51], [52], [53]	Low	High
	Server [12], [14]	Low, availability	High
	Configuration [13]	Low	High
	Reduction in IT staff [14]	Low	High
2. Usability	Testability [13]	High	Low
	Upfront validation [13]	Easy	Difficult
3. Maintenance	Training [12]	required	Not required
	Upgrading and debugging [12], [15], [16], [37], [40], [42], [45], [48], [53],[54], [55]	Easy, upgrading can be done without affecting the services	Difficult
	Switching provider [12]	Easy	Not possible
	Target scope [12]	Focus shift to main competencies	Overall
	Data and environmental standards [12], [15], [21], [22]	Not ensured	Easy
	Availability & reliability [12], [14]	System and data recovery possible	System and disaster recovery are difficult.
	Controlling single point operation [14]	Easy	Difficult
	Enhancement [12], [22], [37], [40], [42],[45],[48],[53],[54],[55]	Moderately Easy	Difficult

	Compliance [12]	Moderately difficult	Easy
	Resource share and assignment [12], [14], [16], [38], [41], [42], [47], [50],[52],[56],[57],[58], [59] [60], [61],[62]	Easy, improves Agility	Very difficult
	Excellent dedicated staff [5],[32]	Required	Not required
4. Scalability	Extended services [12],[13],[14],[22],[36]	High	Low
	Integrated applications [12], [15],[16],[18],[21],[36]	Difficult	Possible
	Reports and analyses [12]	All the information can be grouped together easily and reports are generated in the required format	
	Data grouping	Moderately easy	Moderately difficult
5. Implementation	Time [13], [23], [36], [38], [41], [42], [43], [45], [47], [48], [49], [55], [62], [63], [64]	Very little	Long, 6 to 12 months or more
	Change [13],[36]	rapid	long
	Location [12]	Only client machines installed at the customer site	on the company premises
	Requirements [12]	Does not support complete back office requirements	High, rich functionality
	Customization [12],[15],[21],[22], [23],[37],[39],[40],[41],[48],[49],[51],[53],[54],[56],[64],[65],[66]	Provider based approach, integration difficult	Complete customization and integration supported
	Type [14]	SME company	Large enterprise
	Migration [12], [16]	Easy	Moderately complex
6. Security	Control privilege [13], [16], [20], [22], [23]	Low, owned by product owner	High
	Control safety [13],[16],[20],[21],[22],[23],[45], [47],[67],[68]	Low	High
	Attacks targeting shared tenancy environment [14]	High	Low
	Security and confidentiality [12],[36],[45],[49],[65]	Very difficult	Very high
	Web security [8]	High	Low
7. Mobility	System flexibility [12],[15],[16],[22]	High	Low
	Accessibility [12],[36],[41], [43],[56],[59],[67],[69]	Low	High
	Efficiency [1], [4]	Improved	Low
	Decision making process [12]	Improving, easier	Low
8. Quality	Network performance [12],[32]	Completely dependent on	Not dependent on
	Performance optimization [15], [21],[22]	Low	High
	Accuracy [12]	Improved	
	Data integrity [14],[47],[67],[68]	High	Moderately low

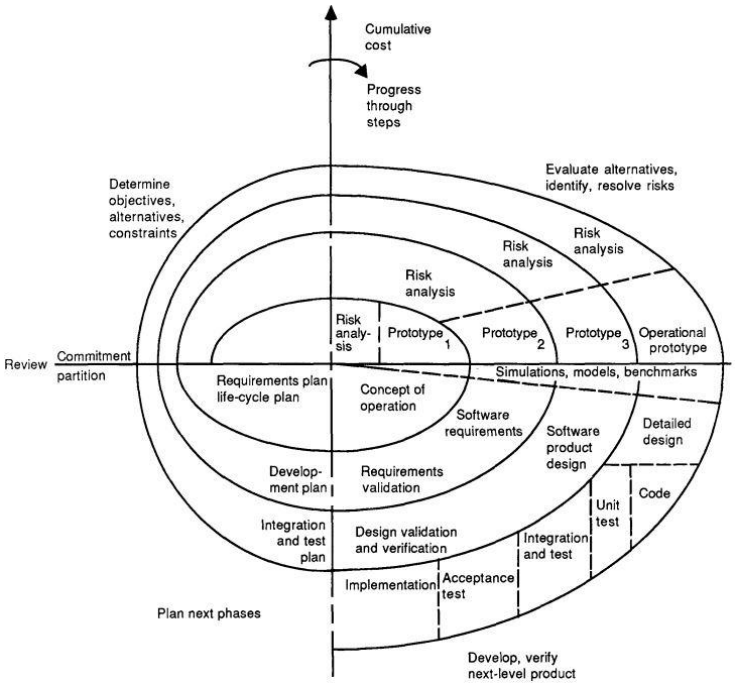


Fig. 1. Spiral model of the software process [70]

3. Research Procedures and Methods

3.1. Research Procedures

For this study, the construction processes used by each commercial cloud ERP vendor as well as the actual work breakdown structure (WBS) and schedule data used when constructing the cloud ERP systems were collected. Based on this, the processes, activities, and tasks for ERP construction were derived in three hierarchical levels through the KJ technique. Furthermore, the on-premise ERP construction process framework, derived as a result of preliminary research conducted separately, was combined with the study results of other researchers related to cloud ERP processes, to develop a customer-based process framework, which was further classified into IaaS, PaaS, CaaS, and SaaS according to the construction type of cloud ERP. To examine the differences, commonalities, and implications, the process engineering characteristics of the finally confirmed cloud-based ERP construction process framework from the perspective of the customer were analyzed and compared with those of the on-premise

ERP construction process framework. Fig. 2 shows the procedures and methods used in the study described above.

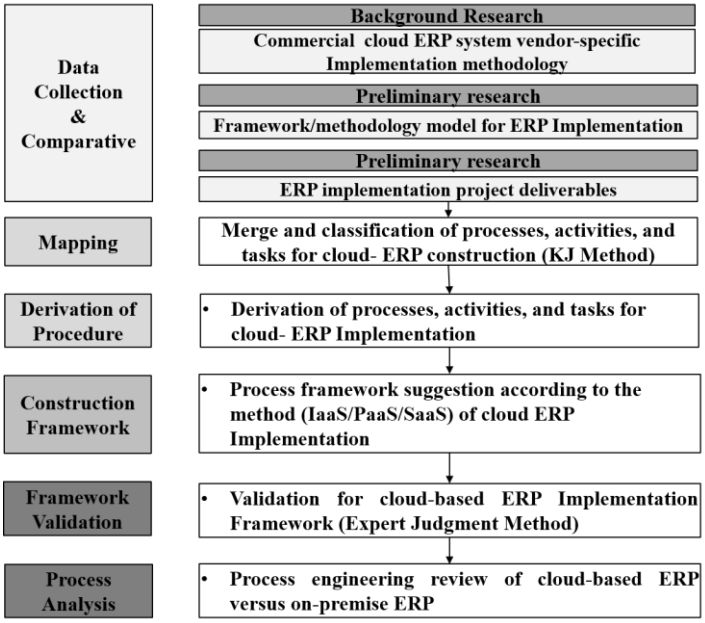


Fig. 2. Research process and Method

3.2. Research Method

KJ Method

The KJ technique, named after Kawakita Jiro, is used to classify and group various items based on their similarity or relevance [33]. In this study, the technique was used to group similar or identical process items and attach representative process names to the grouped results, based on the results of various studies on cloud ERP construction methodologies, processes, and frameworks, as well as vendor-specific methodologies, WBS, and schedules, which are practical types of data. Through this technique, fragmented information can be organized into groups with a high logical cohesion.

Expert Judgment

Expert judgment is defined as judgment provided based upon expertise in an application area, knowledge area, discipline, industry, etc., as appropriate for the activity being

performed, such expertise may be provided by any group or person with specialized education, knowledge, skill, experience, or training [34]. Expert judgement has always played a large role in science and engineering. Increasingly, expert judgement is recognized as just another type of scientific data, and methods are developed for treating it as such [35]. Such judgement was used as a method to validate the results of the cloud ERP construction process framework derived through this study, focusing on experts with ample experience in cloud ERP construction.

Each expert presented opinions on the IaaS, PaaS, CaaS, and SaaS ERP models, referring to a table of the processes, activities, and tasks for constructing the derived framework and the cloud ERP. The experts wrote their opinions in advisory documents, which included questions regarding each phase of the framework, such as, “Are the cloud ERP construction process (commercial vendor phase) and cloud ERP model division appropriate?” In addition, “Are all of the major activities for each process included?” Finally, “Are any major tasks omitted for each process activity?” This study sought validation opinions regarding the study results from experts working at YoungLimWon SoftLab and incorporated these opinions into the study. YoungLimWon SoftLab is a Korean ERP specialist company that launched its own ERP brand, called K-System, in 1997 and currently services over 25,000 customers in Korea and abroad [11]. The experts who provided the expert judgment for this study were the company’s executive vice presidents, who have over 20 years of actual experience in carrying out cloud-based ERP construction projects.

4. Suggestion of a Cloud-based ERP Construction Process Framework in the Customer’s Perspective

4.1. Data Collection

Considering the market share and utilization size, data and processes were collected for vendor-specific construction methods by selecting the commercial ERP software of eight companies: SAP S4HANA Cloud Process, Oracle Cloud, Microsoft Dynamics 365, Acumatica Cloud, Intuit, YoungLimWon SaaS Cloud, Infor Cloud, and Epicor. As shown in Table 2, processes are classified into a minimum of five phases up to a maximum of eight phases. Vendors are observed to follow very different processes. Furthermore, the number and scope of activities performed in each procedure for each process are very different. Detailed construction processes are provided with six phases and 86 tasks / activities for the SAP S4HANA Cloud ERP construction, whereas activities for each process are not clearly and explicitly defined for Microsoft Dynamics 365, Acumatica Cloud ERP, Intuit, and YoungLimWon Cloud ERP.

In examining the intrinsic characteristics of the processes, it was found that the review activities prior to construction were highly sub-divided in the case of Acumatica Cloud ERP, and the live phase was excluded from the processes in the case of Infor Cloud ERP. On the other hand, the processes of Epicor Cloud ERP consisted of prepare and plan, unlike SAP Cloud ERP. Considering the case of cloud-based ERP

construction projects that transform on-premise-based information services to cloud-based information services, the research results of Deok-Soo Oh et al. [24] were reviewed and reflected upon in addition to the activity and task items of each process.

Table 2. Processes and activities by Cloud ERP Vendor

Vendor	Cloud ERP S/W	Process	Activity
SAP	S4HANA Cloud ERP	1. Discover 2. Prepare 3. Explore 4. Realize 5. Deploy 6. Run	71 activities including Discovery Assessment
Oracle	Oracle Cloud ERP	1. Plan 2. Implement 3. Verify 4. Prepare 5. Deliver	21 activities including Project Definition
Microsoft	Dynamics 365 ERP	1. Preparation and planning 2. Procedure review 3. Data preparation 4. Testing and training 5. Rollout and evaluation	15 activities including Project Team Building
Acumatica	Acumatica Cloud ERP	1. Discovery 2. Plan & Monitor 3. Analyze 4. Build 5. Stabilize 6. Deploy 7. Post Go live	12 activities including Project Strategy Development
Intuit	Intuit ERP	1. Research and planning 2. Product-company fit 3. Budgeting 4. Data migration 5. Testing 6. Training 7. Go-live 8. Post-implementation considerations	11 activities including Requirement Review
YoungLimWon	SaaS Cloud ERP	1. Discover 2. Prepare 3. Provisioning 4. Consulting 5. Live	15 activities including Verification of Service Goods
Infor	Infor Cloud	1. Inception 2. Elaboration 3. Construction 4. Transition 5. Optimize	16 activities including SW Supply
Epicor	Epicor Cloud	1. Prepare 2. Plan 3. Design 4. Validate 5. Deploy	17 activities including Scope Definition

Traditional ERP construction consists of the following stages: pre-implementation, project planning, an as-is study, a to-be design, a gap analysis and customization, system configuration, conference room pilot, user training, user acceptance testing, installation and set-up, data migration, go-live, and post-implementation [25].

However, major ERP vendors devise and use their own methodologies for properly implementing their products. Typical examples include methodologies such as SAP's Accelerated SAP (ASAP), Oracle's Unified Method (OUM), and Microsoft Dynamics Sure Step. These methodologies are compared in Table 3, which was created by

reorganizing the studies in [13],[25],[31],[71]. In particular, OUM, which is a follow-up to the Application Implementation Methodology (AIM), reclassified 6 phases and 12 processes of AIM into 5 phases and 35 processes.

Table 3. Vendor specific methodology (ASAP, OUM, Sure Step)

SAP	Preparation	Business Blueprint	Realization	Final preparation	Go Live & Support
	•Initial Project Planning •Project Procedures • Training • Kick Off •Technical Requirements •Quality Check	•Project Management • Organizational Change Management • Training •Develop System Environment •Organizational Structure Definition •Business Process Analysis •Business Process Definition •Quality Check	•Project Management • Organizational Change Management • Training •Baseline Configuration & Confirmation •System Management •Final Configuration & Confirmation •Develop Programs, Interfaces etc. •Final Integration Test •Quality Check	•Project Management •Training •System Management •Detailed Project Planning • Cutover •Quality Check	•Migration to Production Environment •Production Support •Monitoring •Performance Optimization
OUM	Project Design	Configure	Validate	Transition	Realization
	•Plan Project •Conduct Kickoff Meeting •Schedule Workshops •Conduct Functional Design Workshops •Conduct Technical Design Workshops •Conduct Design Review •Develop Security and Validation Strategies and Plans •Conduct Implementation Checkpoint •Project Management	• Setup Applications •Validate Configuration •Load & Validate Data •Build & Validate Integrations •Apply & Validate Extensions and Extensible items •Implements Security •Prepare Cutover strategy •Conduct Implementation Checkpoint •Project Management	•Update Setups •Prepare Validation Scripts •Load & Validate Data •Conduct End-to-End Review •Prepare for Training •Conduct Train-the-Trainer Workshops •Conduct Implementation Checkpoint •Project Management	•Migrate Configuration Production •Migrate Integrations Extensions Production •Load, Reconcile & Validate Data in Production •Conduct Final Validation Review •Verify Production & Operational Readiness •Begin Production Use •Conduct Implementation Checkpoint •Project Management	•Manage toTransition to Steady state Operations &•Post Go-Live toSupport •Handoff to Customer Relationship Manager •Gain Acceptance •Close Project
Sure Step	Diagnostic	Analysis	Design	Development	Deployment	Operation
	•Cultivate Customer Relationship •Pre-Sales Support •Execute Decision Accelerators •Complete SOW	•Finalize Project Plan and Project Charter •Execute Functional Requirements Workshop •Execute Fit Gap Analysis •Develop Test Plan	•Conduct Core Team Training •Develop Functional Design Documents for Configurations (Fits) •Develop FDD's for Customizations (Gaps) •Demo CRP (For Rapid Project Type)	•Configure/Setup solution •Conduct Process Testing •Conduct Integration Testing •Manage Scope and Resolve Issues •Complete Solution Design Document	•Conduct Train the Trainer (TTT) Training •Conduct Use Training •Conduct User-Acceptance Testing •Perform Go-Live Readiness Activities •Ready Solution for Production Deployment	•System Go-Live •Post Production Support [Additional Phases] Optimization - Upgrade

4.2. Merging and Classification of Cloud-based ERP Construction Processes, Activities, and Tasks Using the KJ Technique

After combining all collected data, unique numbers were assigned to the cloud ERP methods, processes, and activities to derive standard processes independent of the vendor. The relevance index was evaluated by focusing on specific activities and tasks. In deriving processes, similar and same processes are classified as one group for each vendor in the primary KJ. In the secondary KJ, processes are grouped again by determining the uniqueness of each process when the task scope is broad or the agent performing the role is unclear. Based on this, a total of six processes were derived through mapping. In the tertiary KJ, tasks derived from a previous study [24] are added onto the processes for the construction of on-premise ERP.

The tasks were all reassigned to vendor-specific activities according to the derived processes. Unique numbers were used instead of activity or task names to efficiently derive tasks for each process. Because the finalized tasks were assigned based on this process, the characteristics, the scopes, domains, and agents of the tasks were very broad. Therefore, tasks were reclassified and grouped to clarify the scope and domain of each, and a suitable name was given to each group, thus deriving a major activity for each process.

4.3. Cloud-based ERP Construction Processes

A refining process was performed to finalize the results classified in the form of process, activity, and task in the final cloud ERP construction process. Through this procedure, the names of processes, activities, and task terms were modified to clarify the meaning. Furthermore, when it became necessary to classify the processes in terms of users and vendors, terms were redefined based on the role, even for the same activity name. The procedural tasks for deriving final construction results, such as understanding of customer's business and project communication, were deleted considering the diversity of the methods.

4.4. Development and validation of Cloud-based ERP Construction Process Framework in the Customer Perspective

The derived processes, activities, and tasks were classified into SaaS, CaaS, PaaS, and IaaS according to the type of cloud ERP construction. Typically, cloud ERP is purchased as SaaS, PaaS, or IaaS based on the construction type. However, four phases, including CaaS, were provided so that vendors and customers could all refer to them for solutions based on the application, and the roles and collaboration conditions could be clarified.

To practically validate the derived framework, review opinions were collected from two expert executives at YoungLimWon SoftLab, and who have many years of experience in constructing actual cloud-based ERP. The first expert has experienced the construction of IaaS- and SaaS-based ERP systems more than 20 times over the course

of a 5-year period. The other expert has over 7 years of experience in ERP construction and has applied SaaS ERP in more than 30 location. For validation, the experts reviewed the derived framework and all of its processes and answered 15 questions to confirm whether the construction procedure, model classifications, and the activities and tasks included in each process were appropriate or had any omissions. After confirming these issues, they provided their opinions. Table 4 provides an overview of the review opinions.

Table 4. Overview of expert opinions

Expert	Overview of expert judgements
Expert 1	- The derived cloud ERP construction process is thought to be suitable for medium-sized companies or larger customers whose business processes are standardized into an Organization Process Assert (OPA) format and whose company possesses adequate human resources. - The process will be difficult to apply to customers who are relatively small businesses and do not have clear process standards and who rely on best-practice processes. - A cloud-based ERP construction model selection activity/task must be added to the project planning and preparation of the customer process. - A configuration management activity/task must be added to the explore and rollout preparation process - A data migration verification activity/task must be added to the realization and data migration process - An integrated test and verification task must be added to the verification and training process activities.
Expert 2	- In relation to cloud ERP construction processes, from the end-user perspective, it would be a good idea to separate the construction used to introduce SaaS-based ERP from the cloud computing-based IaaS and PaaS. - It would also be a good idea to conduct additional research on the customization. - Activities/tasks related to the introduced customization of the customer must be supplemented in the explore and rollout preparation process. - There is a need for activities/tasks related to the provisioning (infrastructure, platform, and software) during the realization and data migration process. - There is a need for a usage analysis activity/task in the Post Go-Live process in the service management dimension.

The cloud-based ERP process framework was finalized to reflect the judgements of the experts. The cloud-based ERP process framework was suggested in the form of adding application content as a service (CaaS), as shown in Table 5, to clarify the scope of modularized unit services and the roles of the process framework between users and the vendor when constructing a cloud-based ERP system. This is to secure the flexibility of the framework that is useful to both users and vendors.

Based on preliminary studies [5], [13], [15], and [72], IaaS and PaaS were reconfigured by reflecting their phases and activities recently provided by vendors. CaaS and SaaS were classified based on Software Engineering Body of Knowledge (SWEBOK), a knowledge system of software engineering that is defined based on ISO/IEC 24773. SWEBOK is largely composed of SW engineering and SW management areas. In the derived activity, the element corresponding to SW engineering was placed as CaaS and that corresponding to SW management was placed as SaaS. Based on the same principle, the framework of commercial vendors such as SAP, Oracle, and MS is presented by dividing them into vendor, customer, and shared roles

based on the activity subject. Therefore, considering the SWEBOK theory and framework of commercial vendors, customers perform service preparation, testing, and data validation through the use of production functions, operations management, and support activities. In contrast, vendors perform activities related to function and technology for services, design, service and security implementation, data transfer and verification, and implementation verification and check. Considering this theoretical basis and the framework of commercial vendors, the service for customers was placed on the SaaS layer, and the vendor role was placed on the CaaS layer.

In addition, to minimize customer risk, which is the biggest advantage of this framework, an ERP SW technical review was placed in the Project Planning and Preparation of Customer Process of the PaaS and CaaS Layers to enable risk analysis by layer and process iteration cycle within one layer. This minimized the risk of cloud ERP deployment by allowing technology-focused analysis of all risks that can be derived from cloud ERP implementations, such as organizational, skills', project management, system, user, and technical risks [71].

Table 5. The process framework for cloud-based ERP construction

	Project Planning and Preparation of Customer	Explore and Rollout Preparation	Realization and Data Migration	Verification and Transition	Deployment and Distribution	Post Go-live
Software-as-a-service (end-user service)	Project governance, ERP project planning, ERP SW technical review, Preparation inner, Enterprise/ organization	ERP project kick-off	Application user training, Data migration, Quality control	Validation	Evaluation, System go-live, Inspection and completion (report)	Evaluation
Application Component-as-a-service	ERP SW technical review (Risk Analysis)	Analysis & design	Realization, Integration, Quality control	Test & verification, Transition, Training, Quality control	Deployment, System go-live	Addition and release of new service, Optimization
Platform-as-a-service	ERP SW technical review, License CAPEX [1], MSP[2] & CSP[3]	Middleware (development tools and processes), System SW (OS, DB, WAS, JDK), Runtime	Realization, Data, migration support, Quality control	Quality control	Biz. Analysis, Monitoring (security and regulation monitoring)	
Infrastructure-as-a-service	Infrastructure and solution, CAPEX, MSP & CSP	Virtual server, Virtual LAN configurations, Storage shares	Execution / Monitoring of project, Request and receive system, Quality control	Execution/ Monitoring of project, System go-live, Quality control	Management-as-a service, Execution/ Monitoring of project	Management-as-a service

1) CAPEX (Capital expenditures), 2) MSP (Managed Service Provider), 3) CSP (Cloud Service Provider)

Table 6 shows the cloud ERP construction processes finally derived through this procedure.

Expert 1 held the opinion that when cloud ERP is introduced, the customer size, data processing personnel, and links to existing legacy systems are extremely important factors in selecting the type of cloud computing-based ERP model. As such, the IaaS ERP construction model was recommended for customers requiring cloud ERP customization, PaaS for organizations with well-organized business development personnel, and SaaS for organizations with relatively small sizes whose systems are

being formed. In addition, because customization according to the customer requirements is practically unallowed for services in an SaaS format, Expert 1 was of the opinion that it is necessary to consider plans for how to flexibly respond if customers introducing SaaS want to develop their own processes as well as processes specialized for their specific industry.

Table 6. Processes, activities, and tasks for cloud-based ERP construction Framework

Process	Activity	Task
Project Planning and Preparation for Customer	Project Governance	Definition of strategy objectives, project initiation and governance, project vision and mission, high-level scoping, system requirement review
	ERP Project Planning	Collaboration between customer and vendor - Customer: application value and scoping, onboarding (on-the-job-training), customer team self-enablement, project initiation and governance, project plans, schedule and budget, project standards, infrastructure, and solution risk analysis. - Vendor: definition of project, project design, project planning, definition of project scope, strategy development, requirement review, cost review, system's initial module and goal setting, definition of requirements and project scope, review of installation plan, project blueprint
	ERP SW Technical Review	Cloud trial, ERP supporting implementation tool access, onboarding (on-the-job-training), initial system access for central business configuration, initial system access for cloud ERP S/W, ERP product review, product-company fit, checking service products, quote and simulation, interoperation scope review, provision and initialization of software, verification of software requirement satisfaction, quality control, commencement report meeting and quality control, choice of cloud-based ERP development model
	Preparation of Inner Enterprise/ Organization	Project team building, process automation review through new ERP functions and technology / process review, data research through examination and quotes of service products, setting the roles of internal organizations and teams, infrastructure, and solution
Explore and Rollout Preparation	ERP Project Kick-Off	Data migration approach and strategy, enable assessment, enable strategy, learning needs analysis for users, content development tool deployment (development tools and processes), system SW (OS, DB, WAS, JDK), runtime, quality check, phase closure and sign-off phase deliverables, system initial setting, user preparation, education, project kickoff meeting, phase closure and sign-off phase deliverables, customization requirement
	Analysis & Design	Analysis of specific business requirements, request system (product, quality), fit-to-standard analysis, customer execution of standard processes, fit-to-standard analysis documentation, integration planning and design, extension planning and design, analytics planning and design, identity and access management planning and design, new scope activation, new scope item activation for solution management, data load preparation, test planning, organizational change/configuration management impact analysis, determination of phase closure and sign-off, phase deliverables, quality management request, initial setup & system setting, quality check, data and workflow verification, service use request, major business data structure review, capturing and tracking specific items, phase closure and sign-off phase deliverables, virtual server, virtual LAN configurations, and storage shares
Realization and Data Migration	Realization	Production system request, request and receive system, alignment activities, quality system initial access, production system initial access, enable content development, required configuration before system use, solution configuration, new scope activation, new scope item activation for solution management, new country / region expansion for solution management, release cycles, setup instructions for customer driven integrations, integration setup in the quality / test landscape, analytics configuration in the quality system, identity and access management configuration, integration setup in the productive landscape, integration prerequisite, output management setup, solution extension development, solution walkthrough, test preparation & execution, enable delivery, support operations and handover plan, cutover preparation, analytics configuration in the production system, solution extension deployment on production, quality check, integration and interface verification, service execution, deployment preparation, provisioning (infrastructure, platform, and software), training plan establishment, environment setup, live input, organizational change/configuration management (OCM), phase

		closure and sign-off phase deliverables
	Data migration	Definition of migration data, refinement of migration data, collection of migration data, final data conversion, data migration execution based on usage and impact analysis, data verification, data conversion rehearsal, integration and interface data verification, phase closure and sign-off phase deliverables, project monitoring, request and receiving system, quality control
	Training	Key-user training, in-depth user training
Validation and Transition	Test & Verification	Data output test, confirmation of requirement satisfaction, functional test, process test, interoperability test, data validation, system test, system process validation, integration test
	Validation	Requirement satisfaction verification, deployment preparation in actual environment, client preparation, final user test, user approval test, solution walkthrough
	Transition	Final data conversion and migration, additional development and transition to changed module, production cutover, configuration in the production system, System Go-Live
	Training	ERP Training
	Quality Control	Continuous change management activities, quality, risk management, phase closure and sign-off phase deliverables, execution/monitoring
Deployment and Distribution	Deploy	New service deployment, execution/monitoring of project, production cutover
	Go-Live	Help desk operation, management-as-a service, biz. analysis, monitoring
	Evaluation	Periodic business closing and validation of settlement task
	Inspection and Completion Report	Final inspection and project completion, phase closure and sign-off phase deliverables
Post go-live	Addition and Release of New Service	Addition and release of new service, new scope activation, continuous improvement, revision and supplementation
	Optimization	Performance optimization, help desk operation, new user setup, fine-tuning, management-as-a service, usage analysis
	Evaluation	Operation status monitoring, evaluation of newly constructed service, evaluation of system operation/usability/quality

In regard to the desired direction of cloud-based ERP construction frameworks, Expert 2 was of the opinion that the ERP construction process for the provisioning and ERP environment settings should be provided as a self-service as soon as customers who introduce complete SaaS ERP are registered as members of the SaaS portal.

4.5. Model Analysis

A. Review of Process Framework for Cloud ERP Construction

Let us examine the characteristics of the process framework for the derived cloud ERP construction. First, the most salient characteristic is that each activity is presented according to the separation of the process derived in the framework into IaaS, PaaS, CaaS, and SaaS depending on the cloud ERP construction type. This can clarify the scope of the roles and responsibilities of the vendor and the company with respect to each software service type.

Next, let us examine the more detailed characteristics of each procedure in each process. The first process is the project planning and preparation of customer phase which consists of the following characteristics. First, detailed procedures and planning

activities must be performed by the customer in the stage of preparing for ERP construction, which can greatly help identify stakeholders and provide smooth communication. Second, project planning is common in both on-premise ERP construction and cloud ERP construction. However, while technical review remains at the requirement level when constructing on-premise ERP, the greater goal is to review the specific details of requirements in cloud-based ERP through the advance installation of ERP software. Third, the high-level requirement definition activity is derived as the most important procedure during the on-premise ERP construction process while the gap analysis of specific functions and non-functional requirements is facilitated in cloud ERP construction. Therefore, the effect of the ERP solutions is expected to be validated. Next, the standardization of regulations, assets, forms, etc. is derived as a very important task in the on-premise processes, whereas the integration with the company system and preparation of process automation are found to be very important prerequisites in cloud-based processes. Finally, the project team building is derived at the activity level in the case of on-premise ERP, but it is reduced to the task level in cloud-based ERP, demonstrating the benefits of reducing operations and management.

The characteristics of the second process, explore and rollout preparation, are as follows. During the on-premise ERP construction process, detailed plans have to be established when initiating the project and a kick-off report meeting should be held; furthermore, a lot of effort is put into constructing the project development environment. However, in the case of cloud-based ERP, general preparation tasks focus on enabling users to initialize the ERP service to derive requirements as quickly as possible, rather than focus on complex preparation for kick-off. Furthermore, compared to the on-premise process, the weight of as-is analysis and gap analysis are significantly less in cloud-based processes. This indicates that the boundary between the analysis and design is blurred in cloud-based ERP construction while the weight and importance of the tasks for the standard compliance and integration increase.

In particular, the detailed architectural design activity in the on-premise ERP construction process is dispersed among IaaS, PaaS, and CaaS in the cloud ERP construction process framework, significantly reducing and simplifying the workload. A lot of resources are injected into the prototype development, standard UI / UX design, business simulation, and master and sample data generation during the on-premise ERP construction process, whereas real-time work support is strengthened in the cloud-based development, which has evolved to operate in conjunction with artificial intelligence (AI) and decision-making services. In the past, when requirements were defined through a mock-up program, the requirement definition and analysis required a lot of time to prepare writing deliverables. In the cloud-based ERP, however, tasks can be performed in the form of planning-design, not planning-analysis-design, despite being complex ERP services.

The characteristics of the third process, the realization and data migration process framework, are as follows. Here, “Activation” has emerged as the primary theme among the tasks, meaning that the activity of selecting services in the nation or organization is performed frequently, based on already-existing modules, and it is determined that the ability to review the selected options is also required from customers. This is a good example demonstrating that one of the most prominent characteristics of cloud-based ERP is the change in “terminology.” Second, organizational change management (OCM) has emerged, which was not an important task in conventional development. In

other words, stakeholder management, communication management, organization management, human resources skill and competence management, and performance management are all important in ERP systems in terms of project management. In fact, according to a 2021 ERP report [8], change management according to organizational structure is one of the most difficult subjects to tackle in ERP operations. Third, by loading AI modules using ERP data, the system is expected to evolve in the direction of analysis and initialization based on the integration of the internal system with the services to be added to the basic tasks. Fourth, despite many issues in data migration, such as data loss and security issues, when constructing a cloud ERP system, the weight and complexity of tasks are significantly reduced compared to on-premise ERP construction. This is the result of vendors applying standardized data migration techniques and know-how as systems evolve in the direction of cloud ERP, rather than relying on the capability of persons in charge of migration for each project, as was the case in the past. Furthermore, although in cloud ERP, user environment preparation for major infrastructure services, initial-version creation of manuals, development of training materials, and operator training have been greatly reduced compared to on-premise ERP construction, a variety of user training tasks have to be performed frequently in the kick-off, realization, and deployment stages.

The characteristics of the fourth process, the validation and transition framework are as follows. The derived process shows that tests are very important in cloud ERP construction compared to on-premise ERP construction. Tests are treated as higher-level activities in cloud ERP. Tests are defined for independent processes during the cloud ERP software construction for every vendor except SAP with the goal of process validation. Second, in the case of transition, the process migration and transition are not needed if services are initiated using default modules in cloud ERP construction. Therefore, the big advantage is that service transition can be achieved by data migration based only on simple data conversion. However, in the case of customizing modules, limited transition tests, transition preparation, and transition may be selectively required. In particular, the transition process is free from the configuration tasks of the system's go-live environment, that is, checking whether the system operates normally, system monitoring, technical support, and checking the go-live operation compared to on-premise ERP. These benefits provided by IaaS are a major advantage. Furthermore, when the project is completed, tasks such as handover, planning related to work handover, stabilization support plan agreement, and support environment setup are greatly reduced or skipped.

In the fifth process, the deployment and distribution framework, the phase completion and inspection procedures are only performed as a matter of formality compared to on-premise ERP construction because of the characteristics of the transition process. In the framework of the post go-live process, the addition of new services is provided in the form of activity. Therefore, the characteristics are much more advantageous for flexible applications at the task level in cloud ERP construction although not optimized for specific corporate business logic compared to on-premise ERP construction.

B. Difference analysis with customer-based on-premise ERP process engineering

For the process analysis, the process engineering characteristics were examined based on the on-premise ERP construction process framework studied by Deok-Soo Oh et al. [24].

The customer-based on-premise ERP construction processes that were defined consist of seven phases: 1. Construction strategy planning, 2. Project kick-off, 3. Detailed analysis, 4. Detailed design and prototyping, 5. Development and testing, 6. Transition and project completion, and 7. Operation and improvement. On the other hand, the cloud ERP construction processes that were defined consist of six phases: 1. Project planning and preparation of customer, 2. Explore and rollout preparation, 3. Realization and data migration, 4. Verification and transition, 5. Deployment and distribution, and 6. Post go-live.

When differences in engineering are examined based on the above listed methods, it is found that first, the on-premise ERP construction processes adopted by most vendors are defined in a form resembling the waterfall model. However, in the case of cloud ERP construction processes, services can be provided on a story or feature basis, exhibiting characteristics very similar to that of the test-based development pursued by agile process models. Cloud ERP though, is different in that it provides a condition to activate already-existing solution applications for immediate distribution and review, instead of directly developing applications on a story or feature basis to meet the customer requirements for construction or expansion. This is in line with recent trends in technology that place importance on the customer's perspective on cooperation to more easily and quickly support business requirements, thus providing a driving force that enables customers to realize strong benefits in terms of judging the success and failure of ERP construction. Customers have a belief that the construction of cloud ERP is very successful and perceive that the return on investment (ROI) is, in fact, very strong [8].
Second, many activities required when constructing an ERP system are eliminated or reduced. In the case of on-premise ERP [24], all processes during project kick-off, detailed analysis, detailed design and prototyping, development and testing, operation and improvement, require tasks, such as change management, quality control, risk and issue management, and report management for integrated management of performance control activities. After construction, application service performance, service quality, and stabilization support are the most important tasks. However, in the case of cloud ERP, all these processes have been taken care of by the vendor, or automatically modified results are received in the form of services. In particular, when a SaaS-based ERP system is implemented, the enterprise-wide services are operated and maintained with minimal selections at the organization level. In software engineering, this is a very innovative change in terms of software management, which includes software configuration management, software engineering management, software engineering process, software engineering models and methods, and software quality, the so-called umbrella processes. For example, the information management department of an organization focuses on operations that can directly contribute to business management, such as technology standardization and planning, corporate data utilization and analysis, new technology applications (e.g., task integration and linkage with artificial intelligence solutions), rather than traditional operations, such as manual updating and upgrading, service quality control, performance management, user management, system monitoring,

and optimization. This provides opportunities for a variety of changes which can enhance the total economic impact (TEI) of the company.

Third, the risk analysis task adopted in the spiral process model is provided explicitly. The spiral model was proposed by Boehm and has the effect of reducing opportunity costs and increasing benefits of testing and feedback by evaluating the risk of failure during cloud ERP construction. When selecting cloud ERP, it is explicitly stated that risk analysis has to be performed in the strategy establishment phase. Thus, the proposed cloud ERP construction model provides an opportunity to focus on maximizing the company-wide use of ERP and the business effects, unlike the on-premise construction, in which many resources are focused on the successful construction of the ERP system. In recent years, even though the size of companies implementing cloud ERP has decreased, the big bang approach has increased by about 7 % [8]. This implies that the companies implementing cloud ERP have determined that the risks of adopting cloud-based ERP software provided by the vendor, changing and expanding technology, and integrating with existing systems are much easier to manage in terms of corporate operations compared to the past on-premise ERP construction.

Fourth, the strategic thinking about operation and maintenance has shifted: the company should be flexible about continuous change and evolution of enterprise-wide information services as work processes change. The major activities in the operation and improvement phase after the construction of on-premise ERP are bug fixes of solutions and engine uploads [24]. However, when the post go-live process of cloud ERP is examined in the derived framework, additional release of new service and activation of new service emerge as major tasks. This is in stark contrast to on-premise ERP, which is very sensitive to linkage failure and focuses only on stabilization in the operation and maintenance phase after construction of services, thus placing low priority on the requirements for new service addition, change, or expansion. In fact, a survey on the benefits of implementing cloud ERP showed that 96.6 % of organizations realize benefits in operational efficiency, 85.7 % in reporting and visibility, 80 % in updating technology, and 68.4 % in corporate growth and competition [8]. These results show that strategic thinking in implementing and operating cloud ERP is sufficiently reasonable.

C. Difference analysis based on commercial cloud ERP methodology

As the biggest difference between typical commercial cloud ERP methodologies and the process framework derived in this study, the derived framework is not limited to the SaaS level, i.e., simply introducing ERP software, but instead determines the systems and major development-related activities and tasks that must be carried out for the customer to introduce cloud-based ERP, and it defines these as IaaS and PaaS activities. By clearly defining the activities and tasks related to implementing the servers and development environment, which are omitted from conventional commercial software processes, the derived framework informs the customer of the many preparations that must actually be carried out when introducing cloud-based ERP. In addition, all procedures, activities, and tasks are defined such that they can be used universally in lieu of construction procedures that are defined by certain commercial ERP vendors, focusing on their own ERP software. As such, organizations that want to introduce a

cloud ERP can choose the construction methods through a tailoring of the process with the ERP vendor and collaboration between the organization introducing the ERP and the organization constructing it. This may help shift toward a collaborative ERP construction in which communication occurs with the customer rather than an ERP vendor-led construction. In this study, there is actually a need for customization after introducing SaaS ERP, which in this case should be clearly separated from the operating SaaS. By considering this point and separating SaaS and CaaS, it becomes easier to distinguish between reference activities and tasks when constructing each unit service.

In addition, Table 7 compares this study approach to the most widely used SaaS ERP construction methodologies provided by certain commercial vendors.

Table 7. Difference by commercial vendor-specific ERP methodology

	ASAP	OUM	Sure Step	Process Framework in present study
Stage classification terminology	Phase-activity	Phase-activity	Phase-activity	Process-Activity-task
Basis for classification	-	-	-	ISO/IEC 12207
1st Layer	**5- Phases system** 1. Preparation 2. Business Blueprint 3. Realization 4. Final preparation 5. Go Live and support	**5- Phases system** 1. Project Design 2. Configure 3. Validate 4. Transition 5. Realization	**6- Phases system** 1. Diagnostic 2. Analysis 3. Design 4. Development 5. Deployment 6. Operation	**6-processes system** 1. Customer project planning and preparation 2. Exploration and rollout preparation 3. Realization and data migration 4. Verification and transition 5. Deployment and distribution 6. Post go-live
2nd Layer	33 activities	35 activities	24 activities (optional +2)	21 activities
3rd Layer	Not defined	Not defined	Not defined	194 tasks
Usability	Vendor specific	Vendor specific	Vendor specific	All vendors and customers
Target of framework	Vendor oriented	Vendor oriented	Vendor oriented	Customer oriented
Coverage	SaaS based	SaaS oriented	SaaS oriented	All type oriented (SaaS, CaaS, PaaS, and IaaS)

5. Conclusion

In this study, a process framework was derived for cloud ERP construction, whereby each process of cloud ERP construction was examined in detail based on the derived results by comparing the process engineering characteristics to those of on-premise ERP construction. To this end, various activities from preliminary research results were collected and mapped to the construction processes of commercial cloud ERP vendors through the KJ technique, deriving as a result six processes, 21 activities, and a very broad range of tasks for each activity. The six processes consisted of project planning and preparation of customer, explore and rollout preparation, realization and data migration, validation and transition, deployment and distribution, and post go-live. There are four derived activities—project governance, ERP project planning, ERP software technical review, preparation inner enterprise / organization —for the project planning and preparation of customer process; two activities—ERP project kick-off and analysis & design—for the explore and rollout preparation process; three activities—realization, data migration, and training—for the realization and data migration process; five activities—test & verification, validation, transition, training, and quality control—for the validation and transition process; four activities—deployment, go-live, evaluation, and inspection & completion report—for the deployment and distribution process; three activities—additional release of new service, optimization, and evaluation—for the post go-live process. Specific unit tasks for construction were defined for each activity. Using the defined results, a framework was proposed by classifying IaaS, PaaS, CaaS, and SaaS according to the cloud ERP construction type for each of the six processes to suggest activities to be performed in each process. The process engineering characteristics were analyzed based on the finally derived framework, and the differences and similarities were examined through comparisons with the on-premise ERP construction.

This study provides a theoretical basis for cloud ERP construction method along with research and standardization. In addition, intrinsic activities and unit tasks are provided for each process of cloud ERP construction, distinct in practice from the on-premise ERP construction processes. The study can be used as a process tailoring tool to provide clear details of activities or tasks to all customers and vendors constructing cloud ERP systems. This will contribute to reliable cloud-based ERP construction in providing clear guidelines for smooth communication, specific preparations, and tasks to focus on for each stakeholder.

However, the cloud-based ERP construction framework in this study, which considers the customer perspective, covers the entire range of cloud computing in which ERP is constructed and used; however, an extremely limited validation was conducted during the validation phase by experts with ample experience under all cloud computing conditions, including IaaS, PaaS, and SaaS types. Therefore, a continued validation and revision must be applied based on additional reviews by experts and the use of actual examples. In future studies, it will be necessary to test these limitations, allowing the customer-oriented cloud-based ERP construction process framework, which considers the applicability, to evolve to the next stage.

References

1. Uwizeyemungu, S., Raymond, L.: Essential characteristics of an ERP system: conceptualization and operationalization. Journal of Information and Organizational Sciences, Vol. 29, No. 2, 69–81. (2005).
2. Zaitun, A. B., Zainol, Z.: The design of a DSS for the selection of ERP system and consultant, Information Management in Modern Organizations: Trends & Challenges, pp. 402-409, Mar. 2010, [Online]. Available: http://eprints.um.edu.my/8611/1/All.pdf
3. O'Leary, D. E.: Enterprise resource planning systems: Systems, Life Cycle, Electronic Commerce, and Risk. Cambridge University Press. Cambridge, UK. (2000) DOI: 10.1017/CBO9780511805936
4. Kiadehi, E. F., Mohammadi, S.: Cloud ERP: Implementation of enterprise resource planning using cloud computing technology. Journal of Basic and Applied Scientific Research, Vol. 2, No. 11, 11422-11427. (2012)
5. Weng, F., Ming-Chien, H.: Competition and challenge on adopting cloud ERP. International Journal of Innovation, Management and Technology, Vol. 5, No. 4, 309-314. (2014)p
6. https://www.forbes.com/sites/louiscolumbus/2017/04/29/roundup-of-cloud-computing-forecasts-2017/?sh=37dbe02331e8
7. https://www.brightpearl.com/cloud-erp, Accessed on Jul. 10, 2021.
8. Panorama consulting group, 2021 ERP Report
9. Panorama consulting group, 2018 ERP Report, https://www.panorama-consulting.com/resource-center/erp-industry-reports/panoramas-2018-erp-report. (2018)
10. Best ERP Software, Available: https://www.selecthub.com/erp-software/, Accessed on Jul. 20, 2021.
11. YoungLimWon SoftLab, https://www.ksystem.co.kr, Accessed on April. 10. 2022
12. Navaneethakrishnan, C.: A comparative study of cloud based ERP systems with traditional ERP and analysis of cloud ERP implementation. International Journal of Engineering and Computer Science, Vol. 2, No. 9, 2866-2869. (2013).
13. Elragal, A., El-Kommos, M.: In-house versus in-cloud ERP systems: A comparative study. Journal of Enterprise Resource Planning Studies.Vol. 2012, No. 659957, (2012). DOI: 10.5171/2012.659957
14. Motalab, M. B., Shohag, S. A. M.: Cloud computing and the business consequences of ERP use. International Journal of Computer Applications, Vol. 28, No. 8, 31-37. (2011), DOI: 10.5120/3406-4751
15. Schubert, P., Adisa, F.: Cloud computing for standard ERP systems: Reference framework and research agenda. Arbeitsberichte aus dem Fachbereich Informatik, 1-29. (2011)
16. Mukherjee, K., Sahoo, G.: Green cloud: An algorithmic approach. International Journal of Computer Applications, Vol. 9, No. 9, 0975-8887. (2010), DOI: 10.5120/1417-1914
17. Okezie, C. C., Udeze, C., Okafor, K. C.: Cloud computing: A cost effective approach to enterprise web application implementation: A case for cloud ERP web model. Academic Research International, Vol. 3, No. 1, 432-443. (2012).
18. Lee, M. J., Wong, W. Y., Hoo, M. H.: Next era of enterprise resource planning system review on traditional on-premise ERP versus cloud-based ERP: Factors influence decision on migration to cloud-based ERP for Malaysian SMEs/SMIs. in 2017 IEEE Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia. 48-53. (2017). DOI: 10.1109/SPC.2017.8313020
19. Peng, G. C. A., Gala, C.: Cloud ERP: A new dilemma to modern organisations? Journal of Computer Information Systems, Vol. 54, No. 4, 22-30. (2015). DOI: https://doi.org/10.1080/08874417.2014.11645719
20. Seethamraju, R.: Adoption of software as a service (SaaS) enterprise resource planning (ERP) systems in small and medium sized enterprises (SMEs). Information Systems Frontiers, Vol. 17, No. 3, 475-492. (2015). DOI: https://doi.org/10.1007/s10796-014-9506-5

21. Johansson, B., Alajbegovic, A., Alexopoulos, V., Desalermos, A.: Cloud ERP adoption opportunities and concerns: A comparison between SMES and large companies. In Pre-ECIS 2014 Workshop" IT Operations Management"(ITOM2014), Konstanz, Germany. pp. 1-13. (2014).
22. Alajbegovic, A., Alexopoulos, V., Desalermos, A.: Factors influencing cloud ERP adoption: A comparison between SMEs and large companies. Department of Informatics, Lund University. (2013).
23. Arora, R., Gera, S., Saxena, M.: Mitigating security risks on privacy of sensitive data used in cloud-based ERP applications. 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 458-463. (2021). DOI: 10.1109/INDIACom51348.2021.00081.
24. Oh, D. S., Kim, H. S., Kim, S. H.: A development of the customer based on-premise ERP implementation process framework. International Journal of Advanced Smart Convergence, Vol. 10, No. 3, 257-278. (2021). DOI: 10.7236/IJASC.2021.10.3.257
25. Nagpal, S., Khatri, S. K., & Kumar, A.: Comparative study of ERP implementation strategies. In 2015 Long Island Systems, Applications and Technology, IEEE, 1-9 (2015) DOI: 10.1109/LISAT.2015.7160177
26. Tongsuksai, S., Mathrani, S., & Taskin, N.:Cloud enterprise resource planning implementation: a systematic literature review of critical success factors. In 2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), IEEE, 1-8, (2019, December). DOI: 10.1109/CSDE48274.2019.9162373
27. Peng, G. C. A., & Gala, C.:Cloud ERP: a new dilemma to modern organisations?. Journal of Computer Information Systems, Vol. 54, No. 4, 22-30. (2014)
28. Patel, A., Seyfi, A., Tew, Y., & Jaradat, A. :Comparative study and review of grid, cloud, utility computing and software as a service for use by libraries. Library Hi Tech News. (2011)
29. Chang, M. K., Cheung, W., Cheng, C. H., & Yeung, J. H.:Understanding ERP system adoption from the user's perspective. International Journal of production economics, Vol. 113, No. 2. 928-942. (2008). DOI: 10.1016/j.ijpe.2007.08.011
30. Kraljic, A., Kraljic, T., Poels, G., & Devos, J.:ERP implementation methodologies and frameworks: a literature review. In 8th European Conference on IS Management and Evaluation (ECIME) Academic Conferences and Publishing International Limited.309-316. (2014)
31. Dick K., Adam C.: Oracle ERP Cloud Implementation Methodology - ERP Cloud SIG Meeting, 2021 computer technology resources, CTR, Inc., Quest ERP Cloud Special Interest Group (2021) https://www.youtube.com/watch?v=DnYnJhzu8nc, Accessed on March. 2022
32. Link, B., & Back, A. :Classifying systemic differences between software as a service-and on-premise-enterprise resource planning. Journal of Enterprise Information Management. (2015).
33. Kim. S.H. and Kim. W.J.:Development of Operational Quality Measurement Attributes of Application Software Using KJ Method, Int. J. of Software Eng. & Its Applications, Vol. 8, No. 3, 171-188. (2014) DOI: 10.14257/ijseia.2014.8.3.16
34. Project Management Institute, A Guide To The Project Management Body Of Knowledge (PMBOK-Guide) – Sixth version, Pennsylvania, USA: Project Management Institute, Inc, (2017)
35. Goossens, L. H. J., & Cooke, R. M.: Expert judgement elicitation in risk assessment. In *Assessment and management of environmental risks*, pp. 411-426. Springer, Dordrecht. (2001)
36. Matros, R., Rietze, C. and Eymann, T.: SaaS und Unternehmenserfolg: Erfolgskategorienfür die Praxis. In: Benlian, A., Hess, T. and Buxmann, P. (Eds), Software-As-A-Service. pp. 239–254. Gabler, Wiesbaden (2010) DOI: 10.1007/978-3-8349-8731-0_16

37. Chen, L., Soliman, K.S.: Managing IT outsourcing: a value-driven approach to outsourcing using application service providers. Logistics Information Management, Vol. 15, No. 3, 180–191. (2002) DOI: 10.1108/09576050210426733
38. Farah, S.: Cloud computing or software as a service – which makes the most sense for HR? Employment Relations Today, Vol. 36, No. 4, 31–37. (2010) DOI: 10.1002/ert.20271
39. Fuller, S., McLaren, T.: Analyzing enterprise systems delivery modes for small and medium enterprises. Proceedings of Americas Conference on Information Systems, paper 380, Lima, August 12–15. (2010)
40. Gill, R.: Why cloud computing matters to finance, Strategic Finance, Vol. 92, No. 7, 43–47. (2011)
41. Höß, O., Weisbecker, A., and Spath, D.: Software as a Service – Potenziale, Risiken und trends. Information Management & Consulting, Vol. 23, No. 4, 6–11 (2008)
42. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., and Ghalsasi, A.: Cloud computing – the business perspective. Decision Support Systems, Vol. 51, No. 1, 176–189. (2011) DOI: 10.1016/j.dss.2010.12.006
43. Olson, D.L., Wu, D.D.: Multiple criteria analysis for evaluation of information system risk. Asia-Pacific Journal of Operational Research, Vol. 28, No. 1, 25–39. (2011) DOI: 10.1142/S021759591100303X
44. Sharma, S.K., Gupta, J.N.: Application service providers: Issues and challenges. Logistics Information Management, Vol. 15, No. 3, 160–169. (2002) DOI: 10.1108/09576050210426715
45. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, Vol. 34, No. 1, 1–11. (2010) DOI: 10.1016/j.jnca.2010.07.006
46. Benlian, A., Hess, T.: Drivers of SaaS-adoption – an empirical study of different application types. Business Information Systems Engineering, Vol. 1, No. 5, 357–369. (2009) DOI: 10.1007/s12599-009-0068-x
47. Brass, D., Zimmermann, R.: Software as a Service – am Beispiel einer business intelligence-Lösung in der Logistik. HMD, Vol. 47, No. 5, 42–51. (2010) DOI: 10.1007/BF03340511
48. Buxmann, P., Hess, T., Lehmann, S.: Software as a service, Business and Information Systems Engineering the international journal of Wirtschaftsinformatik, Vol. 50, No. 6, 500–503. (2008) DOI: 10.1007/s11576-008-0095-0
49. Herbert, L.: When Software-as-a-Service makes sense. Supply and Demand Chain Executive 1. (2007) available at: www.sdcexec.com/
50. Mathew, M., Nair, S.: Pricing SaaS models: Perceptions of business service providers and clients. Journal of Services Research, Vol. 10, No. 1. 51–69. (2010)
51. Sontow, K., Kleinert, A.: Software as a Service – Die schlanke Zukunft für ERP-Lösungen? Ergebnisse eine Anwenderbefragung. ERP Management, Vol. 6, No. 4, 24–27. (2010b)
52. Weiping, L.: An analysis of new features for workflow system in the SaaS software. In: Proceedings of International Conference on Information Systems, ACM pp. 110–114, Milan, May 6–10 (2009) DOI: 10.1145/1655925.1655946
53. Xin, M., Levina, N.: Software-as-a Service model: Elaborating client-side adoption factors. In: Proceedings of International Conference of Information Systems, Paris, paper 86 (2008)
54. Benlian, A., Hess, T.: The risks of sourcing Software as a Service – an empirical analysis of adopters and non-adopters. In: Proceedings of European Conference on Information Systems, paper 142, Pretoria (2010a)
55. Wu, W., Lan, L.W., and Le, Y.: Exploring decisive factors affecting an organization's SaaS adoption: A case study. International Journal of Information Management, Vol. 31, No. 6, 556–563. (2011) DOI: 10.1016/j.ijinfomgt.2011.02.007
56. Benlian, A., Hess, T.: Chancen und Risiken des Einsatzes von SaaS – Die Sicht der Anwender. In Benlian, A., Hess, T. and Buxmann, P. (Eds), Software-as-a-Service, Gabler, Wiesbaden, 173–178. (2010b) DOI: 10.1007/BF03248241

57. Bezemer, C., Zaidmann, A.: Multi-tenant SaaS applications: maintenance dream or nightmare? Joint proceedings of ERCIM EVOL and IWPSE, ACM, 88–92, Antwerp (2010) DOI: 10.1145/1862372.1862393
58. Katzan, H. Jr., Dowling, W.A.: Software-as-a-Service economics. The Review of Business Information Systems, Vol. 14, No. 1, 27–37. (2010) DOI: 10.19030/rbis.v14i1.500
59. Prakash, V., Savaglio, C., Garg, L., Bawa, S., & Spezzano, G.: Cloud-and Edge-based ERP systems for Industrial Internet of Things and Smart Factory. Procedia Computer Science, 200, 537-545. (2022) DOI: 10.1016/j.procs.2022.01.251
60. Mietzner, R., Metzger, A., Leymann, F., Pohl, K. Variability modeling to support customization and deployment of multi-tenant-aware software as a service application. ICSE Workshop on Principles of Engineering Service Oriented Systems, IEEE Computer Society, 18–25, Vancouver, BC (2009) DOI: 10.1109/PESOS.2009.5068815
61. Repschläger, J., Pannicke, D., Zarnekow, R.: Cloud computing: Definitioen, Geschäftsmodelle und Entwicklungspotenziale. HMD, Vol. 47, No. 5, 6–15. (2010) DOI: 10.1007/BF03340507
62. Sääksjärvi, M., Lassila, A., Nordström, H.: Evaluating the software as a service business model: From CPU time-sharing to online innovation sharing. Proceedings of IADIS International Conference on E-Society, 177–186, Qawra (2005)
63. Choudhary, V.: Comparison of software quality under perpetual licensing and software as a service. Journal of Management Information Systems, Vol. 24, No. 2, 141–165. (2007) DOI: 10.2753/MIS0742-1222240206
64. Kakabadse, A., Kakabadse, N.: Application service providers (ASPs): New impetus for transformational change. Knowledge and Process Management, Vol. 9, No. 4, 205–218. (2002) DOI: 10.1002/kpm.149
65. Altaf, F., Schuff, D.: Taking a flexible approach to ASPs. Communications of the ACM, Vol. 53, No. 2, 139–143. (2010) DOI: 10.1145/1646353.1646389
66. Löwer, U.M., Picot, A.: Web services – technologie-hype oder strategie-faktor? Information Management and Consulting, Vol. 17, No. 3, 20–25. (2002)
67. Kim, W., Kim, S.D., Lee, E., Lee, S.: Adoption issues for cloud computing. Proceedings of International Conference on Advances in Mobile Computing and Multimedia, ACM, 3–6, Kuala Lumpur (2009) DOI: 10.1145/1821748.1821751
68. Waters, B.: Software as a Service: A look at the customer benefit, Journal of Digital Asset Management, Vol. 1, No. 1, 32–39. (2005) DOI: 10.1057/palgrave.dam.3640007
69. Heart, T.: Who is out there? Exploring the effects of trust and perceived risk on SaaS adoption intentions. Data Base for Advances in Information Systems, Vol. 41, No. 3, 49–68 (2010) DOI: 10.1145/1851175.1851179
70. Boehm, B. W.: A spiral model of software development and enhancement. Computer, Vol. 21, No. 5, 61–72 (1988) DOI: 10.1109/2.59
71. Sravan M, S., & Archana, M.: Study on the ERP Implementation Methodologies on SAP, Oracle NetSuite, and Microsoft Dynamics 365: A Review. arXiv e-prints, arXiv-2205. (2022).
72. Hakim, A., & Hakim, H.: A practical model on controlling the ERP implementation risks. Information systems, vol. 35, No. 2, 204-214. (2010). DOI: 10.1016/j.is.2009.06.002

Deok-Soo Oh received his bachelor's degree from Kyonggi University. He studied master's degree in the department of IT convergence SW engineering at the Korea University of Technology and Education. He is currently working as an ERP implementation senior consultant at Younglimwon Soft Lab.

Hyeong-Soo Kim received a bachelor's degree at Sungkyunkwan University of Korea. He studied master's degree in the department of IT convergence SW engineering at the Korea University of Technology and Education. Currently, He is a general manager in Business Administration Department at Forelink company.

Seung-Hee Kim received her bachelor's degree from Dongguk University, master's degree from Yonsei University, and doctorate degree in Industry Information System of IT Policy School from Seoul National University of Science & Technology of Korea. Currently, she is an associate professor at Korea University of Technology and Education. The major interest areas consist of blockchain, software engineering and optimization.

Received: December 30, 2021; Accepted: August 05, 2022.

Computer Science and Information Systems 20(1):51–76 https://doi.org/10.2298/CSIS220118046K

SBEO: Smart Building Evacuation Ontology

Qasim Khalid[1], Alberto Fernandez[1], Marin Lujak[1⋆], and Arnaud Doniec[2]

[1] CETINIA, Universidad Rey Juan Carlos, Mostoles (Madrid), Spain
qasim.khalid@urjc.es
alberto.fernandez@urjc.es
marin.lujak@urjc.es
[2] IMT Nord Europe, 59500 Douai, France
arnaud.doniec@imt-nord-europe.fr

Abstract. Semantically rich depiction of the concepts for context-aware indoor routing brings appealing benefits for the safety of occupants of smart spaces in emergency evacuation. In this paper, we propose Smart Building Evacuation Ontology (SBEO[3]), a reusable ontology for indoor spaces, based on three different data models: user, building, and context. We provide a common representation of indoor routing and navigation, describe users' characteristics and preferences, grouping of individuals and their role in a specific context, hazards, and emergency evacuation. Among other characteristics, we consider abilities of individuals, safety and accessibility of spaces related to each person, intensity, impact, and severity of an emergency event or activity. SBEO is flexible and compatible with other ontologies of its domain, including SEAS, SSN/SOSA, SEMA4A, and empathi. We evaluate SBEO based on several metrics demonstrating that it addresses the information needs for the context-aware route recommendation system for emergency evacuation in indoor spaces. In the end, a simulation-based application example exploits SBEO using Context-Aware Emergency Evacuation Software (CAREE)[4].

Keywords: ontology, linked data, smart building, hazard detection, emergency evacuation, indoor route recommendation, navigation.

1. Introduction

Ambient Intelligence (AmI) represents a responsive electronic environment that reacts to the actions of each person and the physical objects within itself. It has been growing fast as a multi-disciplinary field with a high impact on society for the last three decades [44].

It can be used to combine hazard detection and disaster management [5], crowd management [33], and route recommendation [9] in smart spaces [65] to obtain real-time information and aid decision making in dynamically changing emergency evacuation scenarios in large smart buildings with multitudes of occupants.

The objective of both outdoor and indoor navigation systems is to find a path for each user from their initial to target location that optimizes one or more of given performance indicators (e.g., distance, time and/or difficulty) [14]. Dynamic context-aware adaptation of the evacuation route and its communication to each evacuee are required for efficient

⋆ Corresponding author
[3] SBEO can be accessed at: https://w3id.org/sbeo
[4] https://github.com/qasimkhalid/CAREE

evacuation (see, e.g., [23]). While outdoor navigation systems use Global Positioning System (GPS) receivers, indoor navigation cannot rely on them due to the overlapping of the signal through the storeys of the building. Thus, other technologies must be employed for positioning (see, e.g., [46]). For example, Proximity-based Systems, WiFi-Based Systems, Ultra Wide-band Systems.

It becomes challenging and complex to handle indoor navigation in real time when various objectives are combined along with the localization of people. This is the case in the recommendation of routes to users based on their physical abilities and preferences during everyday or emergency scenarios. There are multiple reasons for its complexity. First, indoor location of persons is not entirely precise, always leaving a margin of error. Second, the (close to) real-time processing and fusion of data coming from heterogeneous sensors into a consistent, accurate, and useful information describing the evacuation context is a prerequisite for getting around the effects of possible emergency setbacks.

Here, we use ontologies to describe concepts and relationships between entities as a formal way of conceptualizing the domain knowledge. Ontologies are a key component of the semantic web [61] that is used to represent data, and provide a domain-specific knowledge for the representation of the metadata [48]). Previously, various ontologies and data models for indoor navigation, routing, and emergency and crisis management, were proposed for conventional buildings and spaces without ambient intelligence [4,27,58,15,28,40].

In [31], a semantically-enriched distributed architecture for context-aware and real-time evacuation guidance in indoor smart spaces was proposed. The architecture uses a multi-agent system for the coordination of the evacuation where each agent is responsible of the semantic reasoning concerning the safety of its assigned physical space and uses an ontology for indoor emergency evacuation. As a continuation of the previous work, in this paper, we propose an ontology for smart space context-aware route recommendation to evacuees with smart devices. We name the proposed ontology *Smart Building Evacuation Ontology* (SBEO). SBEO is composed of three modules: User Model, describing the characteristics (i.e., physical abilities) and preferences of an evacuee; Building Model that considers the routing, and geometry, devices and elements other than structural components of the building; and the Context Model, which illustrates the contextual information about the building and the evacuees.

The SBEO ontology is inline with terminologies used in the domain of indoor route recommendation and navigation. It is compatible with several other state-of-the-art ontologies and systems such as, SEAS [29], SOSA/SSN[24], SEMAA4a [40], Indoor Navigation Ontology (INO) and User Navigation Ontology (UNO)[26,58], and General User Model Ontology (GUMO)[21]. It was implemented using OWL 2[5] and the Protégé[6] development environment. SBEO is available at https://w3id.org/sbeo# and holds a GPL-3.0 license.

To the best of our knowledge, this is the first work that combines the concepts of smart spaces with context awareness, route recommendation and hazard detection considering users' relevant evacuation characteristics and preferences. The main contributions of SBEO are expressiveness, which provides a hierarchical description of the concepts in indoor emergency routing in smart buildings. In addition, as sharing and reusability

[5] https://www.w3.org/TR/owl2-overview/

[6] https://protege.stanford.edu/

are considered the fundamental concepts in the ontology engineering field [41], the proposed ontology is also reusable. Here, reusable refers to the adaptation of SBEO according to the need of the user and application. In other words, we can extend the models of SBEO individually. For example, people with impairments not covered thus far might also be described using the same ontology if needed. Consequently, concepts associated with such people might also be extended accordingly. Similarly, in terms of buildings, although SBEO covers the concepts for both conventional buildings and smart buildings, if any new type of buildings or related concepts is introduced, the ontology can easily be broadened. It also improves the automation, accountability, real-time context awareness, information sharing, and personalised routing in smart space evacuation.

The rest of the paper is organized as follows: In Section 2, we highlight related work in emergency management, user and crowd modeling, smart environments, and indoor routing. We describe the followed methodology in the development of SBEO together with the ontology audience and scope in Section 3. Section 4 states the competency questions and formal requirements to be met by the proposed ontology. SBEO is presented in Section 5. In Section 6, the proposed ontology is evaluated using various metrics found in the literature. In Section 7, a simulation-based application example of SBEO is described using Context-Aware Emergency Evacuation (CAREE) system. A task-based evaluation is performed using a hazard context. We conclude the paper with some proposed improvements and future work in Section 8.

2. Related Work

This section provides an overview of related state-of-the-art ontologies together with the positioning of SBEO.

2.1. Ontologies for emergency and crisis management

A general ontology for emergency response by Li et al. proposed in [30] includes the concepts of *response preparation, emergency response, emergency rescue and aftermath handling* and relevant properties and relations that connect these concepts.

The ontology for massive crisis management proposed in [50] covers the concepts related to the allocation of resources and the crisis impact related to the time and place of the crisis occurrence. An ontology-based proactive approach to enhance the response time during both natural (such as earthquake, tsunami, etc) and anthropic (such as terrorist attack, kidnapping, etc) events in [12] covers the concepts of the context, impact, and the services provided during an incident.

Sicilia and Santos [52] described Basic Formal Infrastructure Incident Assessment Ontology (BFiaO) to represent the adverse effects of incidents. BFiaO connects the concepts related to incidents, their causes and evolution, and their possible outcomes. Santos et al. [49] broadened BFiaO and made it more consistent with Coordination of Emergencies and Tracking of Actions and Resources (CESAR) data model to minimise the aftereffects of an incident. They modelled the concepts for the identification of events, mission, and resources and developed a set of rules to anticipate possible chain events connected with the existing/past events, such that the first response officers could prioritise their tasks to avoid jeopardy.

Recently, Gaur et al. [19] presented a rich ontology[7] for emergency management and planning during hazard crises including concepts related to emergency response, hazard type, impact, phase, events, involved individuals, and provided services.

In terms of notification services during emergency scenarios, Malizia et al. [35] proposed an ontology to express the concepts of emergency notification messages with respect to various kinds of users. They developed a class named EMEDIA (Emergency and MEDIA technologies) that provides the concepts and relations about emergency and communication devices and technologies. They integrated that class with other existing ontologies that describe accessibility guidelines, and users' profiles and action capabilities. Afterwards, Onorati et al. [40] extended their work by introducing some new concepts to express personalised routes based on the users' physical abilities, familiarity with the environment, preferences, location, available media for notification, and characteristics of the surrounding.

Bitencourt et al. [8] developed a formal domain ontology to describe the emergency response protocols of fire in buildings considering incident details, information about the victims, possible actions, planning, and operational phases. Related to gathering and medical emergency response, Haghighi et al. [20] presented *Domain Ontology for Mass Gathering* (DO4MG) considering the concepts related to gathering types, venue details, features of the crowd, environmental factors and general mass gathering plans.

One of the most comprehensive, general-purpose light-weight ontologies that can be used in Internet of Things and smart spaces is *Sensor, Observation, Sample, and Actuator* (SOSA) ontology [24]. SmartEnv ontology [3] covers various aspects of a smart environment such as sensing, networking, event, and topology, while *Smart Energy Aware Systems* (SEAS) ontology [29] couples the concepts related to energy and grid, and conceptualises city and building structures, time, weather, and user comfort.

2.2. Ontologies for spatial modeling, indoor navigation and routing

Rasmussen et al. [45] presented a core vocabulary named *Building Topology Ontology* (BOT) to describe the topology of buildings, along with its storeys and spaces. The BOT ontology is completely compatible with other ontologies in the domain such as SOSA and SSN[8].

For navigational and routing purposes, Brückner et al. [27] proposed an indoor-outdoor ontology-based data model for both robots and humans to develop a routing graph with the help of spatial information. Similarly, Yang and Worboys [64] also presented an ontological model for both outdoor and indoor navigation.

Indoor Navigation Ontology (INO) [58] covers concepts of navigation in indoor spaces and the geometry of buildings including multiple floors, elevators, points of interest[9], corridors, exits, etc. The ontology for indoor routing based on American Disability Act (ADA) standards [15] deals with the geometry of a building in such a way that its connections can be represented as horizontal and vertical paths for routing purposes.

[7] https://w3id.org/empathi/

[8] https://www.w3.org/TR/vocab-ssn/

[9] A point of interest is defined as any object or physical space that might be of importance or useful for the occupants of the building. For example, an (emergency) exit, location (or space) where a person is located subject to specific criteria, fire safety devices such as fire extinguisher, firehouse, fire door.

In the *Ontology Crowd Simulation* (ONTOCS) [10], the concepts of an indoor environment together with its geometry are modelled in the context of emergency evacuation in terms of the routes, travel time and distance.

2.3. Ontologies for user and crowd modeling

One of the pioneer works in the domain of user modeling has been carried out by Heckmann et al. [21]. They developed General User Model Ontology (GUMO) to describe the basic attributes of users such as demographics, abilities, proficiencies, states (emotional, physiological, mental), role, and so forth.

Later on, Kikiras et al. [26] developed User Navigation Ontology (UNO) on the basis of multiple wayfinding theories related to cognitive science, psychology, sensor abilities and physical characteristics of human beings. UNO covers the concepts related to users' navigation in indoor environments based on their demographics, cognitive characteristics, sensor- and motor- abilities, and navigational and interface preferences. Subsequently, Kritsotakis et al. [28] broadened the concepts of UNO and presented a User tracking Ontology (UTO) to describe the concepts for context awareness and tracking of users. On the other hand, Dudas et al. [15] also modeled the users in their ontology based-on American Disability Act (ADA) standards, in which they conceptualize users' preferences, familiarity with the building, and disabilities.

Similarly, Boje and Li [10] also modeled the information about the crowd in their simulated framework. They named it Crowd Simulation Information Model (CSIM) that covers the concepts related to persons, such as preferences, exit choices, and speed.

Whilst the above-mentioned works are interesting and provide the concepts about the geometrical information of the buildings, user characteristics, and routing and navigation in indoor environments, some of them are not available online, hence cannot be reused. In addition, the existing works do not provide sufficient information about the context awareness of the building and persons, especially in emergency management context. By comparison, SBEO aims to conceptualize context-aware indoor route recommendation and emergency evacuation. The proposed ontology not only describes the concepts related to building topology, routing and navigation, classification of users with respect to their abilities and preferences, but also conceptualizes context awareness, such as detection of any hazard, intensity, severity and impact of activities and events, evacuation action plan, social groups, movement of people, and people notification. Furthermore, SBEO has been published online and made available for public after being developed using state-of-the-art methodologies found in the literature.

3. Design Methodology

To date, several methodologies have been proposed for building ontologies, e.g., *Methontology* is a methodology to build ontologies from scratch [18], On-To-Knowledge[56], Digilent[42], Neon[54], and Ontology development 101[38]. In order to develop SBEO, we selected Methontology framework because it allows to develop the ontology from scratch, and also recommends to reuse the concepts from the existing meta-ontologies. In addition, as the ontology development is an on-going process due to the evolving prototype life cycle nature of this methodology, it permits the ontology authors to update the ontology from any of its phases.

Methontology proposes a process consisting of six steps: *specification*, where the purpose of the ontology, intended audience, scenarios of its use, scope and requirements are taken into account; *knowledge acquisition*, where several techniques can be used to get the specific and detailed knowledge about the topic of the ontology; *conceptualization*, where all the useful or potentially usable domain knowledge is brought together to create a conceptual model, and that model is compared with the existing ontologies to check its scope, completeness and reusability; *integration and implementation*, where to support the reuse of definitions, the concepts of the proposed conceptual model are scrutinized with the existing ontologies as per their scope, and then implemented the filtered concepts using one of the standard languages; *evaluation*, where the each activity of the ontology development process is verified and validated using various methods and metrics; and *documentation*, where each phase of the ontology development process is documented in natural language available on the Web or in the scientific literature.

Knowledge acquisition. Knowledge acquisition is a primitive stage for ontology development that is also considered as an on-going process. In the same way, we also reviewed the literature to acquire relevant knowledge and to get a broader view of the needs and requirements for the creation of an ontology (i.e., SBEO). In particular, the studies reported in: [58,15,28,40,10,32,7,2,37], represent the systems and/or models for indoor route recommendation during emergency evacuation; [1,55,11], depict crowd management and grouping during hazardous situations using simulated environments; [25], discuss indoor routing for people with special needs; [29], describes building geometry using an ontology; [65,63], define the smart spaces, together with its requirements, and [34,16], discuss some case studies related to the user behaviour during an emergency evacuation in a smart indoor environment.

Scope and Audience. SBEO offers a data model for building geometry, devices and elements not only regarding structure, route sets based on building topology, users' characteristics and preferences. It also considers the context awareness of both buildings (e.g., hazard detection, the status of spaces and evacuation routes in terms of availability and occupancy, severity and impact of activities and events, and safety of spaces) and an user/occupant (e.g., route tracking, coordination with their evacuation group, adaptation of an evacuation route in terms of fitness and accessibility to spaces). Hence, the proposed ontology has an ample scope to couple the information about a building with its occupants in order to use it for indoor localization, detection of a hazard or disaster, and preference-based routing during emergency evacuation. The target audience of SBEO is building occupants (e.g., visitor, resident, worker), building managers, civil engineering specialists, indoor designers, and architects involved in building design and development, but also researchers in building evacuation safety.

4. Ontology specification

From the requirements engineering point of view, we set two goal-level requirements that should be fulfilled by the proposed ontology. First, formal requirements, which are used to express the needs of the domains covered by the ontology, including preference-based route recommendation and context awareness in smart buildings; and Second, functional requirements, in the form of specific competency questions which must be answered by the ontology.

4.1. Formal Requirements

In terms of formal requirements, SBEO must be able to represent the concepts related to the users (i.e., occupants), the buildings, and the context related to both users and the building. These are as follows:

Users:

- Demographics of a person (e.g., age, name, individual and group identity numbers, and family members).
- Physical abilities (e.g., mental, spatial, sensor, mobility).
- Navigational and routing preferences (e.g., avoidance of stairs or crowded areas, fastest route, simplest route with least turns).
- Level of involvement (i.e., role) while performing a specific activity (e.g., either a person is dependent or responsible for others during immediate egress from an area).
- Type of an impairment of the person (if it exists).

Buildings:

- Structural elements (e.g., stairs, elevators, corridors, rooms).
- Devices installed in the building, which are not a part of the building structure (e.g., sensors, equipment for safety and access control).
- Representation of a building as a traversable graph (i.e., for routing purposes).
- Classification of routes (e.g., shortest path and simplest path [14]);

Context related to the users and buildings:

- Individuals' fitness status (e.g., fit, injured).
- Motion status (e.g., running, walking, standing).
- Deviation status (e.g., classification of persons based on the
- Frequency of deviations from their provided route).
- Safety level of spaces concerning each person.
- Availability and accessibility of spaces (where the availability of space states that either space is usable or not, the accessibility of space, on the other hand, refers to a specific person or type of persons, either it is accessible for him/her/them or not).
- Intrinsic concepts related to activities and events (e.g., type, starting and ending time), along with their effects. For example, Intensity which refers to the magnitude, severity which relates to the specific persons and varies accordingly, and Impact, that refers to their effects on users.
- Comprehensive concepts, such as time taken by an individual and role of an individual while performing an activity (e.g., responsible, visitor, group leader).

4.2. Functional Requirements: Competency Questions

For functional requirements, we use competency questions. A competency question (CQ) is a question in a natural language that is supposed to be answered by an ontology. Usually, it has a specific pattern [60]. Ren et al. [47] defined some patterns using a feature-based modelling method to describe competency questions.

Table 1. Some sample competency questions to be asked by SBEO.

Module Type	Competency Questions
	User Model
Characteristics	1. Who is not capable of running? 2. How many families are located in the building? 3. Who has a bad quality of hearing ability (in the building)? 4. What are the types of people concerning their physical characteristics?
Preferences	5. What are the route preferences (for emergency evacuation, e.g., simplest path, shortest path) of each person? 6. What are the notification preferences (in terms of description, e.g., audio, textual) of each person?
	Building Model
Spatial Information	7. What is the relative occupancy ratio of all corridors? 8. How many points of interest are located on each floor of the building? 9. Which other spaces are adjacent to a specific space (e.g., kitchen) in the building? 10. Which space (e.g., a specific building block) is a sub-part of which space (e.g., building)? 11. What is the area of all corridors (it can be of any shape, such as, rectangular, circular, trapezoidal or triangular)? 12. Which spaces are excluded (due to any reason such as limited access on account of mobility impairment or privacy policy of spaces, e.g., hotels) for which person?
Route Graph	13. How many nodes and edges are there in the graph-based representation of the building? 14. What is the type of each route in terms of its graph-based representation (e.g., Shortest Path or Simplest Path)? 15. What is the travel time of all exit routes for each person (the starting and ending points of each exit route are considered as origin and destination, respectively)?
Devices	16. Who is using a hand-held device, and of what type? 17. Which sensors are installed in each space of a specific type (e.g., office)? 18. How many fire protection devices are installed on the same floor where a specific person is located?
	Context Model
Building	19. Which activities (e.g, visit, evacuation, shopping) are being done in the building? 20. What is the availability status (i.e., Available or Unavailable) of each space?
User	21. Where is each person located in the building? 22. What is the role of each member within any group? 23. How many times a person has deviated from the provided path? 24. What is the fitness status (i.e, Exhausted, Fit, or Injured) of each person? 25. Which route is assigned to whom of which group (refers to a number of people that are classified together, e.g., a family)? 26. What is the motion state of each person (refers to the movement of a person, e.g., walking, standing, running, rolling, or scooting)? 27. What is the navigational state of each person (refers to the state while following a path to check whether a person is following the provided path or deviating from it)?
Event (e.g., Emergency Evacuation)	28. Is there an incident in the building? 29. At what time an incident occurred? 30. What is the availability status of the spaces that are a part of emergency evacuation routes? 31. How many groups are still in the process of evacuating the building? 32. What is the impact of activities on persons having the mild quality of seeing ability? 33. What is the severity of the incidents for mobility-impaired persons (of all types)? 34. What are the intensities (refers to the magnitude or strength) of the events occurred? 35. Who has evacuated the building successfully (refers to the activity status of a person who completes his/her provided exit route)?

In this study, we adopted the same patterns to determine the scope of the proposed ontology. Table 1 includes some competency questions the proposed ontology should answer. The motivation for choosing these competency questions is based on our previous work (see, e.g., [31], [33], and [32]), where not only the concepts related to both buildings and users were limited, but it also lacked the contextual information of these entities. In this regard, most of these questions are explicitly shaped to answer the specific attributes of buildings, users, and the relevant context. Nevertheless, the potential ontology user may develop their own set of customized competency questions concerning the application type within the scope of the ontology. On the other hand, these competency questions might also be used as a metric to evaluate each ontology module such that their answers could be used in the typical application scenarios the ontology is aimed at, like the task-based evaluation mentioned in Section 6 and the application described in Section 7.

The table is divided into three information models: user, building and context. The first column expresses a module type for each information model, and the second column states the list of competency questions which cover the aforementioned information model and module type.

In the information model, firstly, *User model* represents the occupants (of any category such as visitor, resident, worker) of the building, along with its two modules to indicate their demographics and preferences. Secondly, *Building model* portrays the indoor spaces (and only those outdoors ones which are used to connect indoor ones), together with its module types and relevant competency questions. Lastly, *Context model* denotes context awareness of the aforesaid information models, hence its module types—user and building. The event module type is also added in this model as per the scope of the ontology.

5. Ontology Description

This section describes SBEO based on the knowledge acquired from the literature and the specification mentioned in the previous sections. SBEO reuses various concepts from the existing ontologies such as FOAF[10], Semantic Sensor Network Ontology[11], The Ordered List Ontology[12], and SEAS Building Ontology[13]. On the contrary, some ontologies e.g., Indoor Navigation Ontology (INO)[58], User Navigation Ontology (UNO)[58], User Tracking Ontology (UTO)[28], are not available online. Therefore, the relevant concepts are borrowed by `sbeo` namespace.

Fig. 1 shows the Smart Building Evacuation Ontology (SBEO) in a nutshell. SBEO encompasses three main parts, (i) User model, for specifying characteristics and relations of buildings' occupants; (ii) Building model, for describing buildings topology and infrastructures installed in them; and (iii) Context model, for representing the dynamic changing state of buildings and occupants.

Fig. 2 shows the core concepts of Smart Building Evacuation Ontology (SBEO). In the following, we describe each of the models in more detail.

[10] http://xmlns.com/foaf/spec/
[11] https://www.w3.org/TR/vocab-ssn/
[12] http://purl.org/ontology/olo/20100723/orderedlistontology.html
[13] https://w3id.org/seas/BuildingOntology-1.0

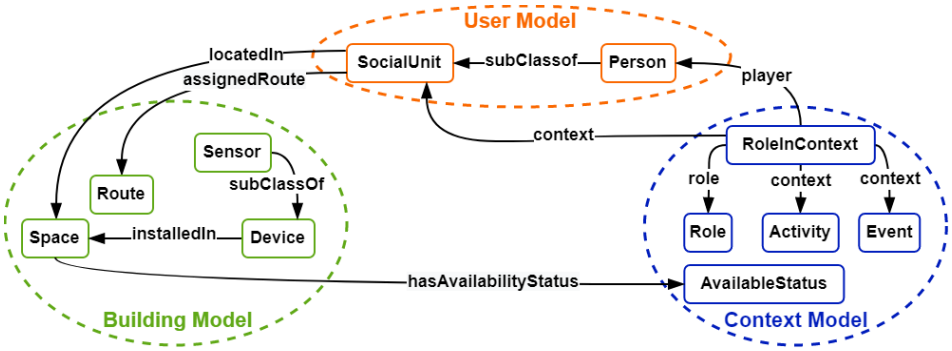


Fig. 1. Smart Building Evacuation Ontology (SBEO) in a nutshell

5.1. User Model

The User Model is used to represent the demographics, physical abilities, and people preferences (e.g. type of route or notification means). The demographics part includes the basic information about a person using object (`acquaintanceOf` and `responsibleTo`) and/ data properties (e.g. `foaf:firstName`, `foaf:lastName`, `foaf:gender`, `foaf:age`, and `id`).

A route or even a route element (e.g., space in the route) may not be appropriate for a specific person (or group of persons). Thus, it is crucial to model the physical abilities of individuals for personalized route selection. Ontologies like User Navigation Ontology (UNO)[26] and General User Model Ontology (GUMO)[21], provide a core knowledge base for users and their characteristics by modeling the abilities such as mental abilities, mobility capabilities, along with their quality. In the same way, the physical abilities of users are conceptualized in SBEO based on UNO and GUMO. Furthermore, we know that we can only describe a binary relation (i.e., either between two individuals or an individual and a value) in Semantic Web languages (e.g., RDF or OWL). As a solution, we may use n-ary design pattern to link an individual to more than one individual or even a value. A potential reader may consult [39] for further information. Thus, we also exploit n-ary relations to make use of the aforementioned concepts to associate them with a user. A new concept, `PersonAbility`, is also introduced to express the ability (using `hasAbility` property) of each person, together with its quality (using `hasQuality` property. Note that the `Ability` class is similar to UNO and GUMO. Under this parent class, other sub-classes are introduced to mention different types of abilities such as `MentalAbility`, `SpatialAbility`, `SensorAbility` and `MobilityAbility`.

In terms of navigational preferences, three relations–`hasNavigationalType`, `route Preference`, and `meansOfNotification`–are used. For example, `hasNavigati onalType` relation is used to express what type of navigation is provided to (or performed by) a person. The possible types of navigation can be `AssistedNavigation`, `AutonomousNavigation`, `Collborative Navigation` or `MultiObjective Navigation`. In `AssistedNavigation`, a person is assisted by another person or a machine to perform a specific activity. In `AutonomousNavigation`, a person plans and executes their path without any hu-

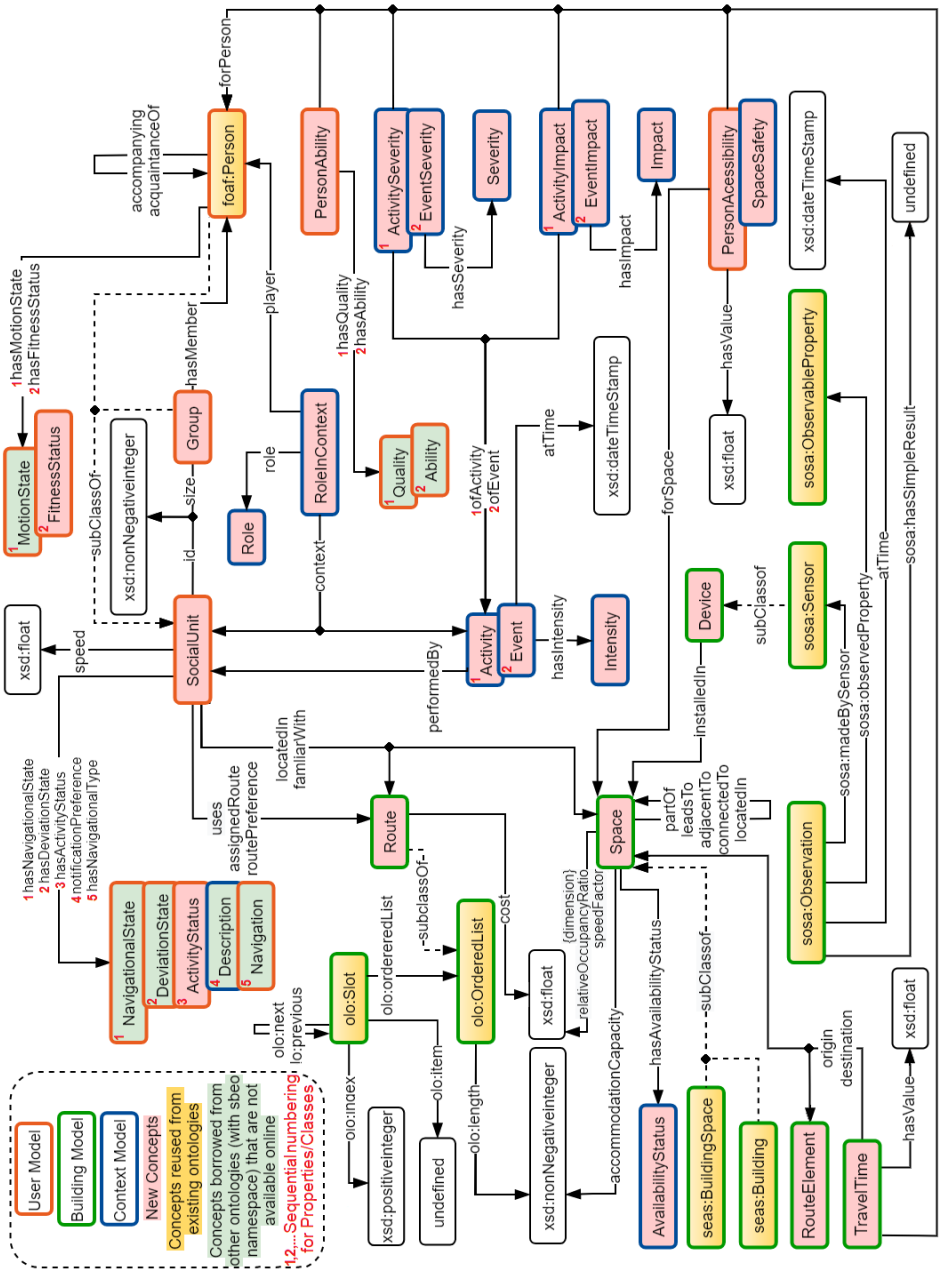


Fig. 2. Core concepts and relationships of Smart Building Evacuation Ontology (SBEO).

man or machine intervention. In `CollaborativeNavigation`, two or more persons are involved that may or may not have same objectives. Lastly, in `MultiObjective Navigation`, there can be various tasks to be done in it, such as visiting numerous points of interest, picking up multiple dependent persons.

Similarly, an individual may use `routePreference` relation to specify their route preference, such as shortest path and simplest path[14]. `meansOfNotification` property is used to choose the method for notifying a person about any piece of information related to space, route, activity, event, or any route element (e.g., door, stairs, waiting zone, assembly point, entrance, exit). An instance of user model is given in Fig. 3.

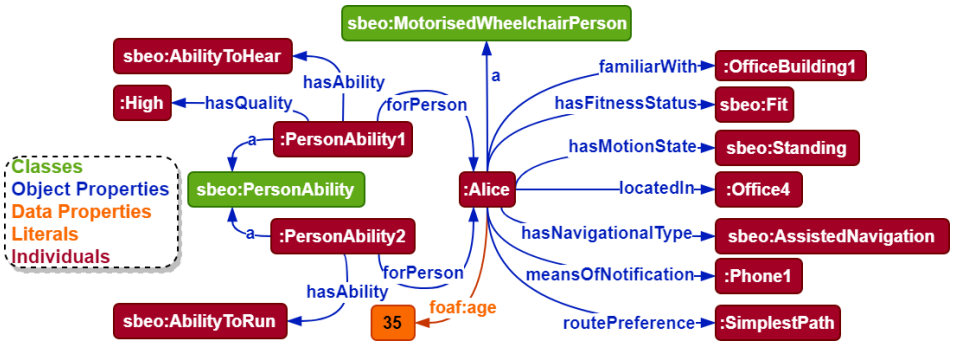


Fig. 3. User modelling

5.2. Building Model

In this model, concepts related to the geometry (or structure) of the building are described. Other spatial information about the building is also taken into account, such as sensors, fire safety equipment.

Geometrical elements. Geometrical elements are mentioned with the help of `Space` concept that represents any physical space. The type of a building and the specific site of any building can be described using `seas:Building` and `seas: SiteOf Building` respectively. All other atomic parts of a building (e.g., room, hall, door, stairs) are mentioned as the sub-classes of `seas:BuildingSpace`. These atomic elements use `locat edIn` property to mention where these are located in a specific building, whereas `partOf` property is use to mention which building or an atomic part of the building belongs to which other building. If any space is connected or adjacent to any other space, `connecte dTo` and `adjacentTo` properties are respectively used to express that relation between them. Similarly, as each space, e.g., `seas:Corridor`, `seas:Hall`, `seas:Escalat or`, has a specific shape, data properties such as `length`, `width`, `height`, `base`, `radius`, and `area`, are defined. Additionally, `accommodation Capacity` property is used to express the accommodation capacity of a space in terms of persons where as, another data property named `relativeOccupancyRatio` is introduced that states the ratio of occupied to total usable (`accommodationCapacity`) space. `Congestion` can also be expressed using a boolean property named as `hasCongestion`. An instance

of building geometry model is given in Fig. 4 that represents a Kids Area, along with the properties as mentioned in this paragraph.

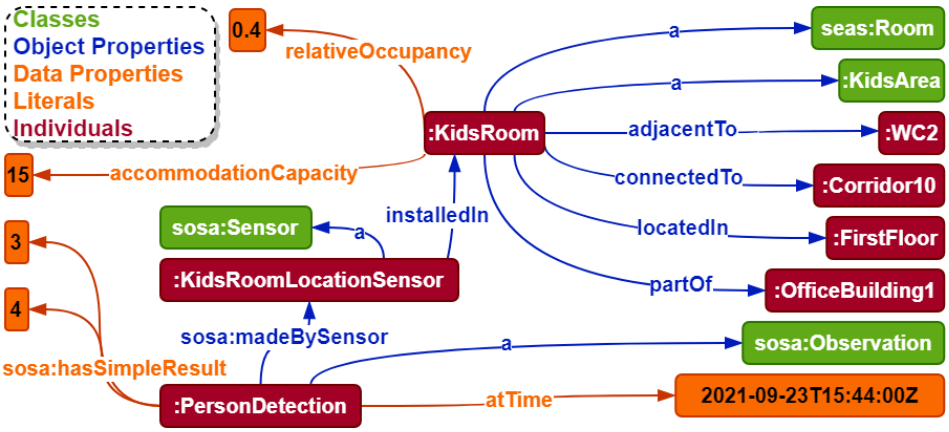


Fig. 4. Building space and sensor modelling

Routes. A route is sequence of connected spaces which is used to go from a starting point to a destination. We can also represent the geometry of a building as a graph, consisting of nodes and edges, such that it can be used for routing purposes. The existing approach in RDF vocabulary for specifying sequences (e.g., a route in this case), i.e. `rdf:list`, is not efficient because finding or accessing any specific element in the sequence is tiresome. To be precise, it doesn't allow to access an element with an index. As a solution, Ordered List Ontology (OLO) provides a simple data structure to express the ordered lists, that can also be used to represent the routes. Moreover, the elements of the routes can also be accessed easily.

In this regard, `Route` is conceptualized as a sub-class of `olo:OrderedList`, and `RouteElement` is introduced to represent the nodes and edges of a graph based on the information about the building structure. The edges and the nodes of a graph are represented with the help of `Passage` (e.g., corridor, door, elevator, stairs) and `RoutePoint` (e.g., entrance, exit, waiting zone, assembly point) concepts respectively. It is a choice of an ontology user how he/she wants to express the routes. For example, e.g., either using nodes or edges. In terms of usage,`Route` is allocated to any `SocialUnit` using `assignedRoute` relation.

In terms of travel time, `TravelTime` class is defined with the help of n-ary relation. In this class, the time (using `hasValue`) from one point to another point (using `origin` and `destination` properties respectively, e.g., `Room`, `Door`, `AssemblyPoint`, `Exit`) can be mentioned for a specific person (i.e., `forPerson`).

Elements other than the building structure. `Device` concept is used to express the elements that are not a part of building structure. It includes `IncidentProtectionDevice`, `Displayscreen`, `Telephone`, etc. In addition, some concepts and properties are reused from SOSA ontology [24], to express the `sosa:Sensor` and its values. In terms of relations, `installedIn` property is used to mention the location of the space

where a device is located (either permanently or movable), where as `uses` property is used to state who is using a specific device. An instance of a sensor is shown in Fig 4.

5.3. Context Model

The context model describes the concepts related to the situation of building and its occupants.

Activity and Event. By definition, activities are different from events. Because an activity is the happening that is being done by someone, for example visiting a museum, whereas an event is the happening of something, for example a fire in a museum. Due to this reason, `Event` and `Activity` concepts are stated separately. To cover the temporal dimension of them, `hasTimeDuration`, `endedAtTime`, and `startedAtTime` are used, respectively, to express the total time duration, ending time, and starting time, of any activity or event.

Some particular events are also defined in SBEO. For example, an `Incident` that expresses an unexpected event or occurrence that may result in property damage or may cause a serious injury or illness to people. Furthermore, it has also been classified in some evacuation-related concepts, such as `Congestion` and `Panic` (including `Stempeding`). In addition, activities may also be divided into different categories, such as `Navigation`, `EmergencyActivity`, `Visit`, whereas a social unit who is involved in a specific activity is linked with it using `performedBy` relation.

SBEO also conceptualizes intensity, severity, and the impact of an event or activity. As we know that the impact and the severity of an event or activity may differ for each person, we created n-ary relation to express these concepts in the ontology. On the other hand, the intensity of any event or activity remains the same for everyone. Thus, it is expressed using a class `Intensity`, along with a `hasIntensity` relation. Fig. 5 shows a fire event (i.e., :Fire1) and and evacuation activity (i.e., :Activity1), along with their intensity, severity, and impact.

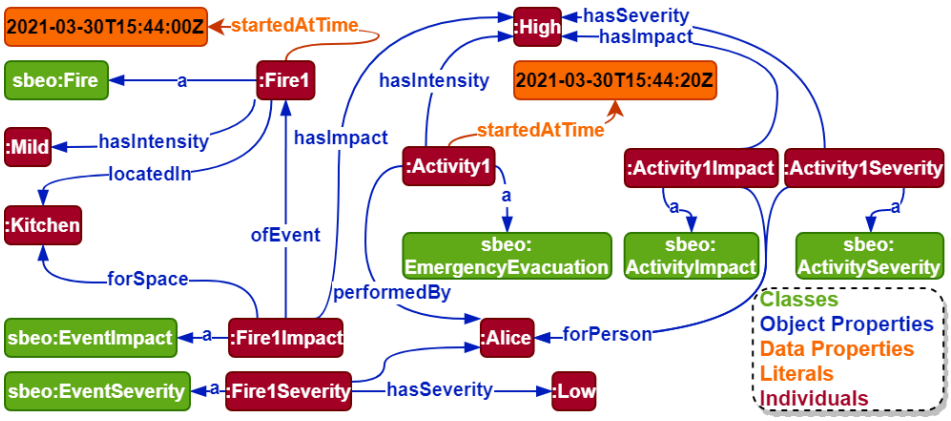


Fig. 5. Activity and event modelling

State and status of individuals. Various states and statuses of individuals related to their motion, navigation, fitness, and deviation, are also described in SBEO. For example, the state of their motion is expressed with the help of `motionState` property whose range can be either of the instances of `MotionState` class. These instances are explicitly enumerated as `Standing`, `Walking`, `Scooting`, `Running`, and `Rolling` (e.g., persons who use a manual wheelchair). The state of the navigation of the individuals using `hasNavigationState` whose range can be either of instances of `NavigationalState` class (i.e., `DeviatingFromPath` or `FollowingPath`). The deviation is further divided into multiple types using `hasDeviationState` relation, whose range can be one of the individuals of `DeviationState` class, i.e., `NoDeviate`, `RareDeviate`, `OftenDeviate` , or `TooOftenDeviate`.

In terms of status, `hasActivityStatus` relation is used to express the instantaneous information about an activity being performed by an individual whose range can be one of the instances of `ActivityStatus` class, for example `Evacuating`, `Evacuated`, `Visiting`, `PickingUpDependents`. Furthermore, `hasFitnessStatus` property states the fitness status of a person that can be either `Fit`, `Exhausted` or `Injured`.

Group and role in a context. Two or more persons can be expressed as a `Group` if they are involved in any common activity (e.g, `Evacuation`, `Visiting`, `PickingUpDep endents`), having family- or friend- ties or an acquaintance, or sharing a common space (e.g., located in the same building, room or building floor). In addition, `hasMember` and `size` properties state the members and the size of a group respectively. If a person becomes responsible or a leader of a social unit (e.g, person or even a group), he/she can also be expressed with the help of `responsibleTo` property.

In terms of role, `RoleInContext` concept is introduced based on n-ary design pattern. It consists of three properties named `role`, `player`, and `context`, to express the information about a role of a person, the person identity, and a context (e.g., `Activity`, `Event` or `SocialUnit`) in which a person is playing that role, respectively.

Space safety and accessibility. The safety of a particular space tells us how safe the space is, for a specific person (or types of persons). On the contrary, the accessibility of a space tells us how accessible the space is, for a specific person (or types of persons). Due to this reason, as the safety and the accessibility of a particular space may differ from one person (or types of persons) to another, two different parameters are introduced; `SpaceSafety` and `PersonAccessibility`. Both of these concepts are linked with the specific space using `ofSpace` property whose safety/accessibility is required to mention, while `hasValue` and `forPerson` properties are used to express the safety/accessibility value of that particular space and the relevant persons associated with that value respectively. These concepts can be seen in Fig. 6, along with their usage. Note that the range of `hasValue` property of each parameter is taken as a fraction because it is a choice of the ontology user that how he or she wants to exploit it in a specific application.

In the context model, some other information related to building spaces is also described using various properties. It is as follows:

1. `relativeOccupancyRatio` - expresses the ratio of occupied to total usable (i.e., capacity) space.

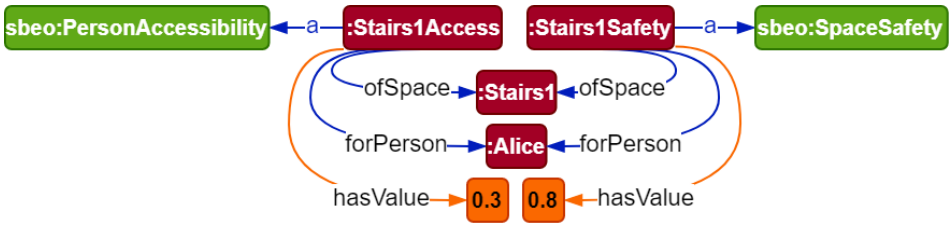


Fig. 6. Space safety and accessibility modelling

2. `accompanying` - a relation to mention who is accompanying whom in a particular space.
3. `speedFactor` - a value bound to any space (if applicable) that may affect the speed of individuals while passing through it. By default, it is equal to unity, but can be changed depending on various factors, such as `Congestion`, `relativeOccupancyRatio`.
4. `hasAvailabilitystatus` - states the availability status of any device, space or route as one of the instances of `AvailabilityStatus` class (i.e., `Available` or `UnAvailable`).
5. `excludedFor` - mentions any specific space that is not preferred (or incapable of accessing) by a person.

Potential Inferences. Inference in Semantic Web is a method of discovering new relationships between resources based on the existing data from the vocabulary. In this regard, some relationships are also inferred in SBEO based on the existing relationships among the individuals (instantiation) of the concepts. These are as follows:

- **Functional:** `hasAbility, hasAvailabilityStatus, hasDeviationState, hasFitnessStatus, hasImpact, hasIntensity, hasSeverity, hasValue, hasQuality, hasMotionState, hasNavigationalState, foaf:age, foaf:gender, accommodationCapacity, relativeOccupancyRatio, hasCongestion, startedAtTime, endedAtTime, hasXTimesDeviated, area, base, height, length, radius, size, speed, width, olo:ordered_list, olo:next, destination, origin, upper, lower`
- **Inverse Functional:** `olo:previous, upper, lower`
- **Transitive:** `accompanying, installedIn, leadsTo, partOf`
- **Symmetric:** `leadsTo, accompanying, acquaintanceOf, adjacentTo, connectedTo`
- **Asymmetric:** `responsibleTo, locatedIn`
- **Reflexive:** `acquaintanceOf`
- **Disjointness:** `adjacentTo and connectedTo, endedAtTime and startedAtTime`
- **Inverse:** `lower and upper, olo:next and olo:previous, olo:ordered_list and olo:slot`

6. Evaluation

Turchet et al. [59] mentioned that ontology designing is somehow a matter of subjectivity similar to the implementation of an algorithm, which is an interpretation of the computer programmer. Hlomani and Stacey [22] discussed several approaches, methods and metrics to evaluate an ontology. They found out that there are two major perspectives which are needed to evaluate any ontology; quality and correctness. Accordingly, SBEO is evaluated using various formal methods and approaches, and metrics, to find out its quality and correctness.

Ontology Metrics. Fernandez et al. [17] proposed twelve different measures to evaluate the ontology in terms of its generality and performance. In this study, we have shortlisted some of these metrics that fit the scope of SBEO. These metrics give an insight to the potential user of the ontology in terms of concepts and their relationships, popularity (i.e., current usage), and reliability (or availability).

Table 2. A comparison of SBEO with other ontologies in the field using 'Knowledge coverage and popularity measures' proposed by Fernandez et al. [17]

Ontology	No. of classes	No. of properties		No. of individuals	Direct popularity	Indirect popularity		
		Data	Object			Ontology Imports (Direct, Indirect)	Classes	Properties
SBEO	191	31	52	33	low	0, 0	21%	18%
SEAS (Building)	102	3	32	5	low	1, 8	34%	85%
SOSA	16	2	21	1	high	0, 0	0%	0%
SSN	23	2	36	2	very high	1, 1	21%	58%
empathi	237	98	171	10	low	9, 0	31%	98%
BOT	10	1	16	5	medium	0, 0	30%	0%

Table 2 shows a comparison of these ontology metrics of SBEO with other ontologies in the field and which are cited in the related works section. In the table, the number of properties is further divided into two sub-columns; object properties and data properties. As for the proposed ontology, there are 191 classes and 83 properties (both object and data) described in it in which 40 classes are reused from other ontologies, and the remaining 151 classes are created from scratch. The term direct popularity means how many existing ontologies are importing the given ontology, whereas inverse popularity [59] means how many concepts and properties are imported from existing ontologies to develop the given ontology. In this regard, as SBEO has been developed recently, its direct popularity is low. On the other hand, in terms of indirect popularity, concepts and properties from various four existing ontologies (i.e., seas, olo, sosa, foaf) are used in sbeo.

Oops! Pitfall Scanner. We have evaluated SBEO using a tool named Oops! Pitfall Scanner [43] that assesses an ontology qualitatively by checking its quality across three various dimensions, namely: structural, functional and usability-profiling. In addition, it also examines the consistency, completeness and conciseness of an ontology.

Among 41 pitfalls (i.e., checking points), 3 minor (i.e., P08, P13,and P22) and 2 important (i.e., P11 and P30) pitfalls have been identified due to: (1) missing annotations; (2) the absence of inverse relationships; (3) naming convention other than CamelCase; (4) missing domain/range; (5) some concepts seem equivalent. As regards, first, third and fourth points, depends on the concepts we have reused from the existing ontologies (i.e., SEAS, SOSA/SSN, OLO, FOAF) in SBEO. For the second point, most of the properties are either n-ary relations or do not support an adequate converse term, therefore these are exempted from this rule [62]. The justification of fifth point is, all of these concepts have different meanings in the proposed ontology, hence they will be kept in their current form.

The results from this tool imply that the quality of the proposed ontology meets the best practices. Consequently, critical problems related to modelling and reasoning might be avoided, such as logical inconsistencies or undesired inferences.

Reasoners to find any inconsistency. Three different reasoners—FaCT++ (version 1.6.5) [57], Pellet (version 2.2.0) [53], HermiT (version 1.4.3.456) [51]—have been used to check the logical consistency of SBEO, and no inconsistencies have been found in it. It implies that the SBEO classes may have instances (OWL individuals), and useful knowledge can be inferred from it.

Answering Competency Questions (CQs). The competency questions (CQs) are an important part to evaluate an ontology. In this regard, SPARQL-based queries are used to answer the competency questions stated in section 4. Due to the space issue, the answer to each CQ can be found here[14].

MIRO: Minimum Information for the Reporting of an Ontology Matentzoglu et al. in [36], defined some guidelines named Minimum Information for Reporting an Ontology (MIRO). According to them, MIRO guidelines provide a better level of completeness and consistency to an ontology documentation. Hence, SBEO is described using MIRO guidelines. The report[15] can be found on Github repository.

Task-based Evaluation - A Use-case Task-based evaluation is one of the methods to evaluate an ontology by measuring the quality of the results a specific application delivers [48]. In this regard, a simple scenario is described where SBEO is used to define the semantics for a smart building evacuation system. Due to the lack of the space the use-case of the scenario[16] can be found on Github repository.

7. Application example: CAREE

In this section, we describe a Context-Aware Emergency Evacuation (CAREE)[17] system, which uses SBEO ontology for knowledge representation. CAREE uses complex event processing and semantic stream reasoning technologies for analysing streams of data coming from sensors installed in a smart building, identifying emergency conditions (e.g. hazardous situations that can be dangerous for the safety of the occupants of the building) and proposing safe and efficient individual evacuation routes to the occupants

[14] https://github.com/qasimkhalid/SBEO/blob/master/Competency%20Questions.md

[15] https://github.com/qasimkhalid/SBEO/blob/master/MIRO%20Evaluation.md

[16] https://github.com/qasimkhalid/SBEO/blob/master/Examples/SmallOfficeSpace/Documentation/SBEO_TaskBasedEvaluation_SmallOfficeScenario.docx

[17] https://github.com/qasimkhalid/CAREE

of the building according to the context (status of the building and people's characteristics).

Fig. 7 shows a block diagram of CAREE architecture. The raw data from sensors (raw events) are annotated using SBEO and organised in several data streams according to their type (e.g., locations of people, temperature, humidity, and smoke).

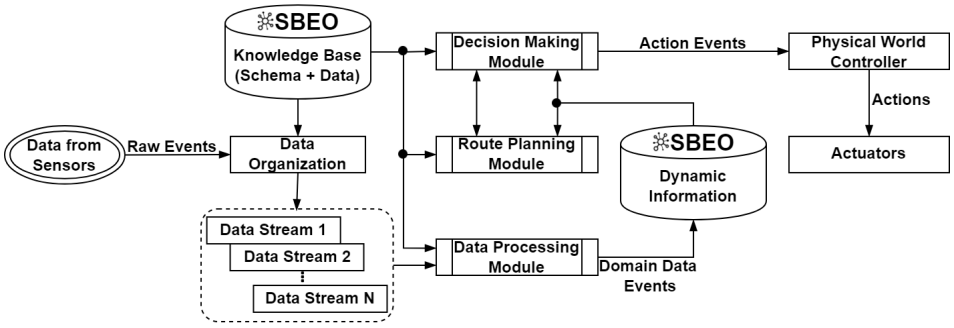


Fig. 7. The architecture of Context-Aware Emergency Evacuation (CAREE) software

The *Data Processing Module* aims to generate contextual information by processing data streams and static knowledge (i.e., building topology, user information). We use C-SPARQL[6]), an engine for processing continuous streams of RDF data. C-SPARQL[18] queries are attached to specific data streams to identify patterns in the data and generate pieces of contextual information (e.g. movement of people, fire detection, etc.). The contextual information is generated using SBEO ontology and is stored in an RDF repository on a real-time basis. For example, if *Sensor1* detects *PersonA*, and *Sensor1* is installed in *Office1* then the triple (*PersonA sbeo:locatedIn Office1*) is added to the context repository.

The *Decision Making Module* processes the information from Domain Data Events and the knowledge base running SPARQL queries. If a building evacuation is needed, it communicates with the *Route Planning Module* to get the optimal routes for each person. The *Route Planning Module* calculates available, safe and accessible routes to the persons depending on their physical characteristics and preferences, as well as the instantaneous situation of the building. Lastly, the *Decision Making Module* generates relevant Action Events according to the predefined criteria.

The actions events are then fed to the *Physical World Controller* such that specific actions could be performed as remedies to these events, such as assigning routes to persons, making hazardous spaces unavailable, and informing persons about the Points of Interest. The Physical World Controller works as a bridge between the system and the physical world (Actuators, i.e. IoT devices).

We have developed an agent-based simulated environment to test CAREE, where each person is considered a separate agent in a common and shared environment. We used Java and SPARQL languages for its development. Also, we have exploited Apache Jena and

[18] C-SPARQL language is a variation of the SPARQL query language for RDF, including stream processing characteristics such as windows and continuous processing

C-SPARQL frameworks to extract and update the SBEO-based data model. The simulator replicates the free-flow movements of people between two nodes that share a common arc and generates the values of Temperature, Smoke, and Humidity sensors, in a custom format. These simulated values are then fed into CAREE in the form of data streams after a customized time interval (e.g., one second). This simulated environment is deterministic in nature, and a scheduler is used to carry out the movement of each person in the building that gets updated after every timestep (e.g., one second). As soon as, any hazard is detected and the evacuation process is set off, the evacuation route (i.e., a path from a person's location to the nearest and feasible exit) is calculated in terms of timesteps and updated in the scheduler. Later on, the scheduler simulates the movement of the persons on each timestep until they reach their destination (i.e., exit). Once, a person reaches his/her destination, he/she is eliminated from the scheduler. Listing 1 shows a snapshot of the output of the simulator. This is updated after every timestep.

```
//Edge
(node#, node#) | cost | safety value | capacity
(node1, node2)    10        0.5           2
(node1, node3)    15        0.3           3
(node2, node3)    20        0.1           1

//Node
node# | safety value | capacity |  No. of persons positioned at a node
node1       0.0           14                    2
node2       1.0           10                    1
node3       0.4           10                    0
node4       0.6           16                    3

//Person
person# | location of a person
person1         node1
person2         node2
person3         node1

//Inaccessible edges list
person# | list of edges that are not apt for evacuation
person1             {} //empty set
person2             {(node1, node3)}
person3             {(node2, node3)}
```

Listing 1. Simulator output after every timestep.

According to the scope of the paper, we ran a simple scenario of a building floor in our simulated environment, as shown in Fig. 8. Each entity, such as space, floor exit, and fire extinguisher, is represented using Smart Building Evacuation Ontology (SBEO). Also, specific attributes of spaces, such as accommodation capacity, connections with other spaces, and the distance between the connected spaces (e.g., cost of each Origin-Destination (O-D) pair), are expressed.

In addition, the building floor is further represented as a graph **G** = (**N**, **A**), as seen in Fig. 9, with 17 nodes, where each node in **N** represents an entity shown in Fig. 8, e.g., closed space, junction[19], point of interest, or entrance or emergency exit. On the other hand, **A** represents the arcs between the connected nodes. We also assume that each node, as seen in Fig. 8 with a diamond symbol, is equipped with four types–Temperature, Smoke, Humidity, and Human Detection–of sensors and modelled using SBEO. In the end, we also modelled ten persons (including their demographics and physical characteristics) using SBEO, in which two of them are mobility impaired.

[19] A junction is an imaginary route element that connects multiple corridors or other route elements (i.e., nodes).

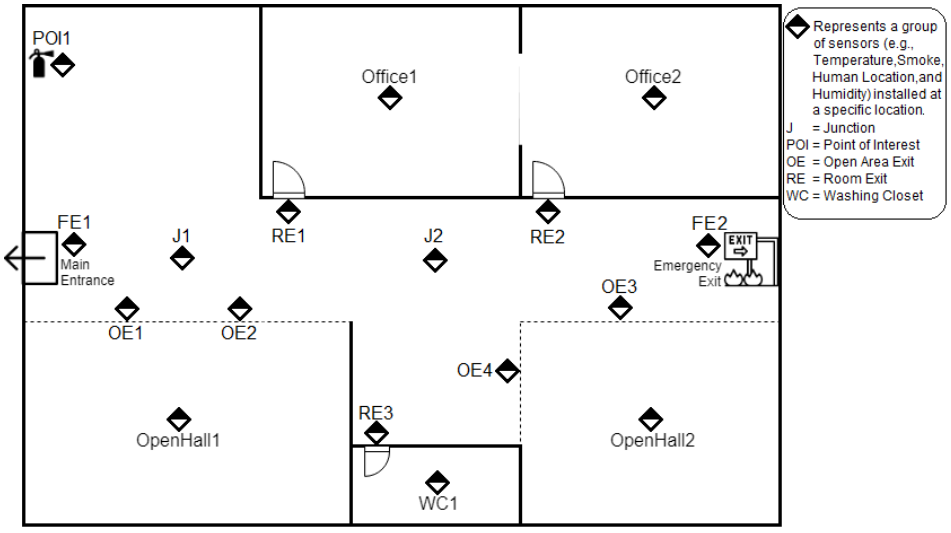


Fig. 8. A building floor plan with an entrance (which may also be an exit), an emergency exit and some closed spaces

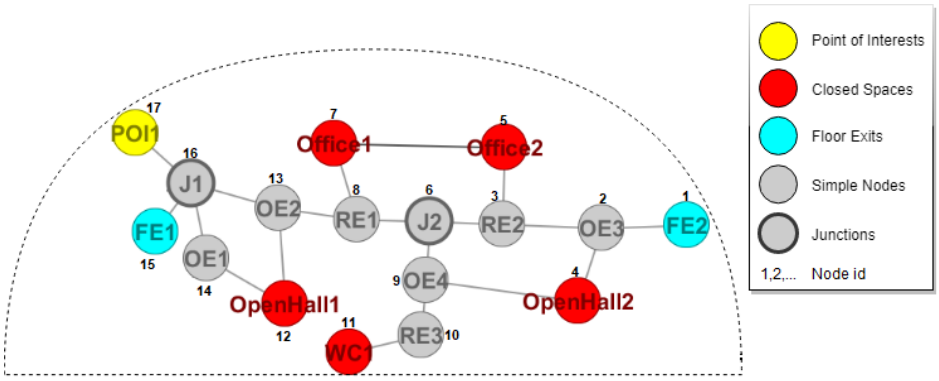


Fig. 9. Network modelling from a smart building floor plan. Nodes are labelled as names and Ids, and Arcs between two connected nodes are expressed as lines.

The access to a specific space is determined with the help of a `sbeo:hasSafetyValue` in SBEO. For this particular application scenario, it ranges between 0 and 1, which express the minimum and maximum safety, respectively. During the usual conditions, the safety of all spaces is 1, as the temperature and humidity are equal to 25 degrees Centigrade and 40%, respectively. On the other hand, we assume that the critical safety for both arcs and nodes is 0.5. Thus, the space whose safety is less than 0.5 is not apt for evacuation for mobility-impaired persons, whereas if it is equal to 0 is not apt for evacuation for anyone.

For the sake of simulation purposes, a fire event is triggered if the following conditions are met for a particular space altogether:

1. The temperature rises to 60 degrees Centigrade.
2. Humidity is less than 20%.
3. Smoke exists.

Here, we describe the results of the simulation-based experimental setup mentioned above. Initially, as we described earlier, at timestep t_0, the temperature ($temp$) of each space was 25 degrees Centigrade, the humidity (h) was 40%, and the smoke (s) was not detected. After each time interval (i.e., one second in this experiment), the temperature value of each sensor was randomly updated within the range of $temp_{t_{x-1}} + 5$ and $temp_{t_{x-1}} - 2$, where x is an integer that increases after each timestep, t. Based on the temperature value of a sensor, the humidity and safety values of the same sensor are also updated. For example, at timestep t_4, *Office1* (i.e., Node 7) had a safety value of 0.87, and one person was in it. Similarly, the corridor (i.e., arc) between *Office1* and *Office2* (i.e., (Node 7, Node 5)) had a safety value of 0.88. It implies that every person can access *Office1* and the corridor between *Office1* and *Office2*. Furthermore, *Person6* is located in *Office1* (i.e., Node 7).

Suddenly, at timestep t_{18}, fire event is detected on the arcs between *POI1* and *J1* (i.e., (Node 17, Node 16)) and *OpenHall2* and *OE4* (i.e., (Node 4, Node 9)). Subsequently, their safety values are also reduced to 0.44 and 0.47. As a result, the Decision Making Module updates the safety of these arcs not to be apt for evacuation to mobility-impaired people and sets off the evacuation process.

Once the evacuation process starts, the details of accessible space nodes depending on the allowed safety values concerning the type of each evacuee, along with the location of each person, are sent to the Route Planning Module. This module calculates feasible, and shortest paths using Dijkstra's Algorithm [13] for each person to evacuate the building by reaching either of the exits (i.e., *FE1* and *FE2*) from their current locations. We assume that one unit cost equals one timestep. For example, if a cost of an arc between two nodes is five units, it takes five timesteps to traverse that arc. Thus, the total cost of a path is equal to the cumulative cost of all the arcs involved. In this regard, each person evacuated the building (i.e., reached one of the safe exits) corresponding to the time equal to the cost of the complete route found and assigned to them by the algorithm.

8. Conclusion and Future Work

In this paper, a light-weight, but comprehensive, ontology was proposed for route recommendation in smart buildings during both normal and emergency conditions. The proposed data model provides the concepts and relationships for an efficient route planning in

smart buildings. It includes the information about users, buildings and the context awareness.

The creation of the ontology is motivated by the need for facilitating the interoperability of smart gadgets and IoT-enabled buildings. The ontology is developed using a well-known methodology (i.e., METHONTOLOGY), and design patterns recommended by W3C. Furthermore, the ontology was evaluated using various metrics and methodologies, found consistent, and considered applicable in its domain. The proposed ontology is compatible and integrated with some popular ontologies such as SOSA, FOAF, SEAS, etc.

As a future work, we plan to integrate SBEO with the digital twin of a smart building in order to test its applicability and reliability. Afterwards, we will discuss the acquired results with the emergency response officers such that we might compare these results with the real data captured by them. That will allow us to evolve and evaluate the ontology based on the potentially expected real-world use-cases.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Science, Innovation, and Universities, co-funded by EU FEDER Funds, through grant RTI2018-095390-B-C31/32/33 (MCIU/AEI/ FEDER, UE), and by the AGROBOTS Project of the Rey Juan Carlos University funded by the Community of Madrid, Spain.

References

1. Akinwande, O.J., Bi, H., Gelenbe, E.: Managing crowds in hazards with dynamic grouping. IEEE Access 3, 1060–1070 (2015)
2. Al-Nabhan, N., Al-Aboody, N., Al Islam, A.A.: A hybrid iot-based approach for emergency evacuation. Computer Networks 155, 87–97 (2019)
3. Alirezaie, M., Hammar, K., Blomqvist, E.: Smartenv as a network of ontology patterns. Semantic Web 9(6), 903–918 (2018)
4. Anagnostopoulos, C., Tsetsos, V., Kikiras, P., et al.: Ontonav: A semantic indoor navigation system. In: 1st Workshop on Semantics in Mob. Env. (SME05), Cyprus (2005)
5. Augusto, J.C., Liu, J., Chen, L.: Using ambient intelligence for disaster management. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. pp. 171–178. Springer (2006)
6. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-sparql: Sparql for continuous querying. In: Proceedings of the 18th international conference on World wide web. pp. 1061–1062 (2009)
7. Billhardt, H., Dunkel, J., Fernández, A., Lujak, M., Hermoso, R., Ossowski, S.: A proposal for situation-aware evacuation guidance based on semantic technologies. In: Multi-agent Systems and Agreement Technologies, pp. 493–508. Springer (2016)
8. Bitencourt, K., Durão, F.A., Mendonça, M., SANTANA, L.L.B.D.S.: An ontological model for fire emergency situations. IEICE Trans. on Inf. and Sys. 101(1), 108–115 (2018)
9. Blache, F., Chraiet, N., Daroux, O., Evennou, F., Flury, T., Privat, G., Viboud, J.P.: Position-based interaction for indoor ambient intelligence environments. In: Aarts, E., Collier, R.W., van Loenen, E., de Ruyter, B. (eds.) Ambient Intelligence. pp. 192–207. Springer (2003)
10. Boje, C., Li, H.: Crowd simulation-based knowledge mining supporting building evacuation design. Advanced Engineering Informatics 37, 103–118 (2018)
11. Chu, M.L., Parigi, P., Law, K.H., Latombe, J.C.: Simulating individual, group, and crowd behaviors in building egress. Simulation 91(9), 825–845 (2015)

12. De Nicola, A., Melchiori, M., Villani, M.L.: Creative design of emergency management scenarios driven by semantics: An application to smart cities. Inf. Sys. 81, 21–48 (2019)
13. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. Numerische mathematik 1(1), 269–271 (1959)
14. Duckham, M., Kulik, L.: "simplest" paths: automated route selection for navigation. In: International Conference on Spatial Information Theory. pp. 169–185. Springer (2003)
15. Dudas, P.M., Ghafourian, M., Karimi, H.A.: Onalin: Ontology and algorithm for indoor routing. In: 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware. pp. 720–725. IEEE (2009)
16. Fang, Z., Song, W., Zhang, J., Wu, H.: Experiment and modeling of exit-selecting behaviors during a building evacuation. Physica A: Stat. Mech. and its Appl. 389(4), 815–824 (2010)
17. Fernández, M., Overbeeke, C., Sabou, M., Motta, E.: What makes a good ontology? a case-study in fine-grained knowledge reuse. In: Asian Semantic Web Conference. pp. 61–75. Springer (2009)
18. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. American Asociation for Artificial Intelligence (1997)
19. Gaur, M., Shekarpour, S., Gyrard, A., Sheth, A.: Empathi: An ontology for emergency managing and planning about hazard crisis. In: 2019 IEEE 13th International Conference on Semantic Computing (ICSC). pp. 396–403 (2019)
20. Haghighi, P.D., Burstein, F., Zaslavsky, A., Arbon, P.: Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. Decision Support Systems 54(2), 1192–1204 (2013)
21. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo – the general user model ontology. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) User Modeling 2005. pp. 428–432. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
22. Hlomani, H., Stacey, D.: Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. Semantic Web Journal 1(5), 1–11 (2014)
23. Huang, H., Gartner, G.: A survey of mobile indoor navigation systems. In: Cartography in Central and Eastern Europe, pp. 305–319. Springer (2009)
24. Janowicz, K., Haller, A., Cox, S.J., Le Phuoc, D., Lefrançois, M.: Sosa: A lightweight ontology for sensors, observations, samples, and actuators. Journal of Web Semantics 56, 1–10 (2019)
25. Karimi, H.A., Ghafourian, M.: Indoor routing for individuals with special needs and preferences. Transactions in GIS 14(3), 299–329 (2010)
26. Kikiras, P., Tsetsos, V., Hadjiefthymiades, S.: Ontology-based user modeling for pedestrian navigation systems. In: ECAI 2006 Workshop on Ubiquitous User Modeling (UbiqUM), Riva del Garda, Italy. pp. 1–6 (2006)
27. Krieg-Brückner, B., Frese, U., Lüttich, K., Mandel, C., Mossakowski, T., Ross, R.J.: Specification of an ontology for route graphs. In: International Conference on Spatial Cognition. pp. 390–412. Springer (2004)
28. Kritsotakis, M., Michou, M., Nikoloudakis, E., Bikakis, A., Patkos, T., Antoniou, G., Plexlousakis, D.: Design and implementation of a semantics-based contextual navigation guide for indoor environments. J. of Amb. Int. and Smart Environments 1(3), 261–285 (2009)
29. Lefrançois, M., Kalaoja, J., Ghariani, T., Zimmermann, A.: SEAS Knowledge Model. Deliverable 2.2, ITEA2 12004 Smart Energy Aware Systems (2016), 76 p.
30. Li, X., Liu, G., Ling, A., Zhan, J., An, N., Li, L., Sha, Y.: Building a practical ontology for emergency response systems. In: Computer Science and Software Engineering, International Conference on. vol. 4, pp. 222–225. IEEE Computer Society (2008)
31. Lujak, M., Billhardt, H., Dunkel, J., Fernández, A., Hermoso, R., Ossowski, S.: A distributed architecture for real-time evacuation guidance in large smart buildings. Computer Science and Information Systems 14(1), 257–282 (2017)
32. Lujak, M., Giordani, S.: Centrality measures for evacuation: finding agile evacuation routes. Future Generation Computer Systems 83, 401–412 (2018)

33. Lujak, M., Ossowski, S.: Intelligent people flow coordination in smart spaces. In: Multi-Agent Systems and Agreement Technologies, pp. 34–49. Springer (2015)
34. Ma, Y., Li, L., Zhang, H., Chen, T.: Experimental study on small group behavior and crowd dynamics in a tall office building evacuation. Physica A: Statistical Mechanics and its Applications 473, 488–500 (2017)
35. Malizia, A., Onorati, T., Diaz, P., Aedo, I., Astorga-Paliza, F.: Sema4a: An ontology for emergency notification systems accessibility. Exp. Sys. with App. 37(4), 3380 – 3391 (2010)
36. Matentzoglu, N., Malone, J., Mungall, C., Stevens, R.: Miro: guidelines for minimum information for the reporting of an ontology. Journal of biomedical semantics 9(1), 6 (2018)
37. Morales, A., Alcarria, R., Martin, D., Robles, T.: Enhancing evacuation plans with a situation awareness system based on end-user knowledge provision. Sensors 14(6), 11153–11178 (2014)
38. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: A guide to creating your first ontology (2001)
39. Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-ary Relations on the Semantic Web, W3C Working Group Note, 12 April 2006. https://www.w3.org/TR/swbp-n-aryRelations/, [Online; accessed January 30, 2023]
40. Onorati, T., Malizia, A., Diaz, P., Aedo, I.: Modeling an ontology on accessible evacuation routes for emergencies. Expert Sys. with Appl. 41(16), 7124–7134 (2014)
41. Pâslaru-Bontaş, E.: A contextual approach to ontology reuse: methodology, methods and tools for the semantic web. PhD Thesis, Universität Berlin, Germany (2007)
42. Pinto, H.S., Staab, S., Tempich, C.: Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In: Proceedings of the 16th European Conference on Artificial Intelligence. pp. 393–397 (2004)
43. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems (IJSWIS) 10(2), 7–34 (2014)
44. Ramos, C., Augusto, J.C., Shapiro, D.: Ambient intelligence—the next step for artificial intelligence. IEEE Intelligent Systems 23(2), 15–18 (2008)
45. Rasmussen, M.H., Lefrançois, M., Schneider, G.F., Pauwels, P.: Bot: the building topology ontology of the w3c linked building data group. Semantic Web 12(1), 143–161 (2021)
46. Ray, B.: How An Indoor Positioning System Works, AirFinder. https://www.airfinder.com/blog/indoor-positioning-system (2018), [Online; accessed January 30, 2023]
47. Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., Van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. In: Euro. Sem. Web Conf. pp. 752–767. Springer (2014)
48. Sabou, M., Gracia, J., Angeletou, S., d'Aquin, M., Motta, E.: Evaluating the semantic web: A task-based approach. In: The Semantic Web. pp. 423–437. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
49. Santos, L.S., Sicilia, M.A., Garcia-Barriocanal, E.: Ontology-based modeling of effect-based knowledge in disaster response. Int. J. on Semantic Web and Information Systems (IJSWIS) 15(1), 102–118 (2019)
50. Segev, A., et al.: Context ontology for humanitarian assistance in crisis response. In: ISCRAM 2013 Conference Proceedings – 10th International Conference on Information Systems for Crisis Response and Management. pp. 526–535. ISCRAM (2013)
51. Shearer, R., Motik, B., Horrocks, I.: Hermit: A highly-efficient owl reasoner. In: Owled. vol. 432, p. 91 (2008)
52. Sicilia, M.Á., Santos, L.: Main elements of a basic ontology of infrastructure interdependency for the assessment of incidents. In: Visioning and Engineering the Knowledge Society. A Web Science Perspective. pp. 533–542. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
53. Sirin, E., Parsia, B.: Pellet: An owl dl reasoner. In: Proc. of the 2004 Description Logic Workshop (DL 2004). pp. 212–213 (2004)

54. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The neon methodology for ontology engineering. In: Ontology engineering in a networked world, pp. 9–34. Springer (2012)
55. Sumam, M.I., Vani, K.: Agent based evacuation simulation using leader-follower model. International Journal of Scientific & Engineering Research (IJSER) 4(8) (2013)
56. Sure, Y., Staab, S., Studer, R.: On-to-knowledge methodology (otkm). In: Handbook on ontologies, pp. 117–132. Springer (2004)
57. Tsarkov, D., Horrocks, I.: Fact++ description logic reasoner: System description. In: International joint conference on automated reasoning. pp. 292–297. Springer (2006)
58. Tsetsos, V., Anagnostopoulos, C., Kikiras, P., Hadjiefthymiades, S.: Semantically enriched navigation for indoor environments. Int. J. of Web and Grid Services 2(4), 453–478 (2006)
59. Turchet, L., Antoniazzi, F., Viola, F., Giunchiglia, F., Fazekas, G.: The internet of musical things ontology. Journal of Web Semantics 60, 100548 (2020)
60. Uschold, M., Gruninger, M.: Ontologies: principles, methods and applications. The Knowledge Engineering Review 11(2), 93–136 (1996)
61. van Heijst, G., Schreiber, A., Wielinga, B.: Using explicit ontologies in kbs development. International Journal of Human-Computer Studies 46(2), 183 – 292 (1997)
62. Villalón, M.P., Pérez, A.G.: Ontology Evaluation: a pitfall-based approach to ontology diagnosis. PhD Thesis, Universidad Politecnica de Madrid, Escuela Tecnica Superior de Ingenieros Informaticos (2016)
63. Wang, X., Dong, J., Chin, C., Hettiarachchi, S., Zhang, D.: Semantic space: an infrastructure for smart spaces. IEEE Pervasive Computing 3(3), 32–39 (2004)
64. Yang, L., Worboys, M.: A navigation ontology for outdoor-indoor space: (work-in-progress). In: Proceedings of the 3rd ACM SIGSPATIAL international workshop on indoor spatial awareness. pp. 31–34 (2011)
65. Yusupov, R., Ronzhin, A.: From smart devices to smart space. Herald of the Russian Academy of Sciences 80(1), 63–68 (2010)

Qasim Khalid is currently a Ph.D. candidate at University Rey Juan Carlos in Madrid (Spain). He is associated with the CETINIA Research Centre. His research interests include multi-agent systems, Complex Event Processing, Semantic Web, Ontology Engineering, and sustainable energy systems.

Alberto Fernández is a Professor at University Rey Juan Carlos in Madrid (Spain), where he is a member of the CETINIA Research Centre. His main research lines are multi-agent systems, knowledge representation, semantic technologies, and open systems.

Marin Lujak is a Distinguished Researcher Lecturer at the CETINIA Research Centre of the University Rey Juan Carlos in Madrid (Spain), as part of the Beatriz Galindo excellence initiative of the Spanish Ministry of Universities. His scientific work relates to distributed and decentralized multi-agent coordination, optimization, and decision-making approaches for large and complex systems.

Arnaud Doniec is a Professor at IMT Nord Europe in Douai (France) and member of Centre of research for Digital Systems. His research work is related to multi-agent systems, behavioural simulation, and constraint-based models.

Received: January 18, 2022; Accepted: August 20, 2022.

Computer Science and Information Systems 20(1):77–93 https://doi.org/10.2298/CSIS220303047X

SEE-3D: Sentiment-driven Emotion-Cause Pair Extraction Based on 3D-CNN*

Xin Xu[1,2], Guangli Zhu[1,2†], Houyue Wu[1,2], Shunxiang Zhang[1,2], and Kuan-Ching Li[3]

[1] School of Computer Science and Engineering,
Anhui University of Science & Technology, 232001 Huainan, China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center,
WangJiang Road 5089, Hefei, 230088, Anhui, China
xuxin.onepiece@gmail.com
glzhu@aust.edu.cn
why3664@163.com
sxzhang@aust.edu.cn
[3] Department of Computer Science and Information Engineering (CSIE),
Providence University, 43301 TaichungTaichung, Taiwan
kuancli@pu.edu.tw

Abstract. As an emotional cause detection task, Emotion-Cause Pair Extraction (ECPE) provides technical support for intelligent psychological counseling, empty-nest elderly care, and other fields. Current approaches mainly focus on extracting by recognizing causal relationships between clauses. Different from these existing methods, this paper further considers the influence of sentimental intensity to improve extraction accuracy. To address this issue, we propose an extraction model based on sentiment analysis and 3D Convolutional Neural Networks (3D-CNN), named SEE-3D. First, to prepare fundamental data for sentiment analysis, emotion clauses are clustered into six emotion domains according to six emotion types in the ECPE dataset. Then, a pre-trained sentiment analysis model is introduced to compute emotional similarity, which provides a reference for identifying emotion clauses. In the extraction process, similar features of adjacent documents in the same batch of samples are fused as input of 3D-CNN. The 3D-CNN enhances the macro semantic understanding ability of the model, thereby improving the extraction performance. The results of experiments show that the accuracy of ECPE can be effectively improved by the SEE-3D model.

Keywords: ECPE, Sentiment analysis, Neural networks, 3D-CNN.

* The initial idea has been presented with some preliminary experimental results in ATCI 2022 [1]. The current paper is an extended version that contains some new ideas, formulations, and more extensive experimental results.

† Corresponding author

1. Introduction

With the accumulation of life pressure and the accelerated pace of life, people need more emotional understanding than ever before. Different from traditional psychological counseling services, ECPE 2 can provide technical support for intelligent psychological counseling, empty-nest elderly care, and other fields. However, although many excellent methods have emerged in recent years, the extraction effect can still be improved if the sentimental intensity and template similarity are considered.

In the related research on sentiment analysis, we propose a key sentence extraction algorithm and a sentiment classification model (SC-CMC-KS) for processing microblog comment texts 3. Combined with a hybrid neural network and ELECTRA, a sentiment classification model 4 is proposed for Chinese short comment texts. In 5, Xu et al. proposed an emotional element extraction model (ALSEE) for extracting keywords from all aspects of product reviews. Unlike our previous works, sentiment analysis can also be utilized for relation extraction tasks at the clause level. To further apply sentiment analysis to the ECPE task, the following two aspects need to be considered: (1) The dataset contains many clauses which need to be divided according to the type of emotion. The intensity of emotion needs to be quantified by a unified indicator. (2) To mine the similarity of documents, it is necessary to build candidate emotion-cause pairs (EC-Pairs) and construct a network to realize parallel processing of multiple documents.

Based on the above two considerations, we constructed an ECPE model based on sentiment analysis and 3D-CNN, named SEE-3D. The process of extracting EC-Pairs can be divided into three parts: encoding clauses as features, sentiment analysis & classification, and 3D fusion extraction. The framework of SEE-3D is illustrated in Figure 1:

(1) **Encoding Clauses as Features**. The word embedding is generated by BERT, which can carry rich semantic information. A word-level BiLSTM is constructed to encode semantic information of context words. To further obtain the intrinsic relationship between clauses, we introduced the attention mechanism to BiLSTM and encoded clauses as features at the coarse-grained level.

(2) **Sentiment Analysis & Classification**. The features generated in the previous step are the original input of this part. Firstly, each clause in the same document is paired with each other as EC-Pairs, and a quantitative method based on Euclidean distance is proposed as an emotional similarity. Then, the classification module concatenates the clause vectors of EC-Pairs, a linear layer is introduced to calculate the probability. By calculating the emotional similarity of emotion clauses in candidate EC-Pairs, we further obtain the emotional similarity of all EC-Pairs. Finally, to fuse emotional information and probability, each document is constructed as a probability matrix in the form of a 2D matrix.

(3) **3D Fusion Extraction**. After analyzing the documents in the dataset, we hypothesize template similarity between documents. To further prove this hypothesis, we stack 2D matrices constructed in the previous part into 3D matrices and pass them into a 3D-CNN for fusion extraction. After performing a 3D convolution operation, the network's output is split into several 2D probability matrices, i.e., the final extraction result is obtained. Ablation experiments in section 4.5 proved the validity of the hypothesis.

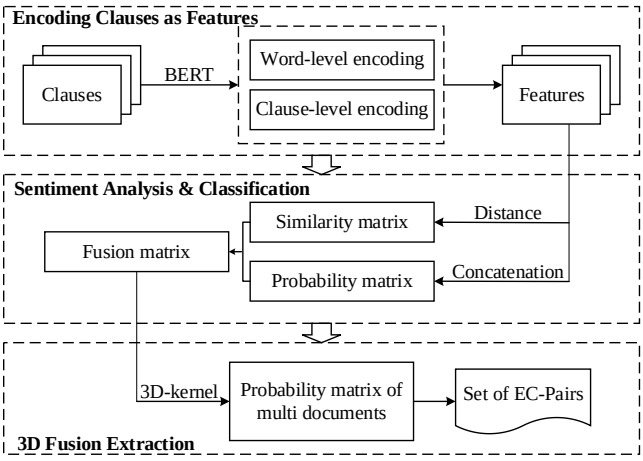


Fig. 1. System Framework of SEE-3D

In this paper, to further represent the sentimental intensity of clauses, an emotional similarity representation method is proposed for the ECPE task. We apply Euclidean distance to measure the space between clauses and the center of the emotion domains. In brief, the closer the distance to the emotion domain is, the greater the probability of the emotion clause is. We incorporate sentimental intensity into the EC-pairs extraction process through this similarity representation. To enhance the understanding ability of the model, two prediction probabilities of documents are fused by a 3D fusion extraction method. Thus, the SEE-3D model can reach relatively high accuracy and significantly improve emotion clause extraction accuracy.

The SEE-3D model is trained by using the ECPE benchmark dataset. The advantage of the encoding module is that the semantic information carried by BERT can be learned. The sentiment analysis module is introduced as a pre-trained model, providing a reference for representing emotional similarity. Meanwhile, 3D-CNN provides strong support for capturing local semantic information. The advantage of the SEE-3D model is that it considers the sentimental intensity of clauses and improves the accuracy of EC-Pairs extraction.

In this paper, the remaining sections are organized as follows. In Section 2, we introduce previous research on ECPE; Section 3 introduces the SEE-3D model by module; Analyzes and discusses the results of experiments are shown in Section 4; Finally, future work and summary are presented in Section 5.

2. Related Work

ECPE is developed from ECE 6. The task goal of ECE emphasizes finding the cause of the emergence of an emotional word, i.e., locating the corresponding cause clause through emotional words. In 7, Xia et al. proposed a hierarchical RNN method for ECE. The task goal of ECPE emphasizes finding out which events are the causes of an

emotional event, i.e., matching the corresponding cause clauses through emotion clause positioning.

2.1. From ECE to ECPE

ECE is based on word-level, focusing on detecting cause events. In 8, Ding et al. combined the three features of relative position, text content, and global label to improve the accuracy of ECE extraction. In recent years, although there have been many excellent research results 9101112, limited by the utilization of emotional words, the research results are not easy to apply to realistic scenarios. ECPE is based on clause-level, focusing on causal event pairing. In 2, Xia et al. split the extraction task into two sub-steps, the first classifying the clauses into sentiment or reason clauses and the second calculating all possible EC-Pairs and filtering them with custom filtering rules.

2.2. Recent research on ECPE

To avoid the error propagation caused by two-step extraction, In 13, Song et al. proposed an end-to-end network based on connection learning, which represents clauses as nodes and extracts EC-Pair through directed connectivity learning between nodes, while the study 14 proposed a range control mechanism to discover the interaction between cause and emotion through multi-task learning. In 15, Tang et al. used a multi-layer attention mechanism to model the intrinsic semantic information in EC-Pairs. Study 16 proposed a classification method that did not depend on the extraction results and improved the model's performance by adding a position awareness mechanism. In 17, Chen et al. constructed a pair graph to represent the relationship pairs in the document,three kinds of dependencies were defined, and the graph convolution neural network was utilized to learn the dependencies for EC-Pairs extraction. Study 18 regarded ECPE as a sequence labeling task and used CNN and BiLSTM to extract emotional cause pairs after the unified labeling of data sets. In 19, Cheng et al. designed an asymmetric network to find the local area EC-Pair by cross-subnetwork local pair searcher. Study 20 proposed a joint framework for multi-task learning and used the sliding window mechanism to reduce the noise generated by long-distance clauses during extraction. In 2122, Yuan et al. and Fan et al. respectively proposed two labeling strategies based on different characteristics of the ECPE task to improve accuracy. By analyzing the construction process of a directed graph with labels, Fan et al. captured the causal relationship between clauses 23. In addition, some end-to-end models 242526 and two-step models 2728 have achieved good performance.

Although ECPE does not need to annotate emotional words in advance, the influence of emotional words on the extraction results cannot be ignored. Therefore, we propose the SEE-3D model in this paper and integrate emotional distance as an essential indicator in the extraction process.

3. Methods

3.1. Architecture

The SEE-3D model consists of three modules: (a) Encoding Clauses as Features; (b) Sentiment Analysis & Classification; (c) 3D Fusion Extraction. The overall structure is shown in Figure 2.

In Figure 2(a), the model regards each clause in documents as a word sequence $c = \{w_1, w_2, ..., w_{|c|}\}$, which w represents a word in the clause and $|c|$ is the number of words, and generates clause-level representation by BiLSTM with an attention mechanism. Figure 2(b) contains two sub-modules:

1. The pre-trained sentiment analysis model aims to convert the output of Encoder into a sentiment vector. By calculating the Euclidean distance between the sentiment vector and the center of the nearest emotion domain, we convert the distance into similarity to judge the probability of an emotion clause. Euclidean distance can be more accurate in calculating the same dimension vector in K-means clustering space. The parameters of the sentiment analysis model are frozen during training.
2. The classification module regards all clauses as emotion and cause to build emotion set and cause set, then construct candidate EC-Pairs as $ECPs = \{(e^1, c^1), (e^2, c^2), ..., (e^m, c^m)\}$, where m is the maximum number of clauses for all documents. For each (e, c) in ECPs, pass $r^e \oplus r^c$ into a linear layer to calculate the probability that (e, c) is true, where r^e and r^c is the output of e and c from BiLSTM in Figure 1.

The input of Figure 2(c) is a three-dimensional matrix, and each element of the matrix is the output weighted sum of the two sub-models in Figure 2(b). 3D-CNN learns the global similarity between samples, and the extraction results of each document are finally output. The red lattice in Figure 2(c) represents the EC-Pair with the highest prediction probability of the model as the final extraction result.

3.2. Emotion Domain

The ECPE benchmark corpus 29 contains six emotions. Our analysis of samples in the corpus can be concluded as follows: the polarity and strength of the six emotions are significantly different, and they are pretty different from the clauses without emotional words. The distribution of emotion types of samples is shown in Table 1. Sadness accounted for the highest proportion of 26.94% of the six emotions, and Surprise accounted for the lowest proportion of 4.18%.

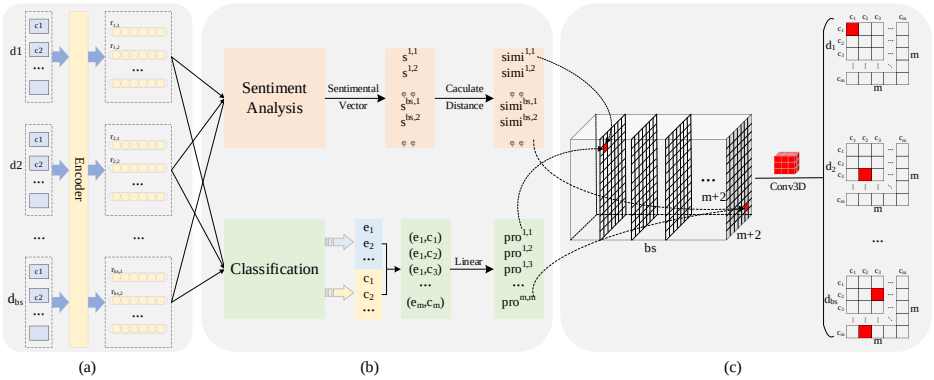


Fig. 2. Architecture of SEE-3D

Table 1. Distribution of emotion types

Emotion	Percentage	Number
Surprise	4.18%	88
Disgust	10.69%	225
Anger	14.35%	302
Fear	18.00%	379
Happiness	25.83%	544
Sadness	26.94%	567

We hope to provide a reference for verifying whether clauses contain emotions by constructing six types of emotion domains. Firstly, a pre-trained sentiment analysis model 4 is utilized to convert each clause-level vector into a sentiment vector. Then all sentiment clauses in the corpus are clustered by a k-means 30 clustering algorithm (k=6) to construct the emotion domains of six sentiment types, denote as $domain^{S} = \{sd_1, sd_2, ..., sd_6\}$. In the follow-up work, we obtain the cluster centers of six emotion domains, denote as $domain^{C} = \{sdc_1, sdc_2, ..., sdc_6\}$, where each represents the center of emotion domains. By calculating the center vector and radius of each cluster, it can provide support for subsequent emotional computing. Based on the above operations, the emotional distance between clauses and emotion domains can be obtained directly.

3.3. Encoding Clauses as Features

The input of the model is one batch of documents is denoted as $D = \{d_1, d_2, ..., d_{bs}\}$, where bs is the batch size of documents. For each document d_i in D, it contains several clauses $\{c^{i,1}, c^{i,2}, ..., c^{i,|d_i|}\}$, where |di| is the number of clauses in d_i. Any clause $c^{i,j}$ is regarded as a word sequence: $\{w_1^{i,j}, w_2^{i,j}, ..., w_{|c^{i,j}|}^{i,j}\}$. Each clause is passed into BiLSTM to

encode semantic information between the words and obtain the hidden state of BiLSTM $h^{i,j} = \{h_1^{i,j}, h_2^{i,j}, ..., h_{|c^{i,j}|}^{i,j}\}$.

$$\alpha_k = \frac{\exp(H_k^{i,j} W_k)}{\sum_{|c_{i,j}|} \exp(H_k^{i,j} W_k)} \quad (1)$$

$$r_j^i = \sum_{k=1}^{|c_{i,j}|} \alpha_k H_k^{i,j} \quad (2)$$

where W_k is a learnable weight parameter. Figure 3 describes the process of generating clause-level representations from a document. The details of BiLSTM 31 and attention mechanism 32 are omitted in this paper.

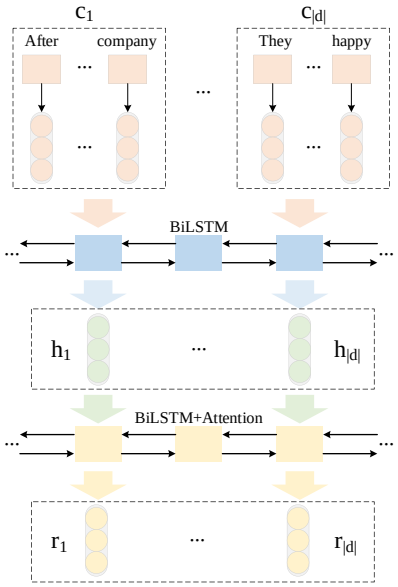


Fig. 3. Process of encoding a document

3.4. Encoding Clauses as Features

This part consists of two subtasks: (1) Generate the sentiment vectors of the clauses and calculate the distances from emotion domains; (2) Regard each clause as both cause and emotion, then construct all candidate EC-Pairs in a document.

The pre-trained sentiment analysis model generates the sentiment vector $s^{i,j}$ for the *j*-th clause $c^{i,j}$ of the *i*-th document. To find the closest emotion type of the clause, we calculate the Euclidean distance between $s^{i,j}$ and centers of the emotion domains, denoted as $\{sdc_1, sdc_2, ..., sdc_7\}$, then select the minimum distance $dis^{i,j}$. For each

emotion domain, the domain radius is denoted as rad_i, which is calculated by Formula (3):

$$rad_i = \frac{1}{|sd_i|}\sum_{j=1}^{|sd_i|}\text{Euclidean}(sdc_i, v_j) \tag{3}$$

$$dis^{i,j} = \min(\text{Euclidean}(sdc_k, s^{i,j})) \tag{4}$$

where $|sd_i|$ is the number of emotion clauses in the emotion domain and v_j represents each clause in the emotion domain. If $dis^{i,j}$ is less than or equal to the corresponding minimum distance, we consider that the clause is more likely to be an emotion clause. Sentiment score $simi^{i,j} \in (0, 1]$ is calculated for screening EC-Pairs according to the distance between a clause and emotion domains.

$$simi^{i,j} = \frac{1}{1+dis^{i,j}} \tag{5}$$

In order to minimize the propagation of error in the classification-pairing mode, we do not directly classify the output vectors of BiLSTM. Each clause is regarded as both cause and emotion, and candidate EC-Pairs are constructed by cross-combination, organized as a two-dimensional matrix. We concatenate the vectors of the clauses in candidate EC-Pairs, then pass them into a linear layer to calculate the probability, denoted as $pro^{i,j}$.

$$pro^{i,j} = \text{Linear}([r^e; r^c]) \tag{6}$$

In Formula (6), r^e and r^c are the output features encoded by the clause-level BiLSTM, and $[\cdot;\cdot]$ means concatenation.

As shown in Figure 4, the red grid in the figure represents the ground truth EC-Pair, and the maximum number of clauses in this batch of documents is denoted as *m*. If the number of clauses is less than m, it needs to be aligned by padding zero. To obtain the probability matrix with the shape of (bs, m, m) by 3D convolution operation., we padded zero in the shadow grids at the boundary in the figure.

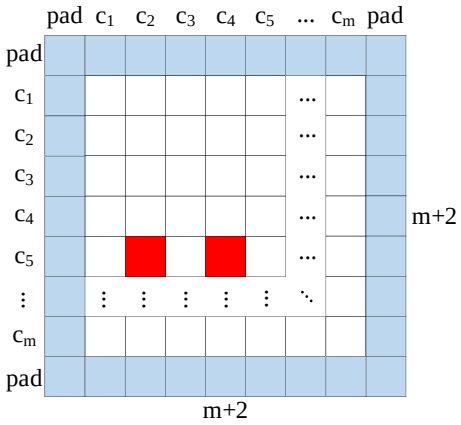


Fig. 4. Candidate EC-Pair in a document

3.5. 3D Fusion Extraction

We analyzed the original corpus of the ECPE dataset and found that the source of the corpus is the news reports of SINA NEWS[‡] in three years, and documents are similar in structure. Considering the normalization of news report writing, we hypothesize that clauses in different documents have strong similarities in position. To capture this similar feature, we stack multiple documents into 3D shapes according to the structure of Figure 4, then pass them into the 3D-CNN network with a 3D convolution kernel.

Different from Figure 4, each element in the 3D matrix $element^{i,j}$ is composed of two dimensions, i.e., the channel of 3D-CNN is 2:

$$element^{i,j} = [\alpha simi^{i,j}, \beta pro^{i,j}] \quad (7)$$

where α, β are the weight parameters of the output of the two sub-modules, which are obtained by experiments. After performing the 3D convolution operation, the model learns similar features between adjacent documents and obtains the probability matrix $\mathbf{W}^p \in ¡^{bs\times m\times m}$ of all candidate EC-Pairs through a Softmax function.

$$\mathbf{W}^p = \text{Softmax}(3\text{D-CNN}(\mathbf{W})) \quad (8)$$

where $\mathbf{W} \in ¡^{bs\times(m+2)\times(m+2)}$ is the input matrix of 3D-CNN. To keep the stability of the sentiment vector generated by the pre-trained sentiment analysis model, parameters of the sentiment analysis module are frozen during training. The loss of 3D-CNN is obtained by calculating the output of 3D-CNN with cross-entropy loss function:

$$Loss = -\frac{1}{N}\sum_{\hat{y}^{i,j}\in\mathbf{W}^p}[y^{i,j}\log(\hat{y}^{i,j}) + (1-y^{i,j})\log(1-\hat{y}^{i,j})] \quad (9)$$

where N is the number of EC-Pairs, $\hat{y}^{i,j}$ and $y^{i,j}$ are the predicted label and ground-truth label of $element^{i,j}$.

4. Experiments

4.1. Dataset & Metrics

In 29, Gui et al. annotated 2105 documents from 20000 articles on the SINA NEWS website for the ECE task. However, documents containing multiple EC-Pairs are split into multiple samples, which is inconsistent with the actual scenario. Study 2 combined and annotated the ECE benchmark dataset, then constructed the ECPE benchmark dataset. The data distribution is shown in Table 2.

Each document in the dataset contains several clauses constituting one or more EC-Pairs. Compared with the ECE benchmark dataset, about 10% of the documents contain multiple EC-Pairs. In order to extract all EC-Pairs, we enumerate all possible clause

[‡] https://news.sina.com.cn/

combinations in a document, then pass the 3D matrix composed of multiple documents into the 3D-CNN network to calculate the EC-Pair probability matrix $\mathbf{W}^p$ for each document.

Table 2. Distribution of ECPE dataset

	Number	Percentage
Number of EC-Pair in Document equals 1	1746	89.77%
Number of EC-Pair in Document equals 2	177	9.10%
Number of EC-Pair in Document greater than 2	22	1.13%
All	1945	100%

Figure 5(a) shows the five clauses contained in the document {C1, C2, C3, C4, C5}, where C5 is the emotion clause (the type of emotion word here is Happiness); Figure 5(b) describes the causal relationship between the clauses in this document. C5 caused by both C2 and C4, i.e., the EC-Pairs contained in the document are {C5←C2} and {C5←C4}.

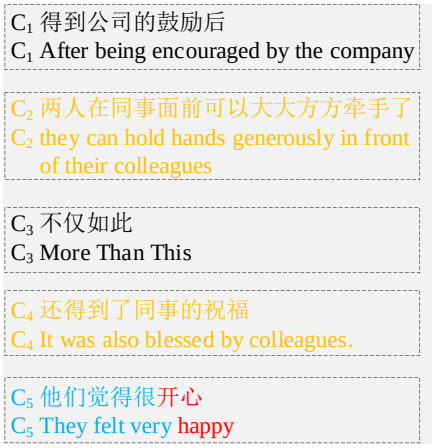


Fig. 5. Example in dataset

We construct two datasets for training and testing, the ratio of them is 9:1, then calculate the average score using ten-fold cross-validation. The same as the baseline model, we apply P(Precession), R(Recall), and F1 score as the metrics of our model.

4.2. Experimental Settings

Compared with word2vec 33 method, BERT 34 is more robust. The Chinese fine-tuning BERT[§] model generates word vectors of clauses, and the dimension is 768. In the encoding process, the number of hidden layer units of BiLSTM equals 100. In the 3D fusion extraction module, the convolution kernel size of 3D-CNN is set to (3, 3, 3). We apply dropout 35 to the linear layer to avoid overfitting and set the dropout rate to 0.3. The default weight and bias of BiLSTM, 3D-CNN, and linear layer are initialized by the Xavier method 36.

During the training period, to make the model's convergence process smoother, the learning rate is set to 3e-5, and introduced Adam 37 as the optimizer of the model. In addition, the batch size is 16 and the epoch of training is 100.

4.3. Compared ECPE Models

We compare these ECPE models with SEE-3D:

Baselines: This model is the benchmark model when Study 2 first proposed the ECPE task, including three baseline models: Indep, Inter-CE, and Inter-EC. This model divides the extraction into two steps:(1) Classifying each clause into emotion clause or cause clause to combine candidate EC-Pairs; (2) Eliminating EC-Pairs of low probability.

RANKCP: This extraction method is proposed by Study 24. The author used a graph attention mechanism to learn the interaction between emotion clauses and cause clauses in documents for filtering EC-Pairs.

E2EECPE: This end-to-end framework was proposed by Study 13. Biaffine attention is utilized to predict the causal relationship between clauses, Song transforming the clause pairing problem into the edge prediction problem.

ECPE-2D: This model is an improved method for **ECPE-2Steps** by Study 25. They represent EC-Pairs as a 2D structure that models the interactions between different EC-Pairs through two different 2D Transformers (Windows-constrained, Cross-road).

DQAN: The author used a dual-questioning attention mechanism to obtain the interaction between clauses in EC-Pair and other clauses, which improved the semantic understanding ability of the model 27.

Trans-ECPE: In 23, Fan et al. regard the EC-Pair extraction process as the construction process of a directed graph. The model regards clauses as nodes of the graph and extracts EC-Pairs while constructing the graph. The two versions of LSTM and BERT represent the different structures used in the model coding characteristics.

[§] https://huggingface.co/bert-base-chinese

4.4. Experimental Discussion

The SEE-3D model and other models are verified on the ECPE benchmark dataset, Table 3 illustrates the results of experiments. In addition to the EC-Pair extraction results, the emotional extraction and cause extraction results of models are also listed in the table.

(1) Our model achieved the **highest precession and F1 score** in **emotion** extraction, 88.43%, and 86.74%. In these comparative models, **Indep** extracts the causes and results separately, without fully considering the interaction between clauses. **Inter-CE** and **Inter-EC** add emotion(or cause) clauses to the classification process of cause(or emotion) clauses, increasing the accuracy by about 1%-2%. Inspired by these, the subsequent models are constructed to explore the internal relationship between different clauses. **RANKCP** constructs global features between clauses and finally achieves the highest recall rate of 87.03% when extracting emotion clauses. Based on this skill, **RANKCP** achieving the highest recall rate of 66.98% when extracting EC-Pairs. The sentiment analysis module of **SEE-3D** calculates the emotional similarity and forward feedbacks the emotional features of clauses in EC-Pairs, which helps to better identify emotion clauses.

(2) In the **EC-Pair** extraction, the precession and the F1 score were finally improved. Compared with the LSTM model, the BERT model used in **Trans-ECPE** coding has significantly improved in all aspects. Besides, **Trans-ECPE** based on BERT has achieved the highest accuracy and F1 score in cause extraction, 75.62%, and 69.74%. **ECPE-2D** based on Inter-EC is selected for comparison because it integrates emotional features into the cause clauses classification process, similar to **SEE-3D**. So we can better mine the influence of sentimental intensity on the extraction results. with **Trans-ECPE**(BERT), the extraction accuracy of emotion clauses increases by 1.27%, and the F1 score increases by 2%, which further improves the performance of 3D fusion extraction.

Table 3. Comparison results with other ECPE models. '♠' denotes the model in this paper. '-' denotes the original paper that had not reported the evaluation result

Model	Emotion Extraction			Cause Extraction			EC-Pair Extraction		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Baselines (Indep)	83.75	80.71	82.10	69.02	56.73	62.05	68.32	50.82	58.18
Baselines (Inter-CE)	84.94	81.22	83.00	68.09	56.34	61.51	69.02	51.35	59.01
Baselines (Inter-EC)	83.64	81.07	82.30	70.41	60.83	65.07	67.21	57.05	61.28
RANKCP	85.48	**87.03**	84.06	68.24	69.27	67.43	66.10	**66.98**	65.46
E2EECPE	85.95	79.15	82.38	70.62	60.30	65.03	64.78	61.05	62.80
ECPE-2D (Inter-EC)	85.37	81.97	83.54	71.51	62.74	66.76	71.73	57.54	63.66
DQAN							67.33	60.40	63.62
Trans-ECPE (LSTM)	80.80	84.39	82.56	67.42	65.34	66.36	65.15	63.54	64.34
Trans-ECPE (BERT)	87.16	82.44	84.74	**75.62**	**64.71**	**69.74**	**73.74**	63.07	67.99
SEE-3D♠	**88.43**	85.12	**86.74**	70.52	63.67	66.92	72.58	66.32	**69.31**

Overall, **SEE-3D** improves the extraction performance by deepening the understanding of emotional semantic information. The above analysis shows that the emotional dependence between clauses has important research value in the ECPE task.

4.5. Ablation Study

To investigate the effectiveness of sentiment analysis, we removed the calculation of emotion domain and emotional similarity when filtering EC-Pairs, i.e., the model only retains the clause encoder, linear layer, and 3D-CNN. When the model only retains the output value of the linear layer as the probability, it is difficult for 3D-CNN to learn the sentimental intensity interaction between documents. The extraction results of EC-Pairs are shown in Table 4. The F1 score of cause extraction is almost unchanged, but the precession, recall, and F1 score of emotion extraction are reduced by 3.82%, 4.23%, and 4.03%. Although the similarity of some cause clauses is learned in 3D-CNN, due to the lack of emotional semantic information, the final F1 score of EC-Pair extraction is 6.02% lower than that of the original **SEE-3D** model.

From the discussion above, we can find that emotional semantic information significantly influences the ECPE task. Due to the starting point of the ECPE task serving for the emotional reasoning task, the fusion of the information learned by sentiment analysis and cause extraction can improve the performance of our model.

Table 4. Ablation Study of Sentiment Analysis. We removed the emotional similarity metric from our model to reveal the effect of the sentiment analysis sub-module, denoted as "-SA"

Model	Emotion Extraction			Cause Extraction			EC-Pair Extraction		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
SEE-3D (-SA)	84.61	80.89	82.71	70.39	**63.75**	66.90	65.46	61.25	63.29
SEE-3D	**88.43**	**85.12**	**86.74**	**70.52**	63.67	**66.92**	**72.58**	**66.32**	**69.31**

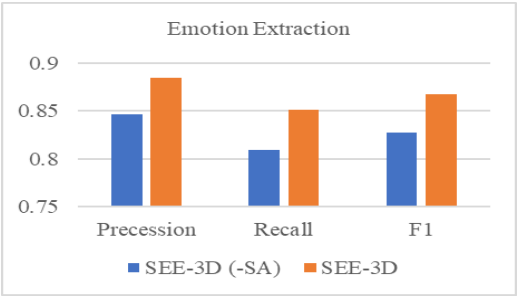


Fig. 6. Ablation Study of Sentiment Analysis

By analyzing the corpus source of the ECPE benchmark dataset, we proposed the introduction of 3D-CNN learning similar features of samples. To prove the effectiveness of 3D operation, we removed 3D-CNN from SEE-3D, and the comparison results are listed in Table 5. When our model directly uses the weighted sum of emotional similarity and linear layer output as the prediction results, the precession, recall, and F1

score are reduced by 2.39%, 2.17%, and 2.28%. In short, 3D-CNN learned the similarity between samples.

Table 5. Ablation Study of 3D-CNN. We removed the 3D-CNN from our model and directly used elements in Wp as extraction results, denoted as "-3D Conv"

Model	EC-Pair Extraction		
	P(%)	R(%)	F1(%)
SEE-3D (-3D Conv)	70.19	64.15	67.03
SEE-3D	**72.58**	**66.32**	**69.31**

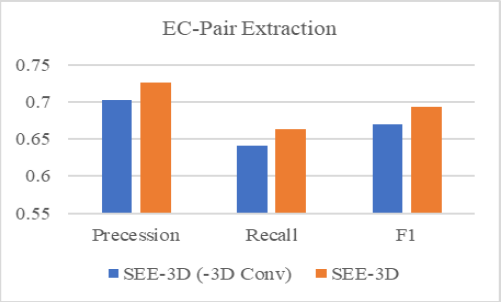


Fig. 7. Ablation Study of 3D-CNN

5. Conclusions

As an essential task in cause detection, ECPE can be applied to emotional reasoning in daily life and is helpful to improve the service quality of psychological counseling. To make emotion-cause detection more efficient, a sentiment-driven 3D extraction model SEE-3D is proposed. Our model integrates sentiment analysis and 3D convolution into the EC-Pairs extraction process and improves extraction performance. In this paper, the achievements can be summarized as follows:

(1) A pre-trained sentiment analysis model is integrated into the ECPE model effectively and enhances the extraction performance. We use the k-means clustering algorithm to process and analysis the emotional clauses, making the model further obtain the emotion domain distribution. The distance between the clause and emotion domain is used to represent the sentimental intensity of the clause.

(2) A 3D fusion extraction method is proposed to learn similar features of documents, which improves the accuracy of ECPE. The outputs of sentiment analysis are combined with the classical pairing algorithm. Thus, the model can learn the similar characteristics between samples, which effectively improves the accuracy of extraction.

The experiments on the ECPE benchmark dataset show that our model can learn the interaction between emotion and cause in documents.

Similar to other methods, our extraction method also has its limitations. It has a remarkable effect only when the samples are relatively uniform and the text writing format has the characteristics of a template. In other cause detection datasets, the 3D fusion extraction method does not necessarily have generalization ability. To further study datasets in other fields, we will continue to conduct research on cause detection and try to apply it to business scenarios.

Acknowledgments. This work was supported by the Graduate Innovation Foundation Project of Anhui University of Science & Technology: [grant number 2021CX2112], the National Natural Science Foundation of China (Grant NO.62076006), the University Synergy Innovation Program of Anhui Province (GXXT-2021-008), and the Anhui Provincial Key R&D Program (202004b11020029).

References

1. Xu X, Wu, H and Zhu, G. SCA-ECPE: Emotion-Cause Pair Extraction Based on Sentiment Clustering Analysis. In 2022 International Conference on Applications and Techniques in Cyber Intelligence (ATCI). (2022)
2. Xia, R. and Ding, Z., Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 1003-1012. (2019)
3. Zhang, S., Hu, Z., Zhu, G., Jin, M., & Li, K. C. Sentiment classification model for Chinese micro-blog comments based on key sentences extraction. Soft Computing, Vol. 25, No. 1, 463-476. (2021)
4. Zhang, S., Yu, H., & Zhu, G. An emotional classification method of Chinese short comment text based on ELECTRA. Connection Science, Vol. 34, No. 1, 254-273. (2022)
5. Xu, H., Zhang, S., Zhu, G., & Zhu, H. ALSEE: a framework for attribute-level sentiment element extraction towards product reviews. Connection Science, Vol. 34, No. 1, 205-223. (2022)
6. Lee, S. Y. M., Chen, Y., & Huang, C. R. A text-driven rule-based system for emotion cause detection. In Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text, Los Angeles, CA, 45-53. (2010)
7. Xia, R., Zhang, M., & Ding, Z. RTHN: A rnn-transformer hierarchical network for emotion cause extraction. arXiv preprint arXiv:1906.01236. (2019)
8. Ding, Z., He, H., Zhang, M., & Xia, R. From independent prediction to reordered prediction: Integrating relative position and global label information to emotion cause identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA, Vol. 33, No. 1, 6343-6350. (2019)
9. Diao, Y., Lin, H., Yang, L., Fan, X., Chu, Y., Wu, D., & Xu, K. Emotion cause detection with enhanced-representation attention convolutional-context network. Soft Computing, Vol. 25, No. 2, 1297-1307. (2021)
10. Hu, G., Lu, G., & Zhao, Y. FSS-GCN: A graph convolutional networks with fusion of semantic and structure for emotion cause analysis. Knowledge-Based Systems, 212, 106584. (2021)
11. Yu, W., & Shi, C. Emotion Cause Extraction by Combining Intra-clause Sentiment-enhanced Attention and Inter-clause Consistency Interaction. In 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), 146-150. IEEE. (2021)
12. Yan, H., Gui, L., Pergola, G., & He, Y. Position Bias Mitigation: A Knowledge-Aware Graph Model for Emotion Cause Extraction. In Proceedings of the 59th Annual Meeting of

the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 3364-3375. (2021)
13. Song, H., Zhang, C., Li, Q., & Song, D. End-to-end emotion-cause pair extraction via learning to link. arXiv preprint arXiv:2002.10710. (2020)
14. Fan, R., Wang, Y., & He, T. An end-to-end multi-task learning network with scope controller for emotion-cause pair extraction. In CCF International Conference on Natural Language Processing and Chinese Computing, 764-776. Springer, Cham. (2020)
15. Tang, H., Ji, D., & Zhou, Q. Joint multi-level attentional model for emotion detection and emotion-cause pair extraction. Neurocomputing, 409, 329-340. (2020)
16. Wu, S., Chen, F., Wu, F., Huang, Y., & Li, X. A multi-task learning neural network for emotion-cause pair extraction. In ECAI 2020, 2212-2219. IOS Press. (2020)
17. Chen, Y., Hou, W., Li, S., Wu, C., & Zhang, X. End-to-end emotion-cause pair extraction with graph convolutional network. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 198-207. (2020)
18. Chen, X., Li, Q., & Wang, J. A unified sequence labeling model for emotion cause pair extraction. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 208-218. (2020)
19. Cheng, Z., Jiang, Z., Yin, Y., Yu, H., & Gu, Q. A symmetric local search network for emotion-cause pair extraction. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 139-149. (2020)
20. Ding, Z., Xia, R., & Yu, J. End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 3574-3583. (2020)
21. Yuan, C., Fan, C., Bao, J., & Xu, R. Emotion-cause pair extraction as sequence labeling based on a novel tagging scheme. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 3568-3573. (2020)
22. Fan, C., Yuan, C., Gui, L., Zhang, Y., & Xu, R. Multi-task sequence tagging for emotion-cause pair extraction via tag distribution refinement. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 2339-2350. (2021)
23. Fan, C., Yuan, C., Du, J., Gui, L., Yang, M., & Xu, R. Transition-based Directed Graph Construction for Emotion-Cause Pair Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 3707-3717. (2020)
24. Wei, P., Zhao, J., & Mao, W. Effective Inter-Clause Modeling for End-to-End Emotion-Cause Pair Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 3171-3181. (2020)
25. Ding, Z., Xia, R., & Yu, J. ECPE-2D: emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 3161-3170. (2020)
26. Singh, A., Hingane, S., Wani, S., & Modi, A. An End-to-End Network for Emotion-Cause Pair Extraction. In Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Zhengzhou, China, Sentiment and Social Media Analysis, 84-91. (2021)
27. Sun, Q., Yin, Y., & Yu, H. (2021, July). A Dual-Questioning Attention Network for Emotion-Cause Pair Extraction with Context Awareness. In 2021 International Joint Conference on Neural Networks (IJCNN), 1-8. IEEE. (2021)
28. Yu, J., Liu, W., He, Y., & Zhang, C. A mutually auxiliary multitask model with self-distillation for emotion-cause pair extraction. IEEE Access, 9, 26811-26821. (2021)
29. Gui, L., Wu, D., Xu, R., Lu, Q., & Zhou, Y. Event-Driven Emotion Cause Extraction with Corpus Construction. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, USA, 1639-1649. (2016)
30. Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. An efficient k-means clustering algorithm: Analysis and implementation. IEEE transactions on pattern analysis and machine intelligence, Vol. 24, No. 7, 881-892. (2002)

31. Graves, A., Mohamed, A. R., & Hinton, G. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing, 6645-6649. IEEE. (2013)
32. Bahdanau, D., Cho, K., & Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. (2014)
33. Mikolov, T., Chen, K., Corrado, G., & Dean, J. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781. (2013)
34. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. (2018)
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, Vol. 15, No. 1, 1929-1958. (2014)
36. Glorot, X., & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 249-256. (2010)
37. Kingma, D. P., & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. (2014)

Xin Xu, born in 1991, Master candidate at Anhui University of Science and Technology. His main research interests include relation extraction and machine learning.

Guangli Zhu, born in 1969. Master, Associate professor, Master supervisor. Her current research interests include Web Mining, Semantic Search, and Calculation theory.

Houyue Wu, born in 1996. Master candidate at Anhui University of Science and Technology. His main research interests are Adversarial Sample Generation and Relation Extraction.

Shunxiang Zhang, born in 1970. PhD, professor, PhD supervisor. He is an professor at Anhui University of Science and Technology, China. His current research interests include Web Mining, Semantic Search, and Complex network.

KuanChing Li, born in 1967, professor, PhD supervisor. He is a University Distinguished Professor at Providence University, Taiwan. His current research interests include Parallel and Distributed Computing, Big Data and Emerging Technologies.

Received: March 03, 2022; Accepted: September 20, 2022.

Computer Science and Information Systems 20(1):95–115 https://doi.org/10.2298/CSIS210804049R

Solving the P-Second Center Problem with Variable Neighborhood Search

Dalibor Ristic[1,*], Dragan Urosevic[1, 2], Nenad Mladenovic[3], and Raca Todosijevic[4]

[1] School of Computing, Union University, Knez Mihailova 6, 11000 Belgrade, Serbia
dalibor.ristic@outlook.com
[2] Mathematical Institute of the Serbian Academy of Sciences and Arts, Knez Mihailova 36, 11000 Belgrade, Serbia
durosevic@raf.rs
[3] Khalifa University, PO Box 127788, Abu Dhabi, United Arab Emirates
nenadmladenovic12@gmail.com
[4] Polytechnic University of Hauts-de-France, Cedex 9, Valenciennes, France
racatodosijevic@gmail.com

Abstract. The p-center problem is a well-known and highly studied problem pertaining to the identification of p of the potential n center locations in such a way as to minimize the maximum distance between the users and the closest center. As opposed to the p-center, the p-second center problem minimizes the maximum sum of the distances from the users to the closest and the second closest centers. In this paper, we propose a new Variable Neighborhood Search based algorithm for solving the p-second center problem. Its performance is assessed on the benchmark instances from the literature. Moreover, to further evaluate the algorithm's performance, we generated larger instances with 1000, 1500, 2000, and 2500 nodes and instances defined over graphs up to 1000 nodes with different densities. The obtained results clearly demonstrate the effectiveness and efficiency of the proposed algorithm.

Keywords: variable neighborhood method, heuristic algorithms, p-second center problem, combinatorial optimization.

1. Introduction

The p-center problem (pCP) was introduced in 1965 [7] as a discrete optimization problem of identifying p from potential n centers in such a way as to minimize the maximum distance between the users and their closest center. The p-center in the real

* Corresponding author

world may appear as a problem of determining locations for the construction of hospitals, so that the distance of the farthest settlement to the hospital closest to it is the smallest possible. The same problem can be applied to the installation of gas stations, fire stations, etc.

The p-center problem is formally defined over an undirected weighted graph G = (V, E), where V is the set of all nodes, i.e. the locations of the centers and the users, while E is the set of all graph edges connecting those locations. The weights of the edges correspond to the distance between their ends. Let *d(i, j)* be the shortest distance between the *i* and *j* nodes in the G graph. If nodes *i* and *j* are not connected, *d(i, j)* is equal to infinity. The solution to the p-center problem is a set of nodes $P \subset V$, of cardinality p, so that the maximum distance from the users to the assigned center is minimized:

$$pCP(V,E) = \min_{\substack{P\subset V,\\ |P|=p}} \max_{i\in V} \left\{ \min_{\substack{j\in P,\\ (i,j)\in E}} d(i,j) \right\} \quad (1)$$

The p-center is an NP-hard problem [11], but it has been known and studied for a long time, so there are many articles and algorithms that deal with the problem. In the literature there are many exact mathematical models and heuristic algorithms that successfully find solutions to the p-center problem. Exact methods, such as [3], [4], [6], [10] and [13], deal with smaller problems, while solutions of larger instances are found by heuristic algorithms. The best-known heuristic algorithms are presented in [5], [9], [15] and [16].

The p-center does not offer a solution to the problem when the assigned center is not able to serve the user. The problem arises when, in the conditions of humanitarian catastrophes, the center becomes overloaded or there is a failure at the center caused by any reason. A simple solution is to assign a backup center to each user in the event of a primary failure. Guided by this idea, Albaredo-Sambola et al. [1] defined the p-next center problem (pNCP) in 2015. The problem of the p-next center is a generalization of the pCP. In the pNCP, it is required to select p from potential n centers in such a way as to minimize not only the maximum distance between the users and the centers closest to them but also the distance between the center and its closest center. Over the same graph G = (V, E) as in the case of the pCP, the p-next center problem is formally defined as:

$$pNCP(V,E) = \min_{\substack{P\subset V,\\ |P|=p}} \max_{i\in V} \left\{ \min_{\substack{j\in P,\\ (i,j)\in E}} d(i,j) + \min_{\substack{k\in P,\\ (j',k)\in E\\ k\neq j'\in \arg\min_{j\in P} d(i,j)}} d(j',k) \right\}, \quad (2)$$

where $arg\ min_{j\in P}\ d(i,j)$ corresponds to node *j* which is closest to node *i*

The G graph is a plane graph with no limits in terms of number of edges and connectivity. The weights of the edges correspond to the Euclidean distances between their nodes and therefore the distances satisfy the triangle inequality.

The p-next center problem, as a generalization of the pCP, is another NP-hard problem. The authors in the paper [1] present a few exact mathematical models as a solution to the problem, which are applicable to smaller instances of the problem. In

2019, Lopez-Sanchez et al. in their work [12] published the first heuristic algorithms for solving the p-next center problem.

It is expected that it is known in advance whether it is necessary to visit the backup center, and therefore, the p-second center problem (pSCP) is defined in response to the potential failure of the primary center [12]. The p-second center problem is a generalization of the p-center problem, in terms of identifying p out of n centers in order to minimize the maximum sum of the distances from the users to the closest and the second closest center. Formally, let G = (V, E) again be an undirected weighted graph, where the weights of the edges are determined by the distance between their ends, V is the set of all nodes, and E is the set of the edges. The centers, as well as other users, represent graph nodes, while *d(i, j)* is the shortest distance between the i and j nodes, calculated as a result of an algorithm for determining the shortest paths in the graph G. The solution of the p-second center problem is a set of nodes $P \subset V$, of cardinality p, so that the maximum distance from the users ($i \in V$) to the closest center ($j \in P$), plus the distance to the second closest center ($k \in P$) is minimized:

$$pSCP(V,E) = \min_{\substack{P \subset V, \\ |P|=p}} \max_{i \in V} \left\{ \min_{\substack{j \in P, \\ (i,j) \in E}} d(i,j) + \min_{\substack{k \in P, \\ (i,k) \in E, \\ k \neq j}} d(i,k) \right\} \quad (3)$$

The p-second center problem, as an extension of the pCP, is an NP-hard problem. To solve this problem, we propose a heuristic algorithm based on the variable neighborhood search method that includes an efficient local search method to accelerate the convergence to the local optimum. To this end, in the next section, through the description and pseudocode, we present the proposed algorithm. In the third section, we present a modification of the algorithm capable to recognize whether the found solution is optimal. In the fourth section, we present the obtained results of testing the proposed algorithms over a OR-Library [2] set of test instances, as well as two additional test sets generated for testing the algorithm on large instances of the problem. We end the paper with a short summary and an announcement of future work. We also compare the p-second center problem with the p-center and p-next center problems. There is an example graph that illustrates the pCP, the pNCP and the pSCP in Appendix 1.

Briefly, the contributions of our study are:

1. It is interesting to note that the property of finding the next better critical point in the interchange neighborhood lies within a circle whose radius is the current objective function value, which holds for all 3 variants of p-center problem. We proved this property for the p-second center problem.
2. However, for the p-center [15] and the p-next center [17] problems, we must keep track of the first and the second closest centers in the data structure, to assure efficient updating in local search. For the p-second center problem, we prove that the fast interchange move, first proposed by Whitaker in 1983 for solving p-median problem, can be performed by taking track of the third closest center of any user as well. This is another contribution of our study.
3. We included the fast interchange local search into the basic Variable neighborhood search and perform extensive numerical analysis on 40 OR-Library instances with up to 900 facilities and on new generated larger instances.

2. Algorithm

The proposed algorithm for solving the p-second center problem is based on Variable Neighborhood Search metaheuristic (VNS). The VNS was introduced by Mladenovic and Hansen (1997) [14] as a generic framework for building search algorithms. Starting from a predefined current solution, the VNS method continues searching within the randomly selected solution from the appropriate neighborhood of the current solution. The first neighborhood of a solution P, denoted by $N_1(P)$, contains solutions that differ from the solution P in exactly one element. In general, the set of solutions of the k-th neighborhood of the solution P is defined as:

$$N_k(P) = \{P \setminus \{u_1, u_2, \dots, u_k\} \cup \{v_1, v_2, \dots, v_k\} \mid u_1, u_2, \dots, u_k \in P, \quad (4)$$
$$v_1, v_2, \dots, v_k \in (V \setminus P), u_1 \neq u_2 \neq \dots \neq u_k, v_1 \neq v_2 \neq \dots \neq v_k\}$$

After searching through the N_k neighborhood of the current solution P, in case of finding a better solution P', the VNS algorithm rejects the previous solution and sets P' (P = P') as the new current solution. The search continues within the set of solutions $N_1(P)$. On the other hand, if the search in the set $N_k(P)$ does not produce any better solution, the search continues in the set $N_{k+1}(P)$, where $1 \leq k \leq |P| - 1$. The search ends after k exceeds the maximum allowed value, i.e. |P| in our case.

The p-second center problem is the generalization of the p-center problem. Mladenovic et al. (2004) [15] introduced an efficient VNS algorithm for the p-center problem. To offer a solution to the p-second center problem, we took advantage of the original algorithm from [15] with a simple modification so as to minimize not the maximum distance from the users to the closest center but the maximum sum of the distances to the closest and the second closest centers.

Basically, the VNS algorithm in each of its iterations tries to improve the current solution P. It consists of alternating procedures of random selection of a solution from the $N_k(P)$ neighborhood and a local search of the selected solution in order to find the local optimum. To explain the proposed implementation, let us consider an example of the p-second center problem with n = 15 users and p = 3 centers (Fig. 1). The current solution to the problem is a set of centers P = {1, 2, 11}.

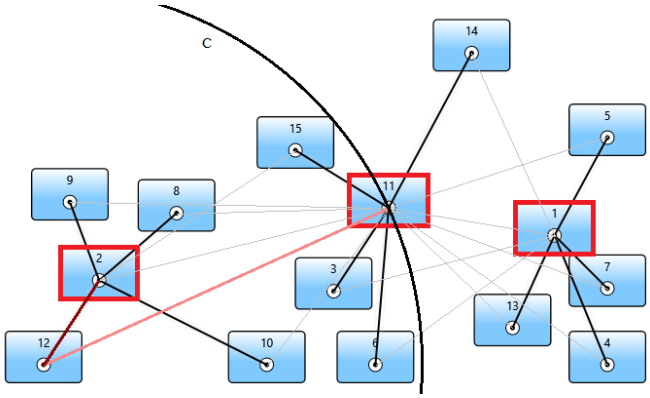


Fig. 1. Example of the p-second center problem with n = 15 and p = 3; the current solution is P = {1, 2, 11}

In Fig. 1, users are connected by darker lines (edges) to their closest, and with lighter to the second closest centers. The distances to the centers and the weights of edges correspond to the lengths of the edges. The user with the largest sum of distances to the appropriate centers is user 12. Let us call it a critical user, $u_c = 12$. The objective function value is just determined by the sum of the distances from critical user to its closest and the second closest center (centers 2 and 11). In order to improve the current function value, it is needed to reduce the distance from the critical user to the closest or/and the second closest center. To that end, it is necessary to find a new center that is closer to the critical user than the second closest center.

Property 1. Let u_c be a critical user, $c_1(u_c)$ its closest, and $c_2(u_c)$ the second closest center in the current solution P, ($c_1(u_c) \in P$, $c_2(u_c) \in P$). Then, if there is a better solution in the neighborhood $N_1(P)$ than the current solution P, the new center c_{in} ($c_{in} \neq c_1(u_c)$) must be closer to the critical user u_c than its previously second closest center $c_2(u_c)$.

Proof. Let f(P) be the function value for the current solution P and $f(u_i)$ the function value for the user u_i; $d(u_i, u_j)$ represents the shortest distance between the u_i and u_j nodes. Then, it applies:

$$f(P) = \max_{i=1,\dots,n} f(u_i) = f(u_c) = d\big(u_c, c_1(u_c)\big) + d\big(u_c, c_2(u_c)\big) \quad (5)$$

The objective function value is determined by the function value for the critical user. Therefore, in order to reduce the objective function value, it is necessary to include a new center c_{in} into the current solution, so that:

$$d\big(u_c, c_1(u_c)\big) + d(u_c, c_{in}) < d\big(u_c, c_1(u_c)\big) + d\big(u_c, c_2(u_c)\big) \quad (6)$$

i.e.

$$d(u_c, c_{in}) < d\big(u_c, c_2(u_c)\big) \quad (7)$$

Corollary. If the critical user is not unique, Property 1 should be applied for each of the critical users, i.e. $(\forall u \in U_c)\left(d(u, c_{in}) < d\big(u, c_2(u_c)\big)\right)$, where U_c is a set of all critical users. Otherwise, there is not any solution in the neighborhood $N_1(P)$ better than the current solution P.

Now, we can show how the cardinality of the set of potentially new centers decreases and thus accelerates convergence toward the local minimum. Previous property allows us to reduce the size of the complete $p * (n - p)$ neighborhood of P to $p * |N(u_c)|$, where

$$N(u_c) = \{u | d\big(u_c, c_1(u_c)\big) + d(u_c, u) < f(P)\} \quad (8)$$

Moreover, this size decreases with subsequent iterations and the speed of convergence to a local minimum increases on average. However, the acceleration reduces only the constant in the heuristic's complexity, but not its worst-case behavior.

Based on Property 1, we reduce the search just to the nodes u which meet the condition $d(u_c, u) < d(u_c, c_2(u_c))$. In the specific case from Fig. 1, the set of potentially new centers consists of nodes within the circle C (nodes closer to the critical user 12 than the second-closest center). Let us assume that node 3 is chosen as the new center c_{in}. The new center certainly reduces the distance from the critical user 12 to the assigned centers, but whether the objective function value will also be reduced depends on the users who lose one of their centers. If these users keep the sum of distances to the newly assigned centers at a level lower than the previous objective function value, the

objective function value will be improved. The *Move* method (Algorithm 1) identifies a center from the currently searched solution that should be replaced with a new one in order to maximally improve the current solution, and also calculates the new objective function value. In fact, instead of checking all p possible center exclusions, in the *New center* step, it is calculated only function values when one of the assigned centers is deleted for each of the users. The values are stored in the corresponding elements of the array z. Thereafter, the optimal center to be deleted is found based on the z value, as the one corresponding to the minimum value of the objective function (the *Best deletion* step). In this way, the time complexity is reduced from O(pn) to O(n) + O(p) ≈ O(n). Finally, in the *Function calculation* step, the new objective function value is calculated. In the worst case, the time complexity of the *Move* method is O(n).

It is assumed that the current solution P represents the first p elements of the array x_{cur}, while the last $n - p$ elements contain the rest of the users. The index of the new center is denoted by c_{in}. Additionally, the suggested algorithm uses the following structures:

- *dist(u, v)* – the shortest distance between the u and v nodes;
- *c1(u)* – the closest center of the user u;
- *c2(u)* – the second closest center of the user u;
- *c3(u)* – the third closest center of the user u;
- *z(v)* – the maximum function value among all users to whom center v was assigned either as the closest or the second closest center after the center v was removed.

The proposed implementation relies on the algorithms and data structures from the paper [15], provided that the solution is extended with the array c_3 which contains the third closest centers of all users. The additional structure and extensions of the algorithms are conditioned by the nature of the p-second center problem, i.e. by the requirement that the new second closest center is known in advance if one of the two closest centers is removed from the current solution. The extensions do not affect the correctness and efficiency of the algorithm, so the theoretical discussion and properties from the paper [15] remain fully applicable. Note that authors in [15] were analyzing bipartite graph. They did that in order to make clear difference between centers and users. In this paper there is no such assumption, but it does not affect the correctness of the algorithm. The problem defined over the bipartite graph is equivalent to the plane general graph where the weight of edge between the user and center at the same location is 0. Therefore, we just generalized the algorithm and discussion from [15] to plane general graphs and expanded the solution with the auxiliary structure for the third-closest centers to be able to serve the users if the closest centers have failed. The algorithm [15] assigns only one center to each of the users. Therefore, we implemented new algorithm able to find backup center in case that the closest center is broken. Below, we give the pseudocode for the remaining methods that implement the suggested algorithm.

The *Update* method (Algorithm 2) uses the structures *c1*, *c2* and *c3* which represent the lists of the closest and the second and third closest centers of all users. All of them are both input and output values, while c_{in} (the index of the new center) and c_{out} (the index of the center to be deleted from the current solution), along with the current solution x_{cur} represent only input values. The users in the current solution are iterated one by one and if any of their closest centers is deleted (c_{out}) or the new center c_{in} becomes one of the closest, corresponding *c1*, *c2* and *c3* arrays will be updated. The

worst-case complexity of the method *Update* is O(n log n) when a heap data structure is used for updating the third closest centers.

Algorithm 1. 1-interchange move in the context of the p-second center problem

Move(x_{cur}, c_{in}, c1, c2, c3)
Initialization:
Set $z(x_{cur}(i)) \leftarrow 0$ for all $i = 1, ..., p$

New center:
$in \leftarrow x_{cur}(c_{in})$
For Each *user = $x_{cur}(1), \ldots, x_{cur}(n)$*
  If *dist(user, in) < dist(user, c2(user))*
    ****in* as a new closest or second-closest center****
    z(c1(user)) ← max(dist(user, in) + dist(user, c2(user)), z(c1(user)))
  Else
    ****user keeps the same centers****
    z(c1(user)) ← max(min(dist(user, in), dist(user, c3(user))) + dist(user, c2(user)), z(c1(user)))
    z(c2(user)) ← max(min(dist(user, in), dist(user, c3(user))) + dist(user, c1(user)), z(c2(user)))
  End If
End For Each

Best deletion:
min ← ∞
For Each *i = {1, ..., p}*
  If *min > $z(x_{cur}(i))$*
    min ← $z(x_{cur}(i))$
    $c_{out} \leftarrow i$
  End If
End For Each

Function calculation:
$f_{cur} \leftarrow 0$
out ← $x_{cur}(c_{out})$
For Each *user = $x_{cur}(1), \ldots, x_{cur}(n)$*
  If *c1(user) = out*
    c1 ← c2(user)
    c2 ← c3(user)
  Else
    c1 ← c1(user)
    c2 ← c2(user) ***if*** *c2(user) ≠ out* ***else*** *c3(user)*
  End If
  f ← dist(user, c1) + dist(user, in) ***if*** *dist(user, in) < dist(user, c2)*
    else *dist(user, c1) + dist(user, c2)*
  $f_{cur} \leftarrow max(f, f_{cur})$
End For Each

Return *f_{cur}, c_{out}*

Algorithm 2. Updating the first, the second and the third-closest center

```
Update(x_cur, c_in, c_out, c1, c2, c3)
in ← x_cur(c_in)
out ← x_cur(c_out)
For Each user = x_cur(1), …, x_cur(n)
    **for users whose center is deleted, find new one**
    If c1(user) = out
        If dist(user, in) ≤ dist(user, c2(user))
            c1(user) ← in
        Else
            c1(user) ← c2(user)
            If dist(user, in) ≤ dist(user, c3(user))
                c2(user) ← in
            Else
                c2(user) ← c3(user)
                **find third closest center for the user**
                c3(user) ← select center
                           from {x_cur(1), …, x_cur(p)} ∪ {in} \ {c1(user), c2(user), out}
                           where d(user, center) is minimum
            End If
        End If
    Else
        If c2(user) = out
            If dist(user, in) ≤ dist(user, c1(user))
                c2(user) ← c1(user)
                c1(user) ← in
            Else
                If dist(user, in) ≤ dist(user, c3(user))
                    c2(user) ← in
                Else
                    c2(user) ← c3(user)
                    **find third closest center for the user**
                    c3(user) ← select center
                               from {x_cur(1), …, x_cur(p)} ∪ {in} \ {c1(user),c2(user), out}
                               where d(user, center) is minimum
                End If
            End If
        Else
            If dist(user, in) ≤ dist(user, c1(user))
                c3(user) ← c2(user)
                c2(user) ← c1(user)
                c1(user) ← in
            Else
                If dist(user, in) ≤ dist(user, c2(user))
                    c3(user) ← c2(user)
                    c2(user) ← in
                Else
                    If dist(user, in) ≤ dist(user, c3(user))
                        c3(user) ← in
                    Else If c3(user) = out
                        **find third closest center for the user**
                        c3(user) ← select center
                                   from {x_cur(1), …, x_cur(p)} ∪ {in} \ {c1(user), c2(user), out}
                                   where d(user, center) is minimum
                    End If
                End If
            End If
        End If
    End If
End For Each
Return c1, c2, c3
```

Algorithm 3. The vertex substitution local search for the p-second center problem

LocalSearchVertexSubstitution(x_{cur}, c1, c2, c3, u_c, f_{cur})
Main loop:
While *True*
    $f' \leftarrow \infty$
    For Each *in = p +1, ..., n*
        Find the optimal objective function value improvement within $N_1(x_{cur})$ neighborhood:
        If $d(u_c, x_{cur}(in)) < d(u_c, c2(u_c))$
            f, *out* $\leftarrow$ *Move*(x_{cur}, *in*, c1, c2, c3)
            If $f < f'$
                $f' \leftarrow f$
                $c_{in} \leftarrow in$
                $c_{out} \leftarrow out$
            End If
        End If
    End For Each

    If $f_{cur} \leq f'$
        There was not found improvement in the neighborhood:
        Break Main loop
    End If

    Update(x_{cur}, c_{in}, c_{out}, c1, c2 ,c3)
    $f_{cur} \leftarrow f'$
    $x_{cur}(c_{in}) \leftrightarrow x_{cur}(c_{out})$
    $u_c \leftarrow$ ***select*** *user* ***from*** $x_{cur}(1), \ldots, x_{cur}(n)$
        where *d(user, c1(user)) + d(user, c2(user)) is maximum*
End While
Return x_{cur}, u_c, f_{cur}

The Local Search Vertex Substitution method (Algorithm 3), relying on Property 1, accelerates the convergence towards the local optimum. However, it reduces the search space and constant complexity factor, but not the time complexity in the worst case. The complexity of one iteration of the *Main loop* in the worst case remains $O(n^2) + O(n \log n) + O(n) \approx O(n^2)$. The input values are the current solution, the c1, c2 and c3 arrays, the critical user and the current objective function value. The local search method, using the *Move* method, iteratively finds the optimal pair of centers to be exchanged, which results in the largest reduction in the objective function value. The method is executed as long as it is possible to find such a pair of centers. In the end, the method provides a new current solution, a new critical user and a new objective function value.

After the iteration of the proposed VNS implementation, i.e. of the previously described algorithms, over the problem from Fig. 1, we obtain a new current solution P = {1, 2, 3} (Fig. 2). Center 3 is included in the solution, and center 11 is deleted. The new critical user is $u_c = 14$, the center closest to it is center 1, while the second closest center is center 3. The objective function value is reduced, i.e. $d(14, 1) + d(14, 3) < d(12, 2) + d(12, 11)$.

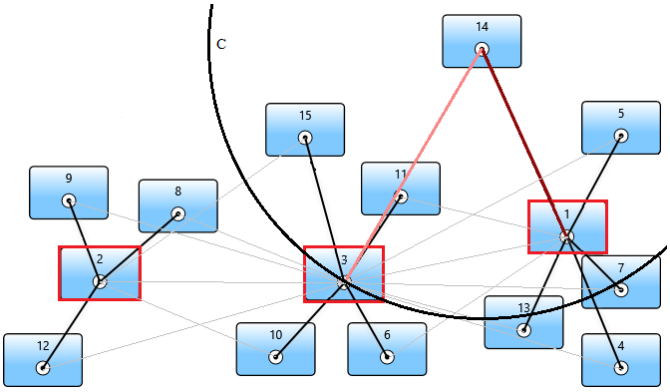


Fig. 2. Example of the p-second center problem with n = 15 and p = 3; the new solution is P = {1, 2, 3}

The pseudocode for the VNS algorithm for the p-second center problem is given in Algorithm 4. The initial value of the variable k is 1. This means that a new solution is searched in the N_1 neighborhood of the initially current solution. In the *Shaking operator* step, the new solution is selected from the N_k neighborhood of the current solution. Same as in the local search method, based on Property 1, the size of the neighborhood N_k set is reduced. It is selected one of the solutions which contain a new center that is closer to the critical user u_c* than the previous second closest center $c2(u_c*)$. After that, the chosen solution x_{cur} becomes the current solution for the local search method (the *Local search* step). If the local search finds an equal or a better solution than x_{cur}, the new solution is set as the "currently optimal" x_{opt}, and the search process restarts, $k = 1$. Otherwise, the value of k is incremented and the search process continues within the set N_{k+1}. If k has reached the maximum value k_{max}, k is set to *1* and the search process is restarted again. The *Main step* is repeated until the maximum allowed execution time t_{max} is reached. The algorithm returns the best solution found during the search process.

Algorithm 4. Shaking procedure for the p-second center problem

VariableNeighborhoodSearch(k_{max}, t_{max})
Initialization:
Randomly initialize x_{opt}; according to x_{opt} initialize arrays c1, c2 and c3, f_{opt}, u_c;*
copy initial solution into the current one, i.e., copy f_{opt}, x_{opt}, c1, c2, c3 and u_c into f_{cur}, x_{cur}, $c1_{cur}$, $c2_{cur}$, $c3_{cur}$ and u_{cur}* respectively.*
Repeat the Main step until the stopping condition is met (e.g., time ≤ t_{max})
Main step:
k ← 1
***While** k ≤ k_{max}*
Shaking operator:
****generate a solution at random from kth neighborhood****
***For Each** j = 1, ..., k*
***1.** Take center c_{in} to be inserted at random **if** $d(u_{cur}*, x_{cur}(c_{in})) < d(u_{cur}*, c2_{cur}(u_{cur}*))$;*
***2.** Find center c_{out} to be deleted at random;*
***3.** Update x_{cur}, $c1_{cur}$, $c2_{cur}$ and $c3_{cur}$,i.e., execute:*
Update(x_{cur}, c_{in}, c_{out}, $c1_{cur}$, $c2_{cur}$, $c3_{cur}$)
$x_{cur}(c_{in}) \leftrightarrow x_{cur}(c_{out})$
***4.** Update f_{cur} and u_{cur}* according to x_{cur}, $c1_{cur}$, $c2_{cur}$ and $c3_{cur}$*

End For Each
Local search:
If *any potentially better solution found*
$x_{cur}, u_{cur}^*, f_{cur} \leftarrow LocalSearchVertexSubstitution(x_{cur}, c1, c2, c3, u_{cur}^*, f_{cur})$
Move or not:
If $f_{cur} \leq f_{opt}$
****Save current solution as the optimal; return to N_1****
$x_{opt} \leftarrow x_{cur}; f_{opt} \leftarrow f_{cur}; u_c^* \leftarrow u_{cur}^*; c1 \leftarrow c1_{cur}; c2 \leftarrow c2_{cur}; c3 \leftarrow c3_{cur}$
$k \leftarrow 1$
Else
**** There was not found better solution; change the neighborhood****
$x_{cur} \leftarrow x_{opt}; f_{cur} \leftarrow f_{opt}; u_{cur}^* \leftarrow u_c^*; c1_{cur} \leftarrow c1; c2_{cur} \leftarrow c2; c3_{cur} \leftarrow c3$
$k \leftarrow k + 1$
End If
End If
End While
Return x_{opt}, f_{opt}

3. Quasi-Exact Algorithm

Property 2. Let u_c be a critical user, $c_1(u_c)$ its closest and $c_2(u_c)$ the second closest center in the current solution P, ($c_1(u_c) \in P$, $c_2(u_c) \in P$). Then, if in the set of potentially new centers there is no center c_{in} that is closer to the critical user u_c than its second closest center $c_2(u_c)$, the current solution P is the optimal solution for the p-second center problem.

Proof. Based on Property 1, it applies $d(u_c, c_{in}) < d(u_c, c_2(u_c))$. If it is not possible to find a new center c_{in} that satisfies previous inequality, it means that it is not possible to improve the current solution, i.e. the current solution is the optimal solution to the p-second center problem.

Corollary. If there is not any center to be opened in order to improve the objective function value related to the critical user u_c, the critical user has already been selected to be a center in the current solution P.

Property 2 can be considered as a generalization of Property 1, i.e. if there is a better solution P' than the current solution P, the solution P' has to contain a new center c_{in} ($c_{in} \notin P$) which is closer to the critical user u_c than the second closest center $c_2(u_c)$ ($c_2(u_c) \in P$).

Based on Property 2, we propose a modification of the VNS algorithm from the previous section, which might be able to recognize the exact solution to the p-second center problem. The idea is, to check if there is at least one center that potentially improves the current solution. If there is not any, the execution is stopped and the algorithm returns the current solution as the optimal one (Algorithm 5).

As future work, it would be interesting to try to accelerate the convergence towards an optimal solution by searching the history of the previous solutions. For example, the solution P1 = {1, 2, 11} in Fig. 1 is improved by adding center 3 and, after deleting center 11, it becomes {1, 2, 3}. The set of all potentially new centers consists of the centers within the circle C, i.e. $C1_{in}$ = {3, 6, 8, 9, 10, 12, 15}. So, the solution {1, 2} is expanded by one of the centers from the $C1_{in}$ set. In Fig. 2, the current solution is P2 = {1, 2, 3}, and the set of potentially new centers $C2_{in}$ = {5, 11, 14, 15}. In case of

deleting node 3, the current solution might be expanded by centers from the set $C2_{in} \cap C1_{in} = \{15\}$, which reduces the cardinality of the search set from 4 to 1. A new structure can be used to store the previous solutions, e.g. a tree that stores in its leaves a set of all potential centers that can extend the solution defined by nodes on the path from the root to that particular leaf of the tree.

Algorithm 5. Shaking procedure for the p-second center problem

VariableNeighborhoodSearch(k_{max}, t_{max})
Initialization:
…
Repeat the Main step until the stopping condition is met (e.g., time ≤ t_{max})
Main step:
$k \leftarrow 1$
While $k \leq k_{max}$
Shaking operator:
****generate a solution from kth neighborhood****
count ← *0*
For Each *j = 1, ..., k*
c_{in} ← ***find*** *center at random* ***from*** $x_{cur}(p + 1), \ldots, x_{cur}(n)$
where $d(u_{cur}^*, x_{cur}(center)) < d(u_{cur}^*, c2_{cur}(u_{cur}^*))$

If c_{in} *not found*
If *count = 0*
****the optimal solution has been found****
Return x_{opt}, f_{opt}
End If
Else
5. *Increment count*
6. *Find center to be deleted (c_{out}) at random;*
7. *Update x_{cur}, $c1_{cur}$, $c2_{cur}$ and $c3_{cur}$,i.e., execute:*
Update(x_{cur}, c_{in}, c_{out}, $c1_{cur}$, $c2_{cur}$, $c3_{cur}$)
$x_{cur}(c_{in}) \leftrightarrow x_{cur}(c_{out})$
8. *Update f_{cur} and u_{cur}^* according to x_{cur}, $c1_{cur}$, $c2_{cur}$ and $c3_{cur}$*
End If
End For Each
Local search:
…
End While

4. Results

The algorithm was implemented in the C++ programming language, and all tests were performed on an Intel Core i7-8700K (3.7 GHz) CPU with a 32 GB RAM configuration. For testing purposes, we downloaded an OR-Library [2] data set containing 40 test instances with 100 to 900 nodes and p between 5 and 200 (not more than n/3, where n is the number of nodes). Additionally, we generated two data sets with larger test instances. The first contains 44 instances with 1000 to 2500 nodes and p between 5 and 200. The second contains 48 instances (500–1000 nodes) defined over graphs with different densities (50%–80%) and p between 5 and 200.

Each of the proposed algorithms was executed 20 times on each testing instance, always starting from a different initial solution. Different combinations of the

parameters $k_{max} = p/4$, $k_{max} = p/2$, $k_{max} = p$, as well as $t_{max} = n$ and $t_{max} = 2n$ were tested. It turned out that the results were slightly better with higher values of the parameter k_{max}. On the other hand, the algorithm usually found the best solution much before the execution time limit expired. Therefore, we decided to present the results only for $k_{max} = p$ and $t_{max} = n$ seconds. The summary results are presented in the following tables, while the detailed results are available in Appendix 2.

4.1. Test results over the OR-Library instances

Table 1 shows the results obtained for original OR-Library test instances. The first column of the table contains the name of the instance, the next three columns represent the value of p (number of centers), n (number of users) and m (number of graph edges). The columns "Best Value", "AVG Value" and "Worst Value" show, respectively, the best, average, and worst solution value that the algorithm found during the 20 executions. The "Time" column (or time-to-target) shows the average time in seconds that was needed to find the best solution for the first time. "Time" does not represent the total execution time. The algorithm is executed until the time limit is reached, i.e. *n* seconds. The last column "#Best" shows the number of times the algorithm found the best solution during the 20 executions. Also, in Appendix 2 we included additional columns containing percentage gaps of the average and the worst solution compared to the best known solution presented in the column "Best(-Known) Value". Since OR-Library instances have not yet been used to test the p-second center problem, we take the best solutions found by our algorithm as the best known solutions.

Table 1. Results for multi-executed OR-Library test instances

	P	N	M	Best Value	AVG Value	Worst Value	Time	#Best
pmed1-pmed5	5-33	100	200	193.80	193.80	193.80	3.30	20.00
pmed6-pmed10	5-67	200	800	120.00	120.02	120.20	4.33	19.60
pmed11-pmed15	5-100	300	1800	83.80	83.80	83.80	2.23	20.00
pmed16-pmed20	5-133	400	3200	65.00	65.00	65.00	34.49	20.00
pmed21-pmed25	5-167	500	5000	58.60	58.61	58.80	48.15	19.80
pmed26-pmed30	5-200	600	7200	56.00	56.00	56.00	35.03	20.00
pmed31-pmed34	5-140	700	9800	53.00	53.00	53.00	19.83	20.00
pmed35-pmed37	5-80	800	1280	51.67	51.83	52.00	149.49	16.67
pmed38-pmed40	5-90	900	16200	54.67	54.98	55.00	29.12	13.67
AVG							**31.32s**	**19.20**

Based on the results, we noticed that most of the smaller instances were solved very quickly. In merely 3 out of 15 cases for n ≤ 300, the average time-to-target was more than 4 seconds. As the size of the test instance increases, so does the time needed to find the best solution. The average time-to-target over a complete test set is 31.32 seconds. On the other hand, in terms of the algorithm stability and solution quality, the best-known solution was not found by each of the 20 executions only for 4 out of 40 instances. Moreover, in just two examples (pmed37 and pmed40), the best solution was found less than eighteen times in 20 executions. In the case of the pmed40 instance, only once the best solution was found, but in all other executions a solution was found the value of which is higher by only 1 than the best value. The algorithm found the best known solution on average in 19.20 out of 20 cases or in 96% of the cases. Regarding the deviation of the average and worst solutions, from Table 6 (Appendix 2) it can be concluded that there are no significant differences between the worst/average and best values, only 0.23% and 0.12% on average, respectively.

To verify the quality of our sophisticated local search method, we applied a simple local search method and tested the VNS algorithm with the same parameters and test data. Unlike the proposed solution, the simple local search method does not filter the centers that do not satisfy Property 1. The method only searches for centers that improve the current solution, as shown in Algorithm 6.

Algorithm 6. The simple local search method

SimpleLocalSearch(x_{cur})
Main loop:
While *True*
  $P \leftarrow [x_{cur}(1), ..., x_{cur}(p)]$
  $(c_{in}, c_{out}) \leftarrow$ *select (user, center)*
    where user in $[x_{cur}(p+1), ..., x_{cur}(n)]$ *and*
      center in $[x_{cur}(1), ..., x_{cur}(p)]$ *and*
      $P \cup \{c_{in}\} \setminus \{ c_{out} \}$ *is better than P*
  If (c_{in}, c_{out}) *found*
    Exchange($x_{cur}(c_{in}), x_{cur}(c_{out})$)
  Else
    Break Main loop
  End If
Return x_{cur}

Algorithm 6 was also executed 20 times and the results are presented in Table 2. For the VNS which uses a simple local search algorithm, we show the best solution obtained in 20 executions (the "Best Found Value" column) and the average time to find the best solution (the "Time" column). The "#Best-Known" column contains the number of algorithm executions with a simple local search method that resulted in finding the best known solution, i.e. the best solution found by the algorithm with a sophisticated local search. Finally, the last column gives the percentage deviation of the best solution found by the algorithm with a simple local search method from the best-known solution. The percentage gap is calculated as $\frac{Best\ found - Best\ known}{Best\ known} * 100$.

Table 2 and Table 7 in Appendix 2 show that a VNS with a simple local search method is not capable of providing satisfactory results. The algorithm did not find the best known solution for 14 out of 40 instances, having found the best known solution only in 8.43 out of 20 cases on average. It turns out that the value of the best found solution of the VNS algorithm with a sophisticated local search method is better by

5.47% on average. Moreover, several instances, such as pmed19, pmed24, pmed33, pmed37 and pmed40, resulted in a significantly larger deviation from the best known solution. Also, the average time to find the best solution increased almost eight times, for up to 246.92 seconds. All these findings point to the advantages of a sophisticated local search algorithm, i.e. much better solutions for a significantly less CPU time.

Table 2. VNS with simple local search

	P	N	Best-Known Value (1)	Best-Found Value (2)	Time	#Best-Known	Gap $\left[\frac{(2)-(1)}{(1)} * 100\right]$
pmed1-pmed5	5-33	100	193.80	193.80	16.15	16.80	0.00
pmed6-pmed10	5-67	200	120.00	120.20	64.97	11.60	0.24
pmed11-pmed15	5-100	300	83.80	86.00	183.92	5.60	3.80
pmed16-pmed20	5-133	400	65.00	68.60	260.17	6.00	7.72
pmed21-pmed25	5-167	500	58.60	62.00	337.51	6.60	8.41
pmed26-pmed30	5-200	600	56.00	57.40	340.79	5.40	3.63
pmed31-pmed34	5-140	700	53.00	55.25	328.85	10.25	6.43
pmed35-pmed37	5-80	800	51.67	55.67	487.02	1.67	12.12
pmed38-pmed40	5-90	900	54.67	58.33	360.99	10.33	12.64
AVG					**246.92s**	**8.43**	**5.47%**

Table 3 contains the results of the execution of the quasi-exact algorithm (Algorithm 5) over all the OR-Library test instances. Compared to the previous ones, the table has been expanded with the "Exact Value" column, which shows whether the exact solution has been found. The algorithm was also executed 20 times with the same parameter values (k_{max} = p and t_{max} = n seconds).

Table 3 shows that the quasi-exact algorithm, despite reducing on average the time for finding the best solution, is not as effective as the initial algorithm. It found the best solution in 9.63 out of 20 executions on average. There were several instances (pmed18, pmed19, pmed23 and pmed40) for which no best-known solution was found. As for the best solutions, the initial algorithm is averagely more successful only for 0.24%. On the other hand, the algorithm managed to identify optimal solutions for 12 out of 40 instances from OR-Library test set, which is indicated in the last column of Table 3. In the column "#Best-Known" is reported how many times the algorithm found the best solution. It is important to note that mostly the larger instances, i.e. problems with a greater number of centers (higher p values), were solved exactly. In case of higher p values, it is more likely that there is a center which is a critical user at the same time and it is not possible to find a closer backup center to be included into the current solution in order to reduce the objective function value (Corollary of Property 2).

Table 3. Results of the quasi-exact VNS algorithm for multi-executed OR-Library test instances

	P	N	Best-Known Value (1)	Best-Found Value (2)	Time	#Best-Known	Gap $\left[\frac{(2)-(1)}{(1)} * 100\right]$	Exact Value
pmed1	5	100	268	268	0.02	3	0	
pmed2	10	100	220	220	0.11	4	0	
pmed3	10	100	208	208	0.09	3	0	
pmed4	20	100	163	163	0.07	1	0	
pmed5	33	100	110	110	0.02	16	0	
pmed6	5	200	180	180	0.23	10	0	
pmed7	10	200	143	143	0.51	3	0	
pmed8	20	200	122	122	0.48	2	0	
pmed9	40	200	85	85	0.20	3	0	
pmed10	67	200	70	70	0.10	20	0	✓
pmed11	5	300	125	125	0.52	16	0	
pmed12	10	300	112	112	1.20	5	0	
pmed13	30	300	78	78	1.00	3	0	
pmed14	60	300	60	60	0.63	1	0	✓
pmed15	100	300	44	44	0.36	20	0	✓
pmed16	5	400	98	98	1.61	16	0	
pmed17	10	400	83	83	4.26	9	0	
pmed18	40	400	62	63	2.70	0	1.61	
pmed19	80	400	42	43	1.51	0	2.38	
pmed20	133	400	40	40	0.81	20	0	✓
pmed21	5	500	85	85	5.00	14	0	
pmed22	10	500	80	80	20.38	3	0	
pmed23	50	500	49	50	6.17	0	2.04	
pmed24	100	500	35	35	2.37	1	0	
pmed25	167	500	44	44	1.48	20	0	✓
pmed26	5	600	80	80	20.52	5	0	
pmed27	10	600	67	67	20.04	7	0	
pmed28	60	600	57	57	2.86	20	0	✓
pmed29	120	600	36	36	3.02	20	0	✓
pmed30	200	600	40	40	3.06	20	0	✓
pmed31	5	700	64	64	7.32	20	0	
pmed32	10	700	72	72	4.58	20	0	✓
pmed33	70	700	35	35	16.87	8	0	
pmed34	140	700	41	41	4.50	20	0	✓
pmed35	5	800	64	64	66.86	6	0	
pmed36	10	800	58	58	50.63	8	0	
pmed37	80	800	33	33	24.95	2	0	✓
pmed38	5	900	61	61	32.37	16	0	
pmed39	10	900	74	74	9.82	20	0	✓
pmed40	90	900	29	30	18.08	0	3.45	
AVG					8.43s	9.63	0.24%	**Total 12**

4.2. Test results over larger instances

Encouraged by the obtained results, in order to further assess the performance of the algorithm, we generated test instances with 1000, 1500, 2000, and 2500 nodes, as well as new instances defined over the graphs with up to 1000 nodes with various density values. New test instances are generated as random k-regular graphs with predefined node and edge count. The initial VNS algorithm was again executed 20 times with the same parameter values ($k_{max} = p$ and $t_{max} = n$ seconds) over the new test examples and the results are presented in the following tables and Appendix 2.

Table 4. Results for large test instances

	P	N	M	Best Value	AVG Value	Worst Value	Time	#Best
rndkreg1-rndkreg11	5-200	1000	50000	14.82	14.90	15.00	82.85	18.36
rndkreg12-rndkreg22	5-200	1500	112500	12.45	12.56	12.64	183.81	17.82
rndkreg23-rndkreg33	5-200	2000	200000	11.64	11.64	11.64	235.01	20.00
rndkreg34-rndkreg44	5-200	2500	312500	10.27	10.51	10.64	576.65	15.18
AVG							**269.58s**	**17.84**

Table 4 shows the execution results of the larger test instances and it is noticeable that the algorithm was not as successful as in other instances. It found the best solution in 17.84 out of 20 executions on average. There have been several instances (rndkreg13, rndkreg34 and rndkreg38) for which the best solutions were found only three or fewer times, while the absolute deviation of the worst and best solutions was not higher than 1. The average time-to-target was 269.58 seconds.

In addition to the previously explained columns, Table 5 contains a new column ("Density"), which shows density of the graph over which the test instance is defined.

Based on the results from Table 5, and taking into account the slightly larger instances of the problem, the average time-to-target increased to 43.07 seconds as compared to the OR-Library instances. The average number of finding the best solution dropped to 18.46 out of 20 executions. On the other hand, the density of the graph did not affect the efficiency of the algorithm. This was to be expected, given that the algorithm does not take into account the number of graph edges, but generates new search solutions from appropriate neighborhoods by a simple replacement of nodes (centers). There were only 8 (out of 48) instances for which the best solution was not found in each of the 20 executions, but the absolute deviation of the worst and the best solution in all these cases was only 1**.**

We also compared the suggested algorithm for the pSCP with the results of the algorithms for the p-center [15], the p-next center [17] and the p-median [8] problems. The results are reported in Appendix 3.

Table 5. Results for test instances defined over graphs with different densities

	P	N	Density	Best Value	AVG Value	Worst Value	Time	#Best
rnddnskreg1-rnddnskreg4	5-200	500	50.10	8.00	8.00	8.00	13.07	20.00
rnddnskreg5-rnddnskreg8	5-200	500	60.12	7.00	7.00	7.00	4.35	20.00
rnddnskreg9-rnddnskreg12	5-200	500	80.16	6.00	6.23	6.25	17.13	15.50
rnddnskreg13-rnddnskreg16	5-200	600	50.08	7.25	7.36	7.75	67.27	17.75
rnddnskreg17-rnddnskreg20	5-200	600	60.10	7.00	7.00	7.00	6.45	20.00
rnddnskreg21-rnddnskreg24	5-200	600	80.13	5.75	5.99	6.25	51.02	15.25
rnddnskreg25-rnddnskreg28	50	800	50.06	6.75	6.83	7.00	97.05	18.50
rnddnskreg29-rnddnskreg32	5-200	800	60.08	6.50	6.50	6.50	37.24	20.00
rnddnskreg33-rnddnskreg36	5-200	800	80.10	6.00	6.00	6.00	28.90	20.00
rnddnskreg37-rnddnskreg40	5-200	1000	50.05	6.50	6.56	6.75	105.35	18.75
rnddnskreg41-rnddnskreg44	5-200	1000	60.06	6.50	6.50	6.50	27.76	20.00
rnddnskreg45-rnddnskreg48	5-200	1000	80.08	5.50	5.71	5.75	61.31	15.75
AVG			63.43%				**43.07s**	**18.46**

5. Conclusions

The paper discusses the p-second center problem, which is a generalization of the well-known and highly studied the p-center problem. The algorithm based on Variable Neighborhood Search method has been designed and implemented as a heuristic approach to problem solving. A solution to the p-second center problem represents the identification of the p centers for the purpose of minimizing the maximum distance from the n users to the closest and the second closest centers.

The proposed algorithm was tested on OR-Library instances from the literature with up to 900 nodes, and the obtained experimental results confirmed the high efficiency of the algorithm in a reasonable amount of time. The algorithm found the best solution on average in 19.20 out of 20 cases or in 96% of the cases. The paper also presents a modification of the initially proposed algorithm as a VNS implementation that is capable of identifying the optimal solutions to the p-second center problem. It was tested over the same OR-Library test set. It turned out that the quasi-exact algorithm on average yields worse results as compared to the initial algorithm, but it managed to identify 12 exact solutions out of 40 OR-Library instances.

To confirm the efficiency of the original algorithm, we also generated larger test instances with 1000, 1500, 2000, and 2500 nodes as well as instances defined over

graphs of varying densities (50%–80%). It was shown that the density of the graph did not affect the efficiency of the algorithm. The best solution was found for 40 out of 48 instances in each of the 20 executions of the algorithm, or in 92.29% of the cases on average. For larger instances of the problem (up to n=2500), the algorithm was not as successful, but it was still stable. It found the best solution in 89.20% executions with an absolute deviation of the worst solution being not higher than 1.

The proposed algorithm successfully solves the p-second center problem, but it would certainly be interesting to generalize the problem by taking into account more than 2 centers, i.e. apply the VNS heuristic to the p-α-closest center problem, where it would be necessary to minimize the sum of the distances to α closest centers. The proposed quasi-exact algorithm for the p-second center problem showed a 30% success rate in identifying exact solutions. As the topic of a future paper, it would be challenging to try to increase the success rate of the algorithm, for instance, by tracking the search history and comparing the current solution with potential extensions of the previous solutions.

Acknowledgment. This work has been partially supported by the Serbian Ministry of Education, Science and Technological Development through Mathematical Institute of the Serbian Academy of Sciences and Arts. This research has also been partially supported by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan, Grant No. BR10965172.

References

1. Albareda-Sambola, M., Hinojosa, Y., Marín, A., Puerto, J.: When centers can fail: a close second opportunity. Computers & Operations Research, 62, 145–156. (2015)
2. Beasley, J. E.: OR-Library: distributing test problems by electronic mail. Journal of the operational research society, 41 (11), 1069–1072. (1990)
3. Calik, H., Tansel, B. C.: Double bound method for solving the p-center location problem. Computers & Operations Research, 40, 2991–2999. (2013)
4. Daskin, M. S.: Network and discrete location: models, algorithms, and applications, 2nd edn. Wiley, Hoboken. (2013)
5. Davidovic, T., Ramljak, D., Selmic, M., Teodorovic, D.: Bee colony optimization for the p-center problem, Computers and Operations Research, 38, 1367–1376. (2011)
6. Elloumi, S., Labbé, M., Pochet, Y.: A new formulation and resolution method for the p-center problem. INFORMS Journal on Computing, 16, 84–94. (2004)
7. Hakimi, S. L.: Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Operations Research, 13 (3), 462-475. (1965)
8. Hansen, P., Mladenovic, N.: Variable neighborhood search for the p-median. Location Science, 5 (4), 207-226. (1997)
9. Hochbaum, D. S., Shmoys, D. B.: A best possible heuristic for the k-center problem. Mathematics of Operations Research, 10 (2), 180-184. (1985)
10. Ilhan, T., Pınar, M. C.: An efficient exact algorithm for the vertex p-center problem. Technical report, Department of Industrial Engineering, Bilkent University. (2001)
11. Kariv, O., Hakimi, S. L.: An algorithmic approach to network location problems. I: The p-Centers. SIAM Journal on Applied Mathematics, 37(3), 513-538. (1979)
12. López-Sánchez, A. D., Sánchez-Oro, J., Hernández-Díaz, A. G.: GRASP and VNS for solving the p-next center problem. Computers and Operations Research, 104, 295-303. (2019)
13. Minieka, E.: The m-center problem. Society for Industrial and Applied Mathematics, 12(1), 138–139. (1970)

14. Mladenovic, N., Hansen, P.: Variable neighborhood search. Computers and Operations Research, 24 (11), 1097–1100. (1997)
15. Mladenovic, N., Labbé, M., Hansen, P.: Solving the p-center problem with tabu search and variable neighborhood search. Networks 42 (1), 48–64. (2003)
16. Pullan, W.: A memetic genetic algorithm for the vertex p-center problem. Evol Comput, 16 (3), 417–436. (2008)
17. Ristic, D., Mladenovic, N., Todosijevic, R., Urosevic, D.: Filtered variable neighborhood search method for the p-next center problem. International Journal for Traffic and Transport Engineering, 11 (2), 294 - 309. (2021)

Dalibor Ristic graduated from Faculty of Electrical Engineering at Department of Computer Science, University of Nis, Serbia in 2007 and started his Ph.D. studies at the School of Computing, Union University, Belgrade, Serbia in 2013. From 2013 to 2017, he was a Teaching Assistant at the School of Computing, Union University where he gave lectures in Design and Analysis of Algorithms, Introduction to Programming, Object-Oriented Design and Methodology and Functional Programming. His main research areas are the development of complex graph algorithms and combinatorial optimization. Since 2008, he has worked as a software developer for a few software companies as a part of experienced teams that designed and developed complex systems in the fields of Energetics and Finance.

Dragan Urosevic is a research professor at the Mathematical Institute of the Serbian Academy of Sciences and Arts. He is a visiting professor at university in Novi Sad (Serbia). He received his B.Sc. degree at Faculty of Mathematics, University of Belgrade, in 1987. He received his M.Sc. degree at Faculty of Mathematics, University of Belgrade, in 1994, with the thesis “Heuristics for scheduling Parallel programs on Multiprocessor systems”. In 2004 he got PhD degree with the thesis “Solving problems on graphs by using Variable neighborhood search” at Faculty of Mathematics, University of Belgrade. Since 2001 he participated in a number of scientific projects, national, bilateral. He is a member of the Program Committees for several international conferences related to optimization and computer science fields. He also was the co-chair of the XIII Balkan Conference on Operational Research (BALCOR, Belgrade, Serbia, May 22-25, 2918). His main research interests include combinatorial optimization, mathematical programming, metaheuristics and computational complexity. He is a co-author of three chapters in monographs, more than 35 papers in refereed international journals, and more than 20 papers in international conference proceedings.

Nenad Mladenovic was born on April 28, 1951 in Jagodina, Serbia. He finished primary and secondary school in Belgrade. He graduated from the Department of Mathematics (major Cybernetics) at the Faculty of Natural Sciences and Mathematics, University of Belgrade in 1976. He received his master degree in 1982 from the Faculty of Organizational Sciences, University of Belgrade with a master's thesis titled “Comparative analysis of some nonlinear programming methods”. He defended his doctoral thesis “New nonlinear programming methods with application in location, allocation and transportation problems” in 1988 at the Faculty of Organizational Sciences. Dr Mladenović, as a visiting professor, visited several well-known world universities. From 2005-2013 he taught Operations Research, Heuristic Optimization

and Operations Management at the Faculty of Mathematics, University of Birmingham and Brunel University in London. From 2013-2016, as an international chair, he gave lectures on Mathematical Optimization Methods to doctoral students at the University of Valenciennes in France. Dr Nenad Mladenović was a very productive researcher, who gained an enviable world reputation in the field of operations research, dealing with problems of decision-making and management in complex systems. He published about 200 papers in scientific journals, 70 papers in edited conference proceedings and 40 books and chapters in scientific monographs.

Raca Todosijevic is an Associated Professor in Computer Science at the Polytechnic University-Hauts-de-France, Valenciennes, France. His principle research line is combinatorial optimization with the special emphasis on an efficient and effective design of heuristic and matheuristic approaches. He has published more than 30 journal papers. He has been a member of program committees of several international conferences related to the domain of his research. In 2016, he received EURO Doctoral Dissertation Award 2016.

Received: August 04, 2021; Accepted: September 18, 2022.

Computer Science and Information Systems 20(1):117–136 https://doi.org/10.2298/CSIS220325052Z

TS-GCN: Aspect-level Sentiment Classification Model for Consumer Reviews

Shunxiang Zhang[1,2], Tong Zhao[1,2], Houyue Wu[1,2], Guangli Zhu[1,2], and KuanChing Li[3]

[1] School of Computer Science and Engineering,
Anhui University of Science & Technology, 232001 Huainan, China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center,
WangJiang Road 5089, Hefei, 230088, Anhui, China
sxzhang@aust.edu.cn
zhaotongmail2022@163.com
why3664@163.com
glzhu@aust.edu.cn
[3] Department of Computer Science and Information Engineering (CSIE),
Providence University, 43301 Taichung, Taiwan
kuancli@pu.edu.tw

Abstract. The goal of aspect-level sentiment classification (ASC) task is to obtain the sentiment polarity of aspect words in the text. Most existing methods ignore the implicit aspects, resulting in low classification accuracy. To improve the accuracy, this paper proposes a classification model for consumer reviews, abbreviated as TS-GCN (Truncated history attention and Selective transformation network-Graph Convolutional Networks). TS-GCN can classify sentiment from both explicit and implicit aspects. Firstly, we process the text by the BERT model and the BiLSTM model to obtain the text features. Secondly, the GCN model completes explicit sentiment classification by training text features. Due to the lack of implicit words, the GCN model cannot classify implicit sentiments. Finally, we predict implicit words based on the TS model, which makes up for the deficiency of the GCN model and completes the sentiment classification of implicit words. TS-GCN is proved on several datasets in the consumer reviews field. The results of experiments show that the TS-GCN can improve the accuracy and F1 of ASC.

Keywords: consumer reviews; aspect-level sentiment classification (ASC); implicit aspect; GCN.

1. Introduction

ASC is one of the important research directions of natural language processing (NLP) [1]. The purpose of the classification is to get the sentiment tendency of the aspect words in the text [2-3]. The results of the classification can provide support for both customers' purchasing decisions and merchants' sales strategies. Due to these positive effects, ASC has become one of the popular research topics.

The earliest ASC methods were based on rules. In 2004, a rule-based ASC model was first proposed by Hu et al. [4]. Later, Nigam et al. [5] improved it by adding a topic

classifier to identify topic-specific sentiment expressions. Although rule-based methods are easier to interpret, the quality of classification is limited by whether the rules are well formulated or not. To address these problems, machine learning-based methods have started to receive widespread attention. Pang et al. [6] first used traditional machine learning algorithms to classify the sentiment polarity of movie reviews. The combination of naive bayesian classifier with support vector machine was proposed by Wang et al [7], which led to a better result for classification. Traditional machine learning methods rely on the selection of features. In contrast, neural network model can automatically generate features from texts. Ruidan et al. [8] proposed an interactive multitask learning network to improve the accuracy of classification by sharing information. A model framework for multi-task learning was demonstrated by Akhtar et al. [9] to accomplish the recognition of aspect words and prediction of sentiment polarity.

Although these advances have been made in ASC, there are still some problems to be studied. Aspect words include both explicit aspect words and implicit aspect words [10-11]. An example shown in Figure 1, "environment" is an explicit aspect word and "price" is an implicit aspect word. The implicit aspect word is the smallest object described by the opinion word but omitted in the sentence. There are a large number of implicit aspect words in the oral consumer reviews. Most of the existing ASC studies have only considered explicit aspects and ignored the implicit aspects. Different from previous work, our work classifies the sentiment in both explicit and implicit aspects. Adding sentiment classification of implicit aspects can improve the accuracy of ASC.

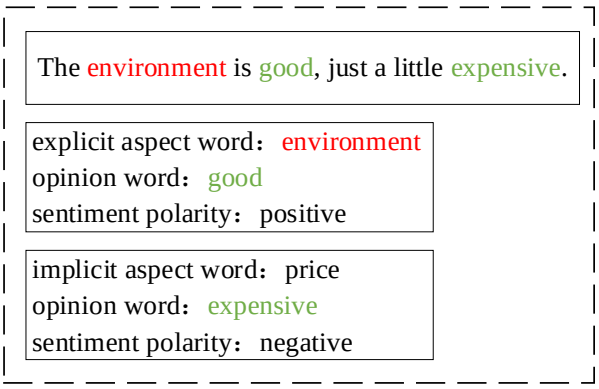


Fig. 1. An example of explicit and implicit aspect words

In the field of consumer reviews, an accurate ASC model needs to consider two aspects. Firstly, the model should accurately classify explicit aspects sentiment; Secondly, the model can classify implicit aspects sentiment when there are implicit aspect words in reviews. Based on the above two considerations, we propose a sentiment classification model named TS-GCN. The TS-GCN consists of the following three main parts, as shown in Figure 2.

(1) The extraction of text features. We use the BERT [12] model to code the semantics and location of consumer reviews and aspect words. After that, we input the encoded information into the BiLSTM model to extract text features.

(2) Explicit aspect sentiment classification. We construct the graph of text features in the form of points and edges. Then, we input text features into the GCN model training to complete explicit sentiment classification. Due to the lack of implicit aspect words, the GCN model cannot complete implicit aspect sentiment classification.

(3) Implicit aspect sentiment classification. Firstly, we input text features into the THA module in the TS [13] model to establish the mapping relationship between aspect words and opinion words. Secondly, we combine the mapping relationship with the results of explicit aspect sentiment classification and then input it into the STN module in the TS model. Finally, we use the TS model to predict the implicit aspect words and then complete the implicit aspect sentiment classification.

The advantage of TS-GCN is to consider the implicit aspect words in consumer reviews and propose a sentiment classification method from both explicit and implicit aspects.

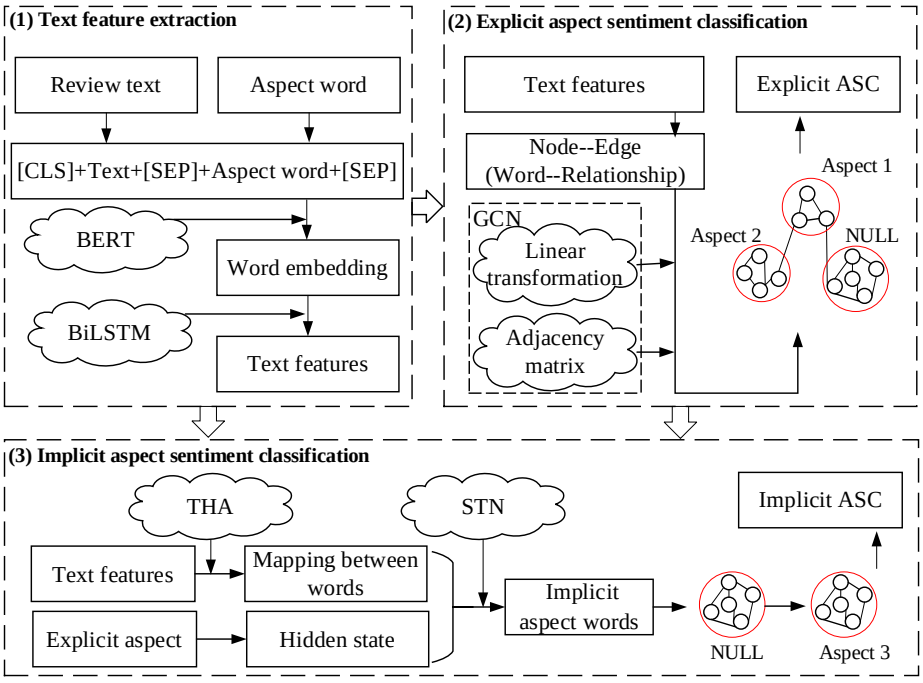


Fig. 2. TS-GCN frame diagram

This paper is organized according to the following sections. In Section 2, we present the related work of this paper. In Section 3, the content of TS-GCN is described in detail. In Section 4, we discuss the results of the experiments. Finally, in Section 5, we give the conclusions of this paper.

2. Related Work

In recent years, there are two main approaches for ASC. One is the classical neural network [14-17] and the other is the graph convolutional neural network (GCN) [18-19].

2.1. Classical Neural Networks

Among the classical neural network models, recurrent neural networks (RNN) [20] and convolutional neural networks (CNN) [21] are widely used for sentiment classification tasks. The powerful text feature extraction ability of RNN and CNN improves the accuracy of sentiment classification [22-23].

In the ASC task, Wang et al. [24] first combined RNN models with attention mechanisms and achieved good training results. However, this model is not good at handling complex text because it does not capture contextual features. Chen et al. [25] remedied this deficiency by proposing a memory recursive attention network called RAM. RAM uses a weighted attention mechanism to filter irrelevant encoded information and captures contextual features more accurately. Huang et al. [26] proposed an attention overload model called AOA. The AOA model was able to focus on the important sections of the text and improved the effect of complex text sentiment classification. After the good results achieved by RNN, researchers also further tried to use CNN to achieve sentiment classification. Xue et al. [27] used a gating mechanism to improve CNNs. This approach is simpler and more independent than models with attention mechanisms and LSTM while weakening the dependence of the model on time. Song et al. [28] used an attention encoder network to fill the gap that RNN models cannot handle the task in parallel. This approach achieves better extraction of aspect words and sentiment polarity in sentences. Fang et al. [29] proposed a target fusion sequence labeling neural network model called IO-LSTM, which can match aspect words and opinion words in sentences more accurately. Unlike the IO-LSTM, Huang et al. [30] used parametric filters to extract aspect features and achieved similarly good results on sentiment classification.

2.2. Graph Convolutional Neural Networks

Compared with classical neural networks, GCN [31] is more adept at dealing with the problem of relationships between entities. In recent years, many studies have shown that the sentiment classification results based on GCN are better than traditional neural networks.

Based on the GCN model, Wang et al. [32] proposed a sentiment prediction model with an encoded tree structure called R-GAT. R-GAT solved the confusion of opinion words and aspect words in connection to a large extent. To avoid the problem of confusion on connections, Sun et al. [33] performed joint inference between entity types and relation types. Zhang et al. [34] improved the accuracy of relation extraction by pruning the relational dependency tree of GCN with a path-centric approach. Li et al. [35] designed a multi-granularity alignment network named MGAN. MGAN addressed

the problem of inconsistent granularity in cross-domain text analysis. Sun et al. [36] tried to extract the text directly on the dependency tree, and this method is easier to extract the syntactic information of the text. Guo et al. [37] showed an attention-guided graph convolution model abbreviated AGGCN. The GCN model in AGGCN prunes the dependency tree by an attention mechanism to ensure that the dependency tree focuses only on useful information. Zhang et al. [38] constructed an affective classification model called ASGCN. ASGCN uses syntactic rules to guide attentional mechanisms to better focus on the occurrence of important words. Zhao et al. [39] considered that aspect words are not independent and proposed the SDGCN model to verify the feasibility of their idea. In their experiments, they demonstrated that there is a connection between aspect words and that this connection can influence the classification results. Xiao et al. [40] demonstrated an improved classification accuracy by enhancing the interaction information between sentences.

We conclude from reviewing related research works that GCN models have more advantages over ASC. Therefore, a classification model named TS-GCN is proposed, which can perform sentiment classification of consumer reviews from both explicit and implicit aspects, which improves the classification accuracy.

3. Methods

We use an aspect unit $U = (A, O, P)$ to denote the result of this aspect sentiment classification. Where 'A' represents aspect words, 'O' represents opinion words, and 'P' represents sentiment polarity. The TS-GCN model is divided into three parts, which are text feature extraction, explicit aspect sentiment classification and implicit aspect sentiment classification. In this section, we will describe the specific working process of each part and the relationship between the parts. A training example of the model is given at the end of this section.

3.1. The Extraction of Text Features.

The word embedding serves to encode the text, where the encoding contains the content and location of the text. We choose the BERT model as the encoder for the TS-GCN model due to its excellent achievement in the field of NLP. In the text $\{w_1^c, w_2^c, ..., w_T^c, w_{T+1}^c ..., w_{T+M}^c, ..., w_N^c\}$ which contains N words, there are M words $\{w_T^c, w_{T+1}^c, ..., w_{T+M}^c\}$ corresponding to K aspect words $\{w_1^a, w_2^a, ..., w_K^a\}$. We construct the text and aspect words as "[CLS] + text + [SEP] + aspect words + [SEP]" structure, and then input them into the BERT model for training. This input structure allows the text encoding to contain both semantic information and the corresponding aspects of word information. After aggregating the codes, we obtain the text content coding sequence $E_c = \{e_1^c, e_2^c, ..., e_N^c\}$ and the aspect word coding sequence $E_a = \{e_1^a, e_2^a, ..., e_K^a\}$. After completing the encoding, we established a mapping relationship between the text content encoding and the sentiment polarity of the words.

The role of the BiLSTM model is to get the text features of consumer reviews. In the BiLSTM model, the forward and backward layers form a bidirectional transfer mechanism and then connect to the same output layer. This type of transmission allows

the model to obtain the degree of association between words. We input the text content encoding sequence into the BiLSTM model to get the hidden state h_t. Each word corresponds to a hidden state. The hidden states of the text is $H_c = \{h_1^c, h_2^c, ..., h_N^c\}$.

3.2. Explicit Aspect Sentiment Classification

This section is based on the GCN model to accomplish the explicit aspect of sentiment classification. Since the GCN model is good at processing graphs, we make the text features and feature relations form the nodes and edges of the graph for the GCN model to process.

We form the text hidden states $H_c = \{h_1^c, h_2^c, ..., h_N^c\}$ into an *N*N* adjacency matrix A_{ij}, and then input it into the GCN model. A_{ij} is shown in formula (1), where r_{ij} represents the relationship coefficient between each text hidden state h_i^c.

$$A_{ij} = \begin{bmatrix} r_{11} & r_{12} & \text{K} & r_{1N} \\ r_{21} & r_{22} & \text{K} & r_{2N} \\ \text{M} & \text{M} & \text{O} & \text{M} \\ r_{N1} & r_{N2} & \text{K} & r_{NN} \end{bmatrix} \tag{1}$$

The GCN model updates the state of the nodes in each iteration, and the nodes gain information in this process. The update of the node state is calculated as shown in formula (2).

$$x_i^l = \sigma(\sum_{j=1}^{N} A_{ij} W^l x_i^{l-1} + b^l) \tag{2}$$

Where each word corresponds to a node, and *N* is the number of nodes, x_i^l represents the current node state, x_i^{l-1} represents the previous state of the node, σ is the nonlinear function, W^l is the linear transformation weight matrix, and b^l is the offset vector. The GCN model performs a linear transformation of the last node state x_i^{l-1}, then updates the node information using the adjacency matrix A_{ij} as the state matrix, and finally obtains the current node state x_i^l.

Each node is influenced by other nodes during iterative computation until it reaches an equilibrium state. When the iteration parameters reach a steady state, the GCN model aggregates the nodes associated with the same aspect word. We set a threshold for the number of nodes aggregated to determine the existence of an aspect unit. After the training of the GCN model, the initial and equilibrium states of the nodes in the graph are shown in a simplified two-dimensional diagram in Figure 3.

After determining the existence of aspect units, we need to find the corresponding aspect words. Our method is to calculate the similarity between the aspect unit and the aspect word encoding sequence $E_a = \{e_1^a, e_2^a, ..., e_K^a\}$ by probabilistic prediction. The probabilistic prediction is shown in formula (3).

$$y_i = \frac{\exp(x_i)}{\sum_{j=1}^{K} \exp(x_{ij})} \tag{3}$$

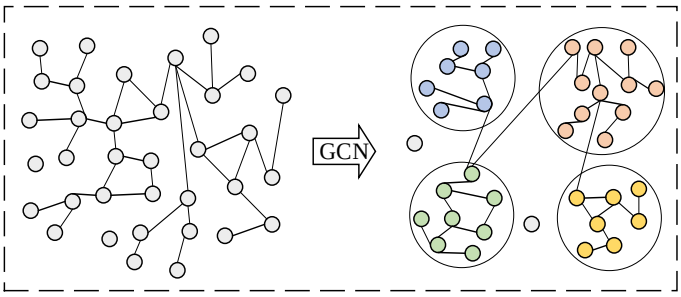


Fig. 3. The nodes before and after the GCN model training. We use different colors to indicate that the nodes are aggregated into different parts after training, and each part corresponds to an aspect word

The working process of the GCN model is shown in the following algorithm:

```
1.      For h in Hc:
2.        Row_stack(A,h)
3.      end for
4.      for each epoch until Δw<threshold:
5.        for each x do
6.          for j in Dimension(x)
7.            temp=LinearMap(x)
8.          end for
9.          x=ReLU(temp)
10.       end for
11.       xl=ReLU(LinearMap(xl))+x
12.     end for
```

In the algorithm above, we complete the explicit aspect of ASC. The adjustment matrix A is constructed in lines 1-3. The working process of multilayer GCN is in lines 4-12. Where line 11 describes the adjustment of the node values. The time complexity of the algorithm is related to three variables, the number of nodes *N*, the number of connections between nodes *N* (including self-connections) and the embedding dimension of each node *d*. The time complexity of the algorithm is $O(N) = O(N^2 d)$.

We combine the matched aspect word a_i, the opinion word o_i in the text and the sentiment polarity p_i of the opinion word to form the aspect unit $u_i = (a_i, o_i, p_i)$. The result of the processing of each sentence in the text is presented in the form of one or more aspect units $U = \{u_1, u_2, ..., u_L\}$, L is the number of aspect units. If there is no match for the corresponding aspect word, the opinion word o_i and the sentiment polarity p_i will be constructed into an implicit aspect unit $u_i = (NULL, o_i, p_i)$. The implicit aspect unit will be processed in the TS model in the next section.

3.3. Implicit Aspect Sentiment Classification

After the training of the GCN model, we have been able to extract the aspect units containing the explicit aspect words. However, we are unable to extract implicit aspect units that do not contain aspect words. For example, for the sentence "The environment

is good, just a little expensive.", the GCN model can only extract explicit aspect units <environment, good, positive> and incomplete implicit aspect units <NULL, expensive, negative>. In this section, the TS model will be used to predict the implicit aspect words and fill in the incomplete implicit aspect unit $u_i = (NULL, o_i, p_i)$, thus completing the sentiment classification of implicit aspects.

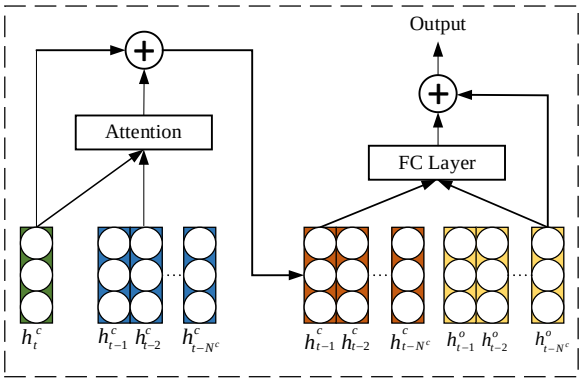


Fig. 4. The TS model

The TS model consists of the truncated history-attention (THA) module and the selective transformation network (STN) module. The structure of the TS model is shown in Figure 4. The input of the THA module comes from the text hidden states $H_c = \{h_1^c, h_2^c, ..., h_N^c\}$ output by the BiLSTM model. The THA module scores each hidden state and obtains the standard score s_i^t after normalization by softmax, as shown in formula (4).

$$s_i^t = \text{Softmax}(v^{\text{T}} \tanh(W_1 h_i^c + W_2 h_t^c + W_3 \tilde{h}_i^c)) \tag{4}$$

Where *t* denotes the processing step at this moment, N^c is the number of hidden states processed by the THA module per unit time, $i \in [t - N^c, t-1]$ is the parameter used for model training. h_i^c is the feature representation of the previous step, h_t^c is the feature representation of the current step, and then h_i^c and h_t^c are used to obtain the historical aware feature representation $\tilde{h}_i^c$ through the THA module. W_1, W_2 and W_3 are the parameters for h_i^c, h_t^c and $\tilde{h}_i^c$.

By calculating the historical aware feature representation $\tilde{h}_i^c$ and the standard score s_i^t, we obtained the historical feature $\hat{h}_t^c$. The calculation process of $\hat{h}_t^c$ is shown in formula (5).

$$\hat{h}_t^c = \sum_{i=t-N^c}^{t-1} s_i^t \times \tilde{h}_i^c \tag{5}$$

We use the nonlinear function ReLU to activate the historical feature $\hat{h}_t^c$. Then, we combine the activated historical feature $\hat{h}_t^c$ with the feature representation of the current step h_t^c to obtain the historical aware feature representation of the current step $\tilde{h}_t^c$. The calculation process of $\tilde{h}_t^c$ is shown in formula (6).

$$\tilde{h}_t^c = h_t^c + \text{ReLU}(\hat{h}_t^c) \tag{6}$$

The STN module builds a fully connected layer using the historical aware feature representation $\hat{h}_t^o$ output by the THA module and the aspect unit feature h_i^o output by the GCN model. And then the STN module uses the set of historical aware feature representations in the fully connected layer as a vocabulary to build an index of candidate aspect words. We obtain the corresponding implicit aspect words by computing the historical aware feature representations of the implicit aspect units. The calculation process of implicit aspect words is shown in formula (7).

$$y_i = \mathrm{ReLU}(W_4 \hat{h}_t^o + W_5 h_i^o) \tag{7}$$

Where W_4 is the parameter of the current historical aware feature representation $\hat{h}_t^o$ and W_5 is the parameter of the aspect unit feature h_i^o.

The loss function is calculated as shown in formula (8).

$$Loss = -\sum_{i=1}^{K}\sum_{j=1}^{c} y_{ij} \log \hat{y}_{ij} + \lambda \|\theta\|^2 \tag{8}$$

The working process of the TS model is shown in the following algorithm:

```
1.      Codes=BERT(Hc)
2.      HStates=LSTM(Codes)
3.      for i=1:Hc.Count:
4.          s= HStates [i]
5.          f=Attention.Feature(s)
6.          Attention.Insert(f)
7.          abstract=FCLayer.Deal(s,f)
8.          FullConn.Insert(abstract)
9.      end for
10.     Initialize out to an empty list
11.     For i=1:U.Count:
12.         u=U[i]
13.         f=Feature(u)
14.         abstract=FCLayer.Deal(f)
15.         a=FCLayer.Deal(FullConn,abstract)
16.         u.A=a
17.         out.Insert(u)
18.     end for
19.     End
```

The algorithm is divided into two parts. The first part is in lines 1-9 of the algorithm. The THA module in the TS model trains the prediction model and calculates the association of all words in the text with aspect words. The second part is in lines 10-19 of the algorithm. The time complexity of the algorithm is related to three variables, the number of hidden states of words *N*, the number of aspect unit *L* and the dimensionality of word embedding *d*. In the TS module, the attention mechanism and the fully connected layer occupy the majority runtime of the model, and the time complexity is $O(N^2d)$ and $O(N^2)$ respectively. The global time complexity is $O(N,L,d)=O(N^3d+L^2d)$.

The STN module in the TS model predicts the implicit aspect words. In the model training step, the TS model takes as input the hidden states of the text processed by the BERT model and the BiLSTM model. The TS model constructs complete connections between words through the positional and semantic relationships between words. Since the source of the corpus for building the model is the review text itself, it is extremely

relevant to product reviews and does not suffer from severe overfitting or poor adaptation of the model to the task. The TS model completes the implicit aspect units by predicting the implicit aspect words to complete the implicit aspect sentiment classification.

3.4. A model training example

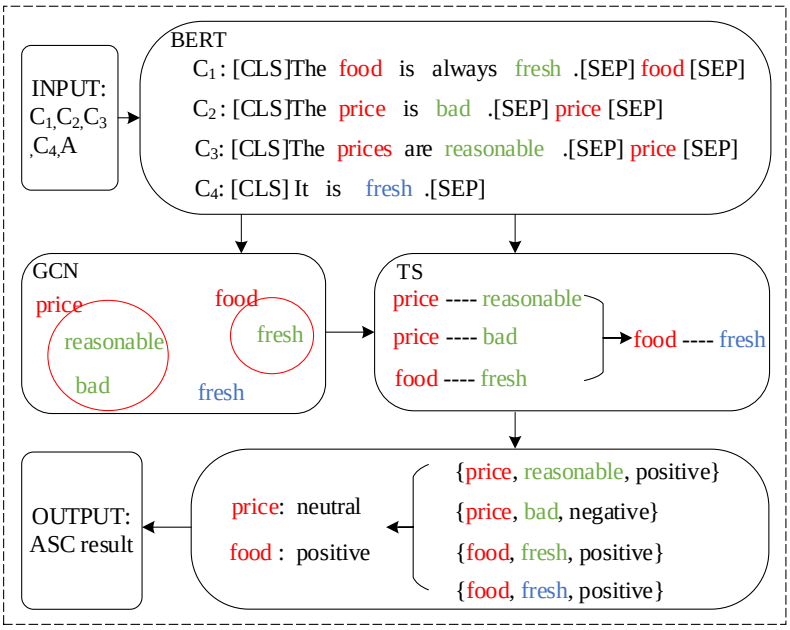


Fig. 5. Example of the training process

We demonstrate the training process of our model TS-GCN with a brief example, as shown in Figure 5. Input sentences C_1, C_2, C_3, C_4 and the marked aspect word A in the sentence. C_1="The food is always fresh.", C_2="The price is bad.", C_3="The prices are reasonable." and C_4="It is fresh.", A="food", "price". We enter the sentences into BERT in the form "[CLS] sentence [SEP] aspect word [SEP]". After BERT encoding, the sentences are input to BiLSTM to mine text features H_c. The GCN aggregates opinion words with the same aspect word, such as "bad" and "reasonable", by adjusting the relationship of H_c. The clustered opinion words are computed to an aspect code. The aspect words, opinion words, and sentiment polarity form the explicit aspect unit $u_i = (a_i, o_i, p_i)$. The GCN module cannot classify the opinion word "fresh" in C_4 because it is not encoded with the corresponding explicit aspect word. The TS module uses text features and aspect units to build a fully connected layer and establish a correspondence system between aspect words and opinion words, such as "food--fresh" and "price--reasonable". Then, the TS module uses the fully connected layer to predict

the implicit aspect words and then form the implicit aspect units. Finally, we count all the aspect units and obtain the results of ASC.

4. Experiments

4.1. Dataset and Experimental Setup

To validate the sentiment classification ability of the TS-GCN model in the consumer review field, we performed experiments on four consumer review datasets. The four review datasets are Laptop 14[1], Restaurant 14[1], Restaurant 15[2], and Restaurant 16[3]. The above experimental dataset contains consumer reviews of laptops and restaurants, and all aspects of sentiment polarity contained in the reviews are annotated. The details of the four datasets are shown in Table 1.

Table 1. Experimental dataset

Data	Laptop 14		Restaurant 14		Restaurant 15		Restaurant 16	
	Train	Text	Train	Text	Train	Text	Train	Text
Positive	992	339	2125	725	910	321	1018	460
Neutral	460	165	626	193	32	32	62	28
Negative	866	123	800	190	244	178	430	110

In this experiment, we use the GloVe model and the BERT model as pre-training models. The experimental setup of the two models is shown in Table 2.

Table 2. Experimental setup for the GloVe and BERT models

Data	GloVe	BERT
Embedding dimension	300	768
Batch-size	32	16
Dropout	0.5	0.1
Learning rate	0.001	0.001

4.2. Comparing models

To validate the performance of the TS-GCN model proposed in this paper, the following seven models were selected for experimental comparison.

RAM[25]: This model uses a weighted attention mechanism to filter irrelevant coded information and captures contextual features more accurately. The model outperformed the ordinary attention model in processing complex text.

AOA[26]: This model processes the known aspect words and sentences to be analyzed through the AOA (attention-over-attention) model. The AOA model makes it easier to focus attention on important parts of the text.

[1] https://alt.qcri.org/semeval2014/task4/index.php?id=important-dates
[2] https://alt.qcri.org/semeval2015/task12/index.php?id=important-dates
[3] https://alt.qcri.org/semeval2016/task5/index.php?id=important-dates

TNet-LF[17]: This model abandons the attention model and uses CNNs and Bidirectional RNNs, which have achieved good performance in classification tasks, to extract aspect and sentiment features and perform classification.

AEN[28]: This model improves on the shortcomings of the RNN model that cannot process the task in parallel by using the attention encoder network (AEN) to accomplish the extraction and classification of aspectual and sentiment polarity in sentences.

ASGCN[38]: The syntactic rules in this model help the attention mechanism to better focus on where important words appear.

AEGCN[40]: This model combines the attention mechanism with GCN to enhance the interaction information between sentences. The model compares the effect of sentiment classification under different word embedding models, which provides a reference for us to choose the pre-training model.

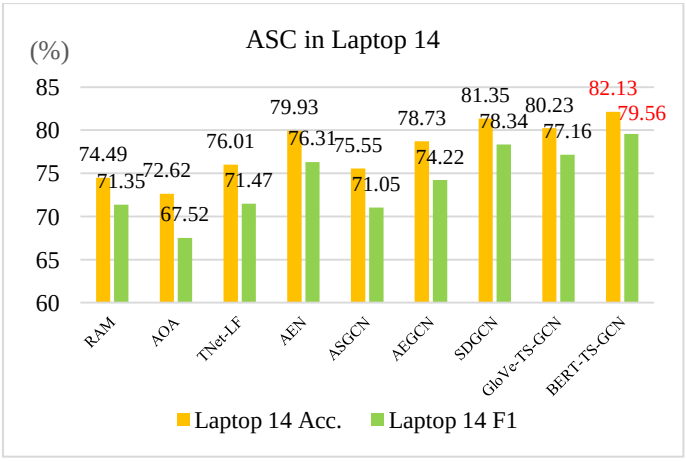
SDGCN[39]: This model proposes that different aspect terms are interrelated. The model captures the relationship between aspect words in sentences through GCN, which improves the accuracy of sentiment classification.

4.3. Experimental results and discussion

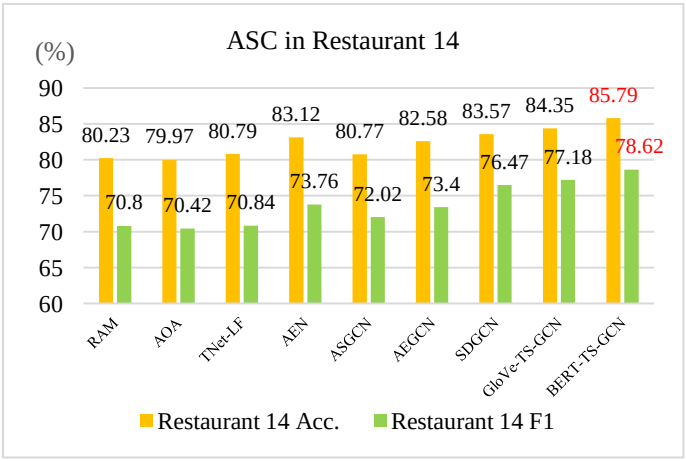
The experimental results of TS-GCN and the seven comparison models are shown in Table 3. We evaluate the model performance using accuracy and F1 as the criteria. Since some comparison models use the GloVe model as a pre-training model, to control the variables, TS-GCN experimented with both the GloVe model and the BERT model as pre-training models. From the experimental results, it can be seen that our proposed TS-GCN model works best among the models that also use the GloVe pre-trained model. And the TS-GCN model under BERT pre-training is better than the TS-GCN model under GloVe pre-training.

Table 3. Comparison results with other ASC models. The experimental results of our proposed model are shown in bold. The experimental results of the comparison model in the table are from the paper that proposed the model. Comparison models that were not experimented on the datasets of Restaurant 15 and Restaurant 16 are indicated by "-"

Model	Laptop14		Restaurant14		Restaurant15		Restaurant16	
	Acc.(%)	F1(%)	Acc.(%)	F1(%)	Acc.(%)	F1(%)	Acc.(%)	F1(%)
RAM	74.49	71.35	80.23	70.80	-	-	-	-
AOA	72.62	67.52	79.97	70.42	78.17	57.02	87.50	66.21
TNet-LF	76.01	71.47	80.79	70.84	78.47	59.47	89.07	70.43
AEN	79.93	76.31	83.12	73.76	-	-	-	-
ASGCN	75.55	71.05	80.77	72.02	79.89	61.89	88.99	67.48
AEGCN	78.73	74.22	82.58	73.40	82.71	69.00	89.61	73.93
SDGCN	81.35	78.34	83.57	76.47	-	-	-	-
GloVe-TS-GCN	80.23	77.16	84.35	77.18	83.95	71.43	90.07	75.14
BERT-TS-GCN	**82.13**	**79.56**	**85.79**	**78.62**	**85.23**	**73.65**	**92.45**	**78.75**



(a)



(b)

Fig. 5. Comparison results of the models on Laptop 14 and Restaurant 14. The experimental results on Laptop 14 are shown in Figure 5(a), and the experimental results on Restaurant 14 are shown in Figure 5(b)

We compared the experimental results of the seven models with those of the TS-GCN under the same criteria. Because the experimental data of dataset Laptop 14 and dataset Restaurant 14 are more complete, the experimental results under these two datasets were chosen to plot the bar chart as shown in Figure 5.

(1) The accuracy and F1 of ASC for models using GCN are mostly higher than those not. We analyzed the possible reasons for such experimental results according to the methods used by the models. Comparing the experimental results of all models, the AOA model performed the worst. AOA combines aspects with sentences in a joint process capturing the interaction between aspects and context. However, the AOA only uses existing information as the full information space at training time and is unable to

analyze new data. The RAM model, on the other hand, makes sentiment features the focus of the model construction, using a multi-attention mechanism to extract multifaceted features. TNet-LF does not use an attention-based approach but treats sentences as bags of words and analyses them using a classification task-based approach. The AEN model was the best result achieved before the addition of classification to the graph neural network.

(2) The TS-GCN under BERT pre-training achieved the best results among all the compared models. Among all the ASC models based on GCN, the ASGCN model introduces syntactic trees, which store the relationships between sentences through syntactic trees. The ASGCN uses a targeted graph convolutional network to process The ASGCN model does not incorporate word location information in the word encoding part, while the AEGCN model compensates for this deficiency. The AEGCN model uses the nodes in the syntactic tree as the individual words of the original utterance, while the SDGCN uses the aspectual words as the nodes of the syntactic tree to better classify the aspectual sentiment through the relationships between the aspectual words in the original text. TS-GCN classifies sentiment from both explicit and implicit aspects, so it improves the accuracy of ASC.

TS-GCN can get the best results in the comparison model because it classifies sentiment from both explicit and implicit aspects. All of the above methods for extracting aspect words are based on the aspect words that appear in the text. However, in consumer reviews, colloquial expressions make it common to omit aspect words from reviews. Therefore, the existing ASC methods are challenged in the consumer review field. TS-GCN constructs a global network of aspectual relations by mining the relationship between aspect words and text words. The model achieves prediction of implicit aspect words through this network, and then further achieves ASC for implicit aspects.

4.4. Case study

To better demonstrate our model, we conducted a case study with several test examples. We visualized the attention scores of words using annotation methods. The words with darker colors have higher scores. We compared the attention scores and prediction results of the four models, as shown in Table 4. These four models are represented by "AEN", "ASGCN", "Ours w/o TS" and "Ours ". "AEN" and "ASGCN" are from two of the comparison models in Section 4.2, "Ours w/o TS" is our model without the TS module, and "Ours" is our model TS-GCN.

The first test example "Great food but the service was dreadful!" is labeled with two explicit aspect words "food" and "service", and two opinion words "good" and "dreadful". The AEN model relies on the attention mechanism to give equal weight to both the word "good" and "dreadful". Therefore, AEN is unable to classify the sentiment of each aspect when multiple aspects are included in the sentence, however, the ASGCN model solves this problem. The ASGCN model correctly identifies multiple pairs of aspect words and opinion words by building dependency trees on the GCN. Our model achieves sentiment classification of multiple explicit aspect words in a sentence by computing aspect words through opinion words. Since this part is implemented in the GCN module, the prediction results of both our model and Ours w/o TS are correct.

Table 4. Case study. The test examples are selected from the dataset used in Section 4.1. When the predicted sentiment polarity is the same as the labeled , the result is ✓, otherwise it is ×

Model	Test	Aspect	Label	Prediction	Result
AEN [28]	Great food but the service was dreadful!	food	Positive	Neutral	×
		service	Negative	Neutral	×
	I found the food to be just as good as its owner, Da Silvano, just much less expensive.	food	Positive	Neutral	×
		price	Negative	-	×
ASGCN [38]	Great food but the service was dreadful!	food	Positive	Positive	✓
		service	Negative	Negative	✓
	I found the food to be just as good as its owner, Da Silvano, just much less expensive.	food	Positive	Neutral	×
		price	Negative	-	×
Ours w/o TS	Great food but the service was dreadful!	food	Positive	Positive	✓
		service	Negative	Negative	✓
	I found the food to be just as good as its owner, Da Silvano, just much less expensive.	food	Positive	Positive	✓
		price	Negative	-	×
Ours	Great food but the service was dreadful!	food	Positive	Positive	✓
		service	Negative	Negative	✓
	I found the food to be just as good as its owner, Da Silvano, just much less expensive.	food	Positive	Positive	✓
		price	Negative	Negative	✓

The second test example "I found the food to be just as good as its owner, Da Silvano, just much less expensive." is marked with an explicit aspect word "food", an implicit aspect word "price", and two opinion words of opposite emotional polarity "good", "expensive". The inability of AEN and ASGCN to recognize the implicit aspect words leads to matching the opinion words of the implicit aspect words to the explicit aspect words. In the example, attention is assigned to the word "good" and "expensive" with the same weight, resulting in the incorrect prediction of the explicit aspect word "food". While Ours w/o TS does not recognize implicit aspect words, it does not match opinion words of implicit aspect words to explicit aspect words incorrectly. Therefore, the sentiment prediction of the explicit aspect words in the sentence is correct. AEN, ASGCN and Ours w/o TS cannot recognize the implicit aspectual word "price" resulting in no prediction results. Our model TS-GCN can predict the implicit aspect words and correctly predicts the sentiment polarity of the aspect word "price".

After the comparison of several models mentioned above, our model has the highest accuracy of ASC in different categories of sentences.

4.5. Ablation study

We performed ablation experiments on TS-GCN to verify the important contribution of each module to the model performance. We have verified the positive effect of the BERT pre-training model. In the ablation experiments in this section, we compared the two models. One is to abandon the TS model directly; The other is to replace the GCN model with the LSTM-CRF model. The results of the ablation experiments are shown in Table 5.

From the results of the ablation experiments, it can be seen that each module plays a crucial role in TS-GCN.

Table 5. Results of ablation experiments. "Ours w/o TS" means our model without TS module, "Ours w/o GCN" means our model without GCN module, and "Ours" means our model TS-GCN

Model	Laptop 14		Restaurant 14		Restaurant 15		Restaurant 16	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Ours w/o TS	73.74	70.03	66.35	70.14	77.59	65.92	86.18	71.24
Ours w/o GCN	79.52	77.27	82.18	74.58	82.67	69.49	88.53	75.05
Ours	82.13	79.56	85.79	78.62	85.23	73.65	92.45	78.75

4.6. GCN layers study

For purpose of verifying the effect of the number of layers of the GCN on the classification effect, we classified four categories of text with the different numbers of layers of the GCN. We performed the classification operation on the text with the GCN of layers 1 to 8. We simplified the GCN classification results into a 2D diagram by PCA. On the 2D diagram, the four categories of text data are represented in yellow, red, green and blue. The simplified diagrams of the classification results for different layers of GCN are shown in Figure 6.

We can see that the 2-layer GCN has the best classification result, so the final TS-GCN model uses a 2-layer GCN.

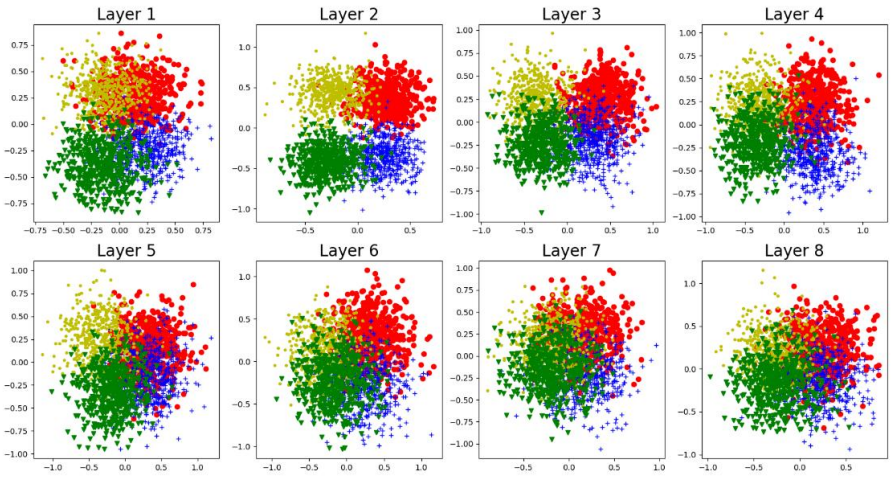


Figure 6. Simplified diagram of the classification results of the GCN with different layers. The coordinates of the data points are constrained to be between -1 and 1

5. Conclusions

To improve the accuracy of ASC in the field of consumer reviews, a model that can accomplish ASC in both explicit and implicit aspects is proposed, which is called TS-GCN. TS-GCN has the following two achievements.

(1) TS-GCN improves the accuracy of explicit aspect sentiment classification by training a GCN model. We use the BERT model and the BiLSTM model to extract text features and then input them into the GCN model for training. The GCN model obtains explicit aspects of ASC by processing the text features.

(2) TS-GCN achieves the implicit aspect of sentiment classification by introducing and training the TS model. The TS model predicts implicit aspect words by establishing correspondences between explicit aspect words and other words. The successful prediction of implicit aspect words compensates for the inability of the GCN model to classify sentiment for implicit aspects.

Through experiments on four datasets, we demonstrate that adding sentiment classification for implicit aspects can improve the accuracy of ASC. In the future, we will further study the ASC combined with multimodal features such as image and audio.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant NO.62076006), the University Synergy Innovation Program of Anhui Province (GXXT-2021-008), and the National Natural Science Foundation of Anhui Province (1908085MF189).

References

1. Habimana, O., Li, Y., Li, R., Gu, X., Yu, G.: Sentiment analysis using deep learning approaches: an overview. Science China Information Sciences, 63(1), 1-36 (2020)
2. Peng, H., Xu, L., Bing, L., Huang, F., Lu, W., Si, L.: Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Vol. 34, No. 05, pp. 8600-8607 (2020, April)
3. Zhou, J., Huang, J. X., Chen, Q., Hu, Q. V., Wang, T., He, L.: Deep learning for aspect-level sentiment classification: survey, vision, and challenges. Institute of Electrical and Electronics Engineers, 7, 78454-78483 (2019)
4. Hu, M., Liu, B.: Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168-177 (2004, August)
5. Nigam, K., Hurst, M.: Towards a robust metric of polarity. In Computing attitude and affect in text: Theory and applications, pp. 265-279 (2006)
6. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 79–86 (2002)
7. Wang, S. I., Manning, C. D.: Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers, pp. 90-94 (2012, July)
8. He, R., Lee, W. S., Ng, H. T., Dahlmeier, D.: An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 504–515, Florence, Italy. Association for Computational Linguistics (2019)
9. Akhtar, M. S., Garg, T., & Ekbal, A.: Multi-task learning for aspect term extraction and aspect sentiment classification. Neurocomputing, 398, 247-256 (2020)

10. Cruz, I., Gelbukh, A. F., Sidorov, G.: Implicit Aspect Indicator Extraction for Aspect based Opinion Mining. International Journal of Computational Linguistics and Applications, 5(2), 135-152 (2014)
11. Poria, S., Cambria, E., Ku, L. W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In Proceedings of the second workshop on natural language processing for social media, pp. 28-37 (2014, August)
12. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 4171–4186. Minneapolis, Minnesota (2019)
13. Li, X., Bing, L., Li, P., Lam, W., Yang, Z.: Aspect Term Extraction with History Attention and Selective Transformation. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Main track. pp. 4194-4200 (2018)
14. Liu, Y., Li, P., Hu, X.: Combining context-relevant features with multi-stage attention network for short text classification. Computer Speech & Language, 71, 101268. (2022)
15. Chen, S., Liu, J., Wang, Y., Zhang, W., Chi, Z.: Synchronous double-channel recurrent network for aspect-opinion pair extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6515-6524. Online (2020, July)
16. Xu, H., Zhang, S., Zhu, G., Zhu, H.: ALSEE: a framework for attribute-level sentiment element extraction towards product reviews. Connection Science, pp. 1-19 (2021)
17. Li, X., Bing, L., Lam, W., Shi, B.: Transformation networks for target-oriented sentiment classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Vol. 1, pp. 946–956. Melbourne, Australia (2018)
18. Zhu, H., Lin, Y., Liu, Z., Fu, J., Chua, T. S., Sun, M.: Graph neural networks with generated parameters for relation extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1331–1339. Florence, Italy (2019)
19. Chiang, W. L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C. J.: Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In Proceedings of the 25th Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining, pp. 257-266. New York, USA (2019, July)
20. Zhang, S., Yu, H., Zhu, G.: An emotional classification method of Chinese short comment text based on ELECTRA. Connection Science, 34(1), 254-273 (2022)
21. Aloysius, N., Geetha, M.: A scale space model of weighted average CNN ensemble for ASL fingerspelling recognition. International Journal of Computational Science and Engineering, 22(1), 154-161 (2020)
22. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Sun, M.: Graph neural networks: A review of methods and applications. AI Open, 1, 57-81 (2020)
23. Fang, W., Jiang, T., Jiang, K., Zhang, F., Ding, Y., Sheng, J.: A method of automatic text summarisation based on long short-term memory. International Journal of Computational Science and Engineering, 22(1), 39-49 (2020)
24. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 606-615. Austin, Texas (2016, November)
25. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of the 2017 conference on empirical methods in natural language processing, pp. 452-461. Copenhagen, Denmark (2017, September)
26. Huang, B., Ou, Y., Carley, K. M.: Aspect level sentiment classification with attention-over-attention neural networks. In International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, pp. 197-206 (2018, July)
27. Xue, W., Li, T.: Aspect based sentiment analysis with gated convolutional networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Vol. 1, pp. 2514–2523. Melbourne, Australia (2018)

28. Song, Y., Wang, J., Jiang, T., Liu, Z., Rao, Y.: Attentional encoder network for targeted sentiment classification. arXiv preprint arXiv:1902.09314 (2019)
29. Fan, Z., Wu, Z., Dai, X., Huang, S., Chen, J.: Target-oriented opinion words extraction with target-fused neural sequence labeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 2509-2518. Minneapolis, Minnesota (2019, June)
30. Huang, B., Carley, K. M.: Parameterized convolutional neural networks for aspect level sentiment classification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1091–1096. Brussels, Belgium (2018)
31. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. Computational Social Networks, 6(1), 1-23 (2019)
32. Wang, K., Shen, W., Yang, Y., Quan, X., Wang, R.: Relational graph attention network for aspect-based sentiment analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3229–3238. Online (2020)
33. Sun, C., Gong, Y., Wu, Y., Gong, M., Jiang, D., Lan, M., Duan, N.: Joint type inference on entities and relations via graph convolutional networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1361-1370. Florence, Italy (2019, July)
34. Zhang, Y., Qi, P., Manning, C. D.: Graph convolution over pruned dependency trees improves relation extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2205–2215. Brussels, Belgium (2018)
35. Li, Z., Wei, Y., Zhang, Y., Zhang, X., Li, X.: Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Vol. 33, No. 01, pp. 4253-4260 (2019, July)
36. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Aspect-level sentiment analysis via convolution over dependency tree. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 5679-5688. Hong Kong, China (2019, November)
37. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 241–251. Florence, Italy (2019)
38. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 4568–4578. Hong Kong, China (2019)
39. Zhao, P., Hou, L., Wu, O.: Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. Knowledge-Based Systems, 193, 105443 (2020)
40. Xiao, L., Hu, X., Chen, Y., Xue, Y., Gu, D., Chen, B., Zhang, T.: Targeted sentiment classification based on attentional encoding and graph convolutional networks. Applied Sciences, 10(3), 957. (2020)

Shunxiang Zhang, born in 1970. PhD, professor, PhD supervisor. He is an professor at Anhui University of Science and Technology, China. His current research interests include Web Mining, Semantic Search, and Complex network.

Tong Zhao, born in 1997. Master candidate at Anhui University of Science and Technology. Her main research interests are Named Entity Recognition and Relation Extraction.

Houyue Wu, born in 1996. Master candidate at Anhui University of Science and Technology. His main research interests are Adversarial Sample Generation and Relation Extraction.

Guangli Zhu, born in 1969. Master, Associate professor, Master supervisor. Her current research interests include Web Mining, Semantic Search, and Calculation theory.

KuanChing Li, born in 1967, professor, PhD supervisor. He is a University Distinguished Professor at Providence University, Taiwan. His current research interests include Paraallel and Distributed Computing, Big Data and Emerging Technologies.

Received: March 25, 2022; Accepted: October 01, 2022.

Computer Science and Information Systems 20(1):137–156 https://doi.org/10.2298/CSIS220921056L

Secure Cloud Internet of Vehicles Based on Blockchain and Data Transmission Scheme of Map/Reduce

Hua-Yi Lin

Department of Information Management, China University of Technology, Taiwan, ROC
calvan.linmsa@gmail.com

Abstract. Over the past few years, because of the popularity of the Internet of vehicles and cloud computing, the exchange of group information between vehicles is no longer out of reach. Through WiFi/5G wireless communication protocol, vehicles can instantly deliver traffic conditions and accidents to the back end or group vehicles traveling together, which can reduce traffic congestion and accidents. In addition, vehicles transmit real-time road conditions to the cloud vehicle management center, which can also share real-time road conditions and improve the road efficiency for pedestrians and drivers. However, the transmission of information in an open environment raises the issue of personal information security. Most of the security mechanisms provided by the existing Internet of vehicles require centralized authentication servers, which increase the burden of certificate management and computing. Moreover, the road side unit as a decentralized authentication center may be open to hacking or modification, but due to personal privacy and security concerns, vehicle-to-vehicle is not willing to share information with each other. Therefore, this study is conducted through blockchain to ensure the security of vehicle-based information transmission. Moreover, the elliptic curve Diffie–Hellman (ECDH) key exchange protocol and a secure conference key mechanism with direct user confirmation combined with the back-end cloud platform Map/Reduce is proposed to ensure the identities of Mappers and Reducers that participate in the cloud operation, avoid malicious participants to modify the transmission information, so as to achieve secure Map/Reduce operations, and improves vehicle and passenger traffic safety.

Keywords: Internet of vehicles, blockchain, ECDH, Map, Reduce.

1. Introduction

Recently, with the gradual popularization of 5G and electric vehicles, the cloud Internet of vehicles (CIoVs) has become more feasible. CIoVs are able to connect the vehicle and surrounding devices to share information with each other, and transfer a considerable quantity of obtained data individually to the back-end platform of cloud computing for a huge amount of data calculation and analysis, and then obtain valuable information.

The framework and components of the cloud Internet of vehicles, as shown in Fig. 1, include V2V vehicle-to-vehicle communication, V2P vehicle-to-pedestrian, V2R vehicle-to-roadside equipment, V2G vehicle-to-group communication, V2N vehicle-to-

network, V2I vehicle-to-infrastructure and V2X vehicle-to-everything [1][2]. Additionally, the Map/Reduce operation includes a master cloud operation server named master, and several mapper servers, which are cloud mapping operation servers, and reducer servers for cloud aggregation operation servers.

When the vehicle is moving, it can communicate and share information through V2X, and transmit the information to the Internet through the base station or the roadside equipment RSU, and then the base station or RSU forwards the message to the cloud service classifier (CSC) through routers. Subsequently, CSC dispatches the received message to the corresponding platform of the cloud service to accomplish the Map/Reduce computation depending on the service category requested.

In the open wireless network, the on board unit (OBU) of the vehicle has a variety of interfaces to accomplish communication with the cloud service, RSUs and inside devices of the vehicle. Along with the navigation function, the vehicle also provides many road information for the unmanned vehicle and receives instructions from the cloud control center. Therefore, the vehicle terminal OBU is also an important target for hackers to attack or tamper with transmission data. In addition, RSU is an important core node of cooperative vehicle-road operations. RSU can connect the basic traffic equipment such as signal lights and the cloud control platform via wired interfaces. However, through these interfaces, intruders can launch attacks on roadside devices, affecting traffic safety [3][4].

In addition, we cannot certify the reliability of the identity of the cloud service platform selected from the cloud to participate in the implementation of the Map/Reduce operation, which highlights the lack of an integrated and effective data security protection architecture for the cloud Internet of vehicles.

Looking at the current stage, most of the solutions focus on the security of the Internet of vehicles, but do not incorporate the back-end cloud computing services. Many proposed solutions only focus on how to achieve the secure information transmission on the Internet of vehicles. For example, Insaf et al. [5] proposed a certificateless signcryption scheme for IoV. Pandi et al. [6] proposed batch authentication and key exchange protocols for VANETs. Since the plaintext message is available to unauthorized use and even vicious manipulation. Therefore, Jianfeng et al [7]. developed a secure message sharing mechanism based on an attribute encryption technique using blockchain, which is structured by RSUs. Or, most studies only focus on the security research of cloud service platforms. For example, Tian et al. [8] proposed the cloud enabled robust authenticated key agreement scheme and Yuting et al. [9] proposed a cloud data security sharing scheme using blockchain.

To sum up, the current research topic of Internet of vehicles information security only focuses on the single Internet of vehicles and does not combine the architecture and mechanism of information security transmission from the vehicle terminal to the cloud platform. In addition, most of the information security transmission protocols of the Internet of vehicles currently rely on the centralized certificate server CA or the trust authentication server TA. Once TA is damaged or computing resources are insufficient, it may not be capable of providing effective information security services for the Internet of vehicles.

However, the blockchain technology can achieve trust decentralization through consensus algorithm, which can avoid relying on a single CA or TA server, and provide the identity authentication and the trust mechanism for the Internet of vehicles. Behind

the digital identity management, blockchain can bring a trusted and unique identity identification to the Internet of vehicles system.

Similarly, the life-cycle information of the device is stored on a distributed ledger, just like it is calibrated for cells in the human body. The key information of the certificate application, certificate issuance, signature check, certificate revocation and other processes can be recorded on the chain to achieve controllable traceability of vehicle production, vehicle registration, property right management, owner identity authentication, IoV equipment authentication and other links.

Based on the trusted identity authentication and security mechanism, the Internet of vehicles system can merge accumulated information by multiple OBU devices to update the traffic environment in real time and provide more precise and real-time information for autonomous vehicles. And drivers don't have to worry about privacy, because blockchain can preserve the privacy of traffic participants by merging with privacy enhancement protocols that provide anonymity and untraceability when sending data.

Accordingly, this study primarily intends to propose a decentralized information security transfer protocol and incorporate the Internet of vehicles to the platform of the cloud service to achieve the secure information delivery on Map/Reduce operations, thus providing a research direction on the cloud Internet of vehicles of information security.

The remainder of this document is organized as follows. Section 2 introduces blockchain and its related research. In Section 3, this study details our proposed blockchain based secure Map/Reduce information transmission. Section 4 describes the secure data transmission analyses and evaluations. Then, in Section 5, we will discuss the status of this study, outline future research options, and conclude this study.

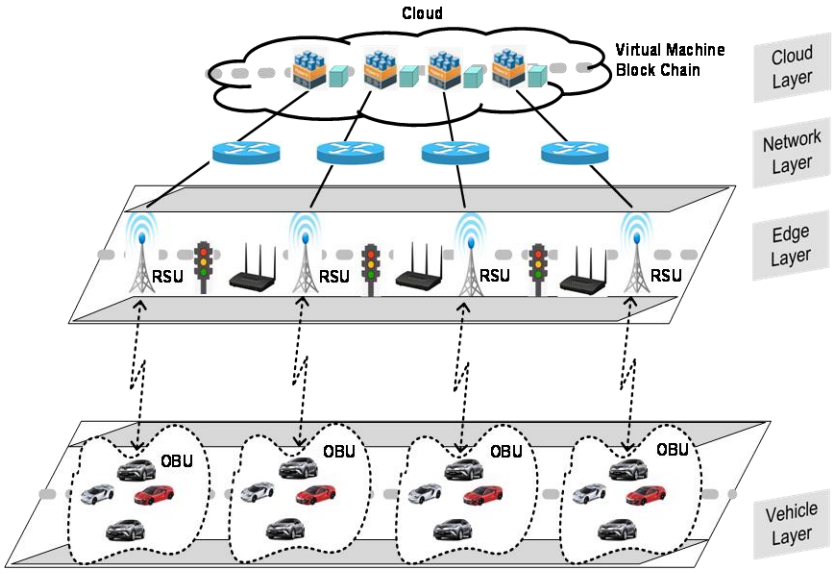


Fig. 1. Framework of cloud Internet of vehicles

2. Related work

Over the past few years, blockchain has become the mainstream of cryptocurrency. A blockchain is a point-to-point decentralized database. Compared with conventional databases, data is stored in one place, and blockchain distributes these data in numerous minor places, which are called nodes.

Blockchain has the following properties: 1. Decentralization 2. Anonymity 3. Tamper-resistance 4. Data consistency 5. Transparency of information [10][11]. In addition, a blockchain is composed of numerous blocks, which are then tied together to form a blockchain. The data of each block contains two types of data, and there are block header and block body. Fig. 2 depicts the detailed block structure, and Fig. 3 represents our proposed blockchain infrastructure of cloud Internet of vehicles.

Moreover, the block header possesses the following types of data [12][13].

(1). Previous block hash: This hash value is computed by the header of the previous block.

(2). Time stamp: The time stamp for generating the current block.

(3). Nonce: The number of the workload algorithm and the difficulty target of the workload algorithm.

(4). Merkle tree root hash: This is the hash value of the body of the current block. It is also the hash value of the root node of the Merkle tree, which is calculated through the Merkle tree algorithm.

As we know, the Merkle tree is a tree-like formation. Each nonleaf node is marked with a hash value. We use this tree-like structure to obtain the hash value of a string of

data, and the time stamp represents what happened at that time on the blockchain. And make sure that each block is linked sequentially. In this case, the Merkle tree root is the root node mentioned above, and the lower node is all transaction records. The block body is equivalent to transactions [14], which includes the following part and trait [15].

(1). Transaction: It is the information of generating the block body, which includes the generation time, the number of accepted transactions, the hash value of the Merkle tree node of the transaction, the address recognized by the transaction, the transaction's digital signature, the index number of the transaction record (used to query the transaction address), data and the record size, etc. Each transaction record has a hash value of a Merkle tree node, which confirms that the transaction cannot be duplicated or forged.

(2). Genesis block. The genesis block is the first block, and its former block hash would be null. When the system generates a blockchain, it creates a genesis block first. Other blocks use the hash of the former block in the header to remember the hash value of the previous block and achieve a complete chain.

(3). Blockchain cannot be modified. Since a change in the transaction record means that the Merkle tree root value of the body in header will be changed, which causes the hash of the entire block header need to change, because the integrity of the chain is cracked. Consequently, the next block's previous block hash must likewise be adjusted, so if someone would like to modify a block's transaction record, they would have to change all subsequent blocks, which is practically impossible.

To sum up, as blockchain refining is rapid and irreversible, once the data is modified in the transmission process, it can be instantly detected, which is very suitable for the distributed cloud Internet of vehicles environment. Therefore, this study would like to employ blockchain to protect the CIoVs security and achieve secure Map/Reduce operation on the cloud platform.

In recent years, many research topics on the Internet of vehicles have been proposed. Azees et al. [16] submitted an anonymous authentication mechanism for a vehicular ad hoc network with security during handovers between RSUs, and consumed less computing power and cost.

Jing et al. [17] provided a lightweight authentication based on blockchain and a key agreement scheme for IoVs that improves the authentication efficiency through a multi-TA model. Authors used blockchain to reserve the authentication information and cross-domain authentication of vehicles, and consequently to defend the privacy information of users. Meanwhile, the recommended scheme employed a lightweight computing operation to lower the authentication time of the vehicle and hence accomplish the entire authentication process.

Bagga et al. [18] designed a batch authentication scheme for IoVs based on blockchain. There are two types of authentications: (1) The authentication of vehicle to vehicle: In a cluster, this mode enables the vehicle to authenticate neighboring vehicles. (2) Batch authentication makes it possible for a group of vehicles to be authenticated by their RSUs. Eventually, the vehicles and RSUs in the cluster can cooperatively create a group key. RSUs then collect secure vehicle data and forms multiple transactions, including the vehicle information and the personal vehicle information for group members.

Cui et al. [19] offered a consortium blockchain based on secure and effective data sharing among vehicles [20]. Within conventional vehicle systems, data sharing should

take place with road side units. In this study, the authors leverage a consortium of decentralized technology to reach trackable data sharing between anonymous vehicles and actually accomplish used data sharing. Additionally, combining 5G and blockchain allows data sharing without the use of RSUs. Through delegation, the authors provided an improved challenge validation consensus algorithm in order to make it more appropriate for distributed IoVs. Eventually, an exhaustive analysis demonstrates that the proposed mechanism is effective and secure.

Li et al. [21] designed a blockchain based distribution scheme for mutual healing group keys in a dedicated network of unmanned aerial vehicles. Primarily, the ground control station (GCS) has built a private blockchain where group keys delivered by GCS are stored. Concurrently, the membership certification of a dynamic list of unmanned aerial vehicles Ad-Hoc network is likewise handled using blockchain. Under various attack patterns, a basic mutual healing scheme and an improved protocol were provided from the mechanism of the longest lost chain to retrieve the lost group keys from the node with the assistance of its neighbors.

Moreover, Lu et al. [22] offered a privacy maintaining authentication scheme based on blockchain for vehicle Ad-Hoc networks. Ma et al. [23] proposed a secure announcement sharing based on attributes among vehicles through blockchain. Authors developed a blockchain based on privacy maintaining authentication named BPPA mechanism for vehicular Ad-Hoc networks. In BPPA, this research used the blockchain to continuously and unalterably store all the certificates and transactions to achieve the transparent and verifiable activities of the semi-TAs. Furthermore, this research extended the traditional blockchain architecture to offer a distributed authentication mechanism without the revocation list. In order to reach circumstantial privacy, this research authorized a vehicle to deploy various certificates. A linkage between the certificate and the real identity is ciphered and reserved in a blockchain. In the event of disagreement, the linkage can only be disclosed.

In general, all of the above studies focus on vehicle NET. However, the information on the Internet of vehicles will eventually be transferred to the cloud for massive data processing to obtain value-added information. Therefore, the above studies lack a discussion of the combination of the Internet of vehicles and the cloud computing of the back end. Consequently, this study will propose an information security transmission architecture utilizing a blockchain mechanism combined the vehicle terminal with cloud.

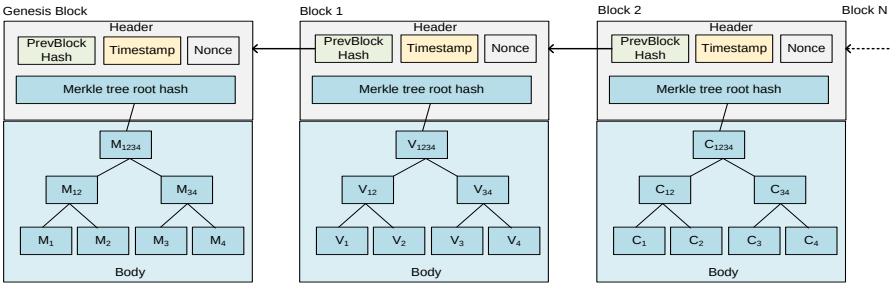


Fig. 2. The detailed block structure

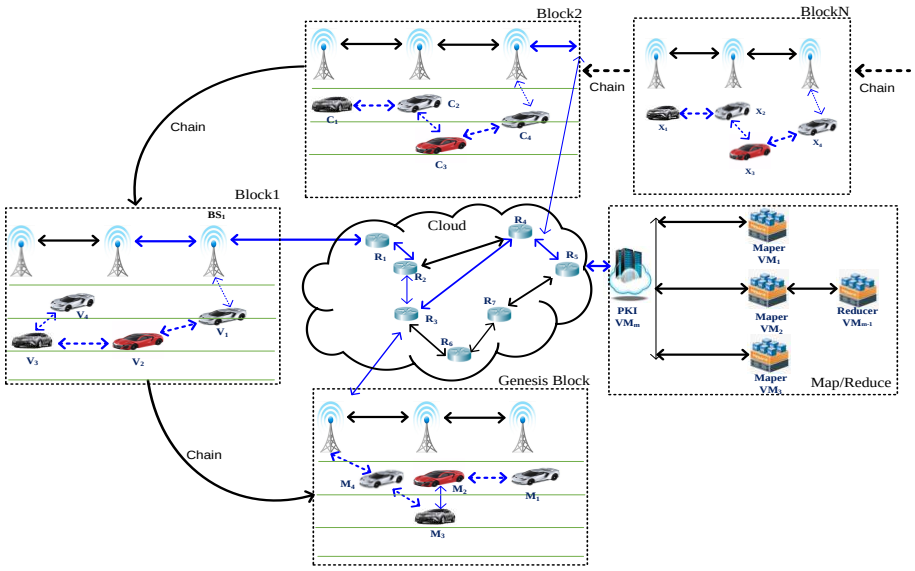


Fig. 3. The blockchain infrastructure of cloud Internet of vehicles

3. Blockchain based secure Map/Reduce data transmission

Based on the above argument, this study proposes a blockchain based secure Map/Reduce data transmission scheme for CVoTs, which can assure that data will not be tampered during the process of data transmission. Moreover, when moving vehicles would like to transmit gathered data by OBU to each other and the back-end cloud computing service platform through vehicles. It also ensures the information security of cloud computing.

For data security, we adopt multisignature to ensure the authenticity of decentralized transactions without trust centers for blockchain. This study assumes that vehicle V_1, V_2, V_3,..., V_i are within the wireless transmission range of the same RSU. Firstly, the vehicles involved in the operation collectively determine a large prime number P at least greater than 512 bits, a primitive element α in FG(P) and a hash function f. Subsequently, each vehicle V_i selects a private key $k_i \in [1, p\text{-}1]$, calculates and announces the corresponding public key $z_i = \alpha^{ki} \bmod p$. In addition, the public key $z = \prod_{i=1}^{n} z_i$ of this group is the multiplication of the public key of all the vehicle members. The process of digital multisignature for the secure data transmission via the group vehicle is as follows:

Step 1. Individual digital signature: Each vehicle member V_i selects an integer $r_i \in [1, p\text{-}1]$ and computes $w_i = \alpha^{ri} \bmod p$ to be the value of the commitment, and subsequently broadcasts w_i to all other vehicle members. When all members have broadcast w_i, i =1, 2, 3..., n. Then each member of the group calculates the following equation for himself.

$$W = \prod_{i=1}^{n} w_i \bmod p. \quad (1)$$

Each member of the group utilizes the private keys k_i and r_i to generate an individual signature for the plaintext data D to be transmitted, and obtains the following individual digital signature.

$$S_i = k_i D' - r_i w \bmod p\text{-}1. \quad (2)$$

$\{w_i, S_i\}$ is the result of the individual signature and satisfies $D' = F(D)$. Subsequently, $\{w_i, S_i\}$ is securely transmitted to the RSU, which verifies the signatures of each vehicle and combines the individual signature into a multisignature. The role of the RSU is to serve the vehicle and it does not possess any key itself. The RSU then adopts the following equation to verify the individual digital signature $\{w_i, S_i\}$ according to the public key z_i of the vehicle member V_i.

$$z_i^{D'} = w_i^{w} \alpha^{Si} \bmod p. \quad (3)$$

Step2. Multisignature: After the RSU has obtained and verified all individual digital signatures, it can transfer the individual signatures w_i and S_i where i =1, 2, 3..., n, and then merge them into a multisignature $\{w, S\}$, which also satisfies

$$S = S_1 + S_2 + \ldots + S_n \bmod p\text{-}1. \quad (4)$$

Step 3. The verification of multisignature: Any member can verify the signature of the plaintext D according to the only announced public key z. The verification equation is as follows.

$$z_i^{D'} = w_i^{w} \alpha^{Si} \bmod p. \quad (5)$$

Lemma: If $z_i^{D'} = w_i^{w} \alpha^{S} \bmod p$, then the multisignature $\{w, S\}$ can be verified and accepted.

Proof: As we know, each vehicle member's signature $\{w_i, S_i\}$ satisfies the following equation.

$$z_i^{D'} = w_i^{w} \alpha^{Si} \bmod p. \quad (6)$$

If the above equation is multiplied for n times, where i=1, 2, 3…, n. This study can infer that the multisignature $\{w, S\}$ is correct as described below.

$$\prod_{i=1}^{n} z_i^{D'} = \prod_{i=1}^{n} w_i^{w} \alpha^{Si} \bmod p.$$

$$\prod_{i=1}^{n} (z_i)^{D'} = (\prod_{i=1}^{n} w_i)^{w} \alpha^{S1+S2+\ldots+Si} \bmod p. \quad (7)$$

$$z^{D'} = w^{w} \alpha^{S} \bmod p.$$

Here, this study assumes when each vehicle on the move sends data D_1~D_i, through the routing path $V_i \leftrightarrow V_3 \leftrightarrow V_2 \leftrightarrow V_1$, as shown in Fig. 4. This study adopts the blockchain algorithm to secure the transfer of data. We assume that four vehicles i = 4 and the number of transmitted data blocks are four. Table 1 describes the usage of notations in secure information transfer.

Step 1. This study employs the multisignature scheme to confirm the transaction record. First of all, each V_1~V_4 performs the multisignature on all transmitted data D_1~D_4. Subsequently, this study adopts SHA1 to compute the hash value of each vehicle's multisignature block to obtain $H(S_{Di})$, and concatenate the multisignature result as Node 0_i=$[H(S_{Di})||(S_{Di})]$, i=1~4. For example, Node 0_1=$[H(S_{D1})||(S_{D1})]$, Node 0_2=$[H(S_{D2})||(S_{D2})]$, Node 0_3=$[H(S_{D3})||(S_{D3})]$, Node0_4=$[H(S_{D4})||S_{D4}]$

Step 2. Subsequently, this study combines the two adjacent nodes and performs the hash algorithm to get their parent $Node_{1[(i+1)/2]} = [H(H(S_{Di})|H(S_{Di+1}))||S_{Di}|S_{Di+1}]$, i = 1, 3, 5, 7…

Step 3. Repeat the operation of Step 2 until obtain the root node named Markle root, as shown in Fig. 5.

When a block is corrupted or modified, the Merkle tree root value can be obtained by recalculating from the corrupted node through to the Merkle tree root node path. In addition, we can also determine where the corrupted node is, according to the following steps.

Step1. Take S_{D1}, S_{D2}, S_{D3}, S_{D4} as input and compute the newer hash value H^* of the root node $Node2_1$ and verify whether the original $H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))$ is equal to the H^* result. If they are different, check their children node $Node1_1$ and $Node1_2$.

Step2. Perform the similar hash operations, if $Node1_1$ is the same and $Node1_2$ is different, then this study checks $Node1_2$'s child $Node0_3$ and $Node0_4$.

Step3. Perform the similar hash operations, if $Node0_3$ is the same and $Node0_4$ is different, then this study checks Node04 and eventually finds out the exact corrupted node.

Table 1. The usage of notations in secure information transfer

Symbol	Description
V_i	Identification of a vehicles
D_x	Delivered data of V_x
S_{Di}	Multisignature of delivered data D_i
R_x	Identification of a router
$H^*(S_{Dx})$	SHA1 hash operation of the multisignature data S_{Dx}, * represents the newer hash operation
$H(S_{Dx})$	SHA1 hash operation of the multisignature data S_{Dx}
VM_i	Identification of a virtual machine
TS	Time stamp
ID_{VMx}	Identity of a virtual machine VM_x
EK_K	Encipher data utilizing the key of K
‖	Concatenation operator

During operation, the proposed method only consumes the $O_{\log}N$ time complexity of comparison, where N is the amount of data blocks. Additionally, the time complexity of generating this Merkle tree is $O(n)$ for the number of hash computations.

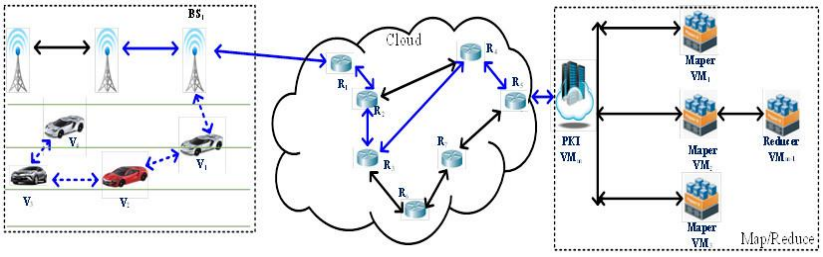


Fig. 4. The partial enlarged drawing of the secure data transmission of CIoVs

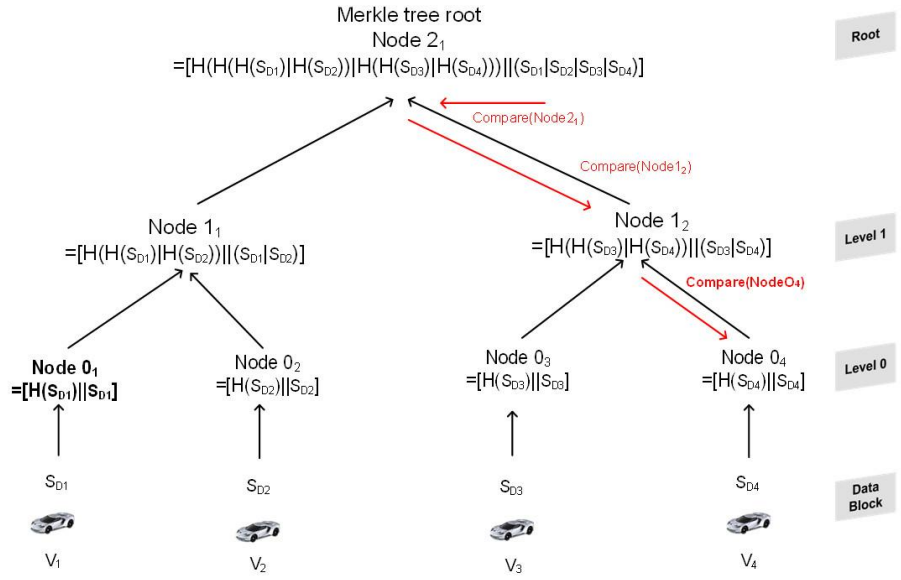


Fig. 5. The operation of the Merkle tree root

3.1. The Method of the Secure Data Transmission

When the message is transmitted from the vehicle V_1 to the cloud via base station BS_1, here we assume that BS_1 and all routers have passed the security authentication before being deployed. Additionally, the delivered message passes through the following routing path $BS_1 \rightarrow R_1 \rightarrow R_2 \rightarrow R_3 \rightarrow R_4 \rightarrow$ PKI VM_m, as shown in Fig. 4.

Initially, when the vehicle V_1 transmits data to the cloud platform through the base station BS_1, this study adopts the ECDH protocol to secure the transfer of data. ECDH is similar to the conventional Diffie-Hellman key protocol. Both parties can establish the session key on the insecure channel. Subsequently, the two parties encrypt and decrypt data through the session key. The key length must be 1024 bits to provide a higher level of security. Although ECDH uses the Diffie-Hellman key protocol to implement elliptic curve cryptosystems, ECDH only requires 160 bits of key length and consumes less computing resources to achieve the same security strength [24][25][26]. Therefore, ECDH is very suitable for the network of vehicles that lack computing resources.

In the ECDH key agreement, the base station BS_1 and the router R_1 need to establish a session key before performing secure communication. Initially, both parties choose the same elliptic curve $y^2 = x^3 + ax + b$, and assign primes belonging to GF(P) as coefficients a and b, where P is equivalent to the Diffe-Hellman generator. BS_1 and R_1 each have a key pair containing ECC private key K, which is a random integer, and perform elliptic curve encryption and decryption with public key C, where $C = KP$. Additionally, (K_V, C_V) represents a key pair V, and (K_R, C_R) represents a key pair R.

Initially, the base station BS_1 selects a private key K_{BS1}, and then calculates $C_{BS1} = K_{BS1}P$. The router R_1 also selects a private key K_{R1} and calculates that $C_{R1} = K_{R1}P$.

Subsequently, the base station BS_1 transmits $C_{BS1} = K_{BS1}P$ to the router R_1, and R_1 transmits $C_{R1}= K_{R1}P$ to the base station BS_1. When each party receives the message sent by the other party, it then multiplies its private key by the received message. Eventually, both sides can figure out the same session key $S_{BS1R1} = K_{BS1}C_{R1} = K_{BS1}K_{R1}P = K_{R1}K_{BS1}\ P = K_{R1}C_{BS1}$. In the same way, the session keys between routers can be deduced in this study as S_{R1R2}, S_{R2R3} and S_{R3R4}, as shown in the blue line of Fig. 4.

3.2. The method of Secure Data Transmission in Cloud

After BS_1 receives the message sent by V_1, in order to protect the security of cloud data transmission, this study employs ECDH key agreement to secure the data transmission in cloud environment. First of all, BS_1 exploits the common session key of BS_1 and R_1, represented as S_{BS1R1}, to encrypt and protect the received Merkle tree root HMAC of blockchain, timestamp, type of service and the routing path, then transmits the encrypted output to the R_1 router.

$\#BS_1 \rightarrow R_1$.

$EK_{SBS1R1}[(BS_1)|ToS|TS|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]$

After receiving, R_1 decrypts the encrypted data via the common session key S_{BS1R1}, and appends its *ID* to the routing path. Subsequently, according to the routing table, R_1 and the next router R_2 cooperatively figure out the common session key S_{R1R2} utilizing the ECDH agreement, and then R_1 enciphers the entire data $[(R_1, BS_1)|ToS|TS|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]$ and forwards the enciphered result to the R_2 router.

$\#R_1 \rightarrow R_2$

$EK_{R1R2}\ [(R_1, BS_1)|ToS|TS|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]$

When R_2 obtains the transferred data, it then deciphers the enciphered data utilizing the S_{R1R2} common session key, then and appends itself *ID* to the routing path. Subsequently, according to the routing table, R_2 and the next router R_3 cooperatively figure out the S_{R2R3} common session key utilizing ECDH, then and R_2 enciphers the entire data $[(R_2, R_1, BS_1)|ToS|TS|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]$ and forwards the enciphered result to R_3.

$\#R_2 \rightarrow R_3$

$EK_{R2R3}[(R_2, R_1, BS_1)|ToS|TS|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]$

When R_3 obtains the transferred data, it then deciphers the enciphered data via the S_{R2R3} common session key, and appends itself *ID* to the routing path. Subsequently, according to the routing table, R_3 and the next router R_4 cooperatively figure out the S_{R3R4} common session key utilizing ECDH, and subsequently R_3 encrypts the entire data $[(R_3, R_2, R_1, BS_1)|ToS|TS|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]$ and forwards the encrypted result to R_4.

$\#R_3 \rightarrow R_4$

$EK_{R3R4}[(R_3, R_2, R_1, BS_1)|ToS|TS|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]$

Once R_4 obtains the transferred data, it then deciphers the enciphered data via the S_{R3R4} common session key, then and appends itself *ID* to the routing path. Subsequently,

according to the routing table, R_4 and the next router R_5 cooperatively figure out the S_{R4R5} common session key utilizing ECDH, and later R_4 encrypts the entire data [(R_4, R_3, R_2, R_1, BS_1)|*ToS*|*TS*|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]] and forwards the encrypted result to R_5.

#R_4→R_5

EKR4R5[(R4, R3, R2, R1, BS1)|ToS|TS|[H(H(H(SD1)|H(SD2))|H(H(SD3)|H(SD4)))||(SD1|SD2|SD3|SD4)]]

After receiving, R_5 executes the aforementioned similar operations and transfers the enciphered result to the PKI VM$_m$.

#R_5→PKI VM$_m$

EK$_{R5VMm}$[(R_5, R_4, R_3, R_2, R_1, BS_1)|*ToS*|*TS*|[H(H(H(S_{D1})|H(S_{D2}))|H(H(S_{D3})|H(S_{D4})))||(S_{D1}|S_{D2}|S_{D3}|S_{D4})]]

Similarly, the PKI VM$_m$ deciphers the enciphered data via S_{R5VMm}, then and determines the type of service according to *ToS*. Since the PKI VM$_m$ is the master virtual machine, it is responsible for the Map/Reduce operations. Subsequently, the PKI VM$_m$ initials the secure Map/Reduce operations as follows.

3.3. The Secure Map/Reduce Operations by Direct User Authentication using the Conference Key Agreement

For confirmation of identity accuracy of back-end servers participating in cloud computing and the security of data transmission between each other, this study employs the conference key protocol of indirect user confirmation function to obtain a common conference key and then achieve a secure Map/Reduce data transfer protocol. The entire system divides the operations into two parts that are prepare phase and the phase of the distribution of the conference key.

Prepare phase:

First step. The PKI VM$_m$ selects N, q, r, s and d as parameters, where N=qr, sd=1 mode L and L=lcm(q-1, r-1).

Second step. The PKI VM$_m$ generates an n dimensional vector B=(b_1,b_2,…,b_n), where $1 \leqq b_i \leqq L$-1 and $1 \leqq i \leqq n$. When B is determined, P can be figured out. Here, this study lets P = (h^{b1} mod N, h^{b2} mod N,…, h^{bn} mod N) = (h_1,h_2,…,h_n), where h is distributed at the root of GF(q) and GF(r). Since this system is verified by name, therefore each participant VM$_i$ has a public ID_i. Subsequently, VM$_i$ applies to the PKI VM$_m$ for registration of the key pair (Z_i, K_i) using binary ID_i. Where Z_i is the secret key of VM$_i$. and binary code ID_i = (D_{i1}, D_{i2},…D_{ik}), $D_{ij} \in \{0, 1\}$, $1 \leqq j \leqq k$. Here we can adopt the MAC address as ID_i. When the PKI VM$_m$ receives ID_i, then it uses a one-way hash function H to compute ID_i so that H(ID_i)=(x_{i1}, x_{i2},…, x_{in}), where $x_{ij} \in \{0, 1\}$, $1 \leqq j \leqq n$. Subsequently, the PKI VM$_m$ calculates the VM$_i$'s secret key Z_i=(ID_i)$^{-d}$ mod N.

Distribution of the conference key:

This study adopts the PKI VM$_m$ as the master, and VM$_1$, VM$_2$,…VM$_{m-2}$, VM$_{m-1}$ are virtual machines. There are M virtual machines joining this operation. Additionally, VM$_1$~VM$_{m-2}$ are also mappers that participate in Map/Reduce operations, and VM$_{m-1}$ is

the Reducer. The following steps describe the distribution of the conference key, and Fig. 6 describes the detailed procedure.

First step. The PKI VM_m selects a key K from 1 to N-1 as the conference key. Then, for each participant VM_i, the PKI VM_m must calculate $H(ID_i)=(x_{i1}, x_{i2},\ldots x_{in})$, $F_i=\prod_{l=1}^{n} (h_l \bmod N)^{xil} \bmod N = h^{ki} \bmod N$.

Second step. After the PKI VM_m calculates F_i for each participant VM_i, it picks an arbitrary number w and computes $C_1 = h^{sw} \bmod N$, $C_2 = Z_M h^{H(t,c1)w} \bmod N$, where H is the hash function announced by the system, and two parameters there are a time stamp t and $K_{VMi}=(F_i)^{sw} \bmod N$.

Third step. When all participants VM_i, where $1 \leqq i \leqq M$-1, have K_{VMi}. Then the PKI VM_m can construct the following Lagrange interpolation polynomial. Subsequently, the PKI VM_m broadcasts (C_1, C_2, a_0, a_1,…, a_{m-2}, t) to VM_i.

$$A(x)=\sum_{s=1}^{M-1}(K + ID_s) \prod_{j=1, j\neq s}^{M-1} \frac{(X-K_{VMj})}{(K_{VMs}-K_{VMj})} \bmod N = a_{M-2}X^{M-2}+\ldots+ a_1X+ a_0 \bmod N. \quad (8)$$

Forth step. After receiving (C_1, C_2, a_0, a_1, …, a_{M-2}, t), VM_i calculates h(t, C_1) and verifies whether the following equation is correct.

$$\frac{(C_2)^s}{(C_1)^{H(t1,C1)}} \equiv ID_M \bmod N. \quad (9)$$

If the verification of the above equation is accurate, the participant VM_i can identify the PKI VM_m to avoid a request from a counterfeiter. VM_i subsequently calculates the following equation.

$K_{VMi} = (C_1)^{ki} \bmod N = h^{swki} \bmod N$. (10)

Eventually, VM_i can figure out the conference key K of this task using the $A(x)$ polynomial.

$A(K_{VMi})=a_{M-2} K_{VMi}+\ldots+a_1K_{VMi} +a_0 \bmod N = K + ID_i \bmod N$, and (11)

$K \equiv K + ID_i - ID_i \pmod N$.

After the identities of all participants in the Map/Reduce operation are confirmed, the system can employ the K conference key to execute secure Map/Reduce encryption and decryption operations as shown in Fig. 7.

When the cloud computing center PKI VM_m receives data from the router R_5, it then uses the hash function to confirm the received data integrity, then and employs the conference key K to encrypt the transmitted data $[H(S_{Di})||S_{Di}]$ and deliver the result to the members of joining Map/Reduce operations, VM_1, VM_2,….VM_{m-2},VM_{m-1}, where VM_{m-1} is the reducer and VM_m is the PKI and master.

When the mapper VM_i receives the encrypted data$[H(S_{Di})||S_{Di}]$ as shown in Fig. 7-①, and then decrypts it utilizing the conference key K and obtains $[H(S_{Di})||S_{Di}]$. Subsequently, VM_i uses the hash function to calculate the newer $H^*(S_{Di})$. If $H(S_{Di})= H^*(S_{Di})$, then the data has not been modified during the data transmission, and thus this study can ensure the integrity of transmitted data as shown in Fig. 7-②. Subsequently, VM_i encrypts $[ID_{VM1}, TS, (H(S_{D1})|S_{D1})]_{EKk}$ and delivers the encrypted result to the Reducer VM_{m-1} as shown in Fig. 7-③.

When the reducer VM_{m-1} receives data segments from VM_1, VM_2,…VM_{m-2}. Each data segment is decrypted utilizing the conference key K to verify the $(H(S_{Di})|\ S_{Di})$ integrity, where $1 \leqq i \leqq m\text{-}2$, using the hash function. If $H(S_{Di})=H^*(\ S_{Di})$, where * represents the newer HMAC result. Then the reducer merges each data segment S_{D1}~S_{Dm-2} to obtain the complete original data D as shown in Fig. 7-④, and subsequently the reducer encrypts the result utilizing the conference key K and sends the encrypted result back to the PKI VM_m to complete the secure Map/Reduce operations as shown in Fig. 7-⑤. The complete secure Map/Reduce operation is as shown in the Fig. 7.

PKI VM_m	VM_i,
1.Selects a key K from 1 to N-1 as the conference key 2.Calculates $H(ID_i) = (x_{i1},x_{i2},\ldots x_{in})$, $F_i=\prod_{l=1}^{n} (h_l \ mod\ N)^{xil} \bmod N = h^{ki} \bmod N$ for each participant VM_i 3.Picks a stamp t, $K_{VMi} = (F_i)^{sw} \bmod N$ and an arbitrary number w and then computes $C_1 = h^{sw} \bmod N$, $C_2 = Z_M h^{H(t,c1)w} \bmod N$	
	4.Then all participants VM_i have K_{VMi}.
5.VM_m constructs the Lagrange interpolation polynomial as the equation (7) 6.Broadcasts $(C_1, C_2, a_0, a_1,\ldots, a_{m-2}, t)$ to VM_i.	
	7.VM_i calculates $h(t,C_1)$ and verifies whether $\frac{(C_2)^s}{(C_1)^{H(t1,C1)}} \equiv ID_M \ mod\ N$ is correct 8.VM_i calculates $K_{VMi} = (C_1)^{ki} \bmod N = h^{swki} \bmod N$ 9.VM_i figures out the conference key K using the equation (7)

Fig. 6. The procedure of obtaining the common conference key

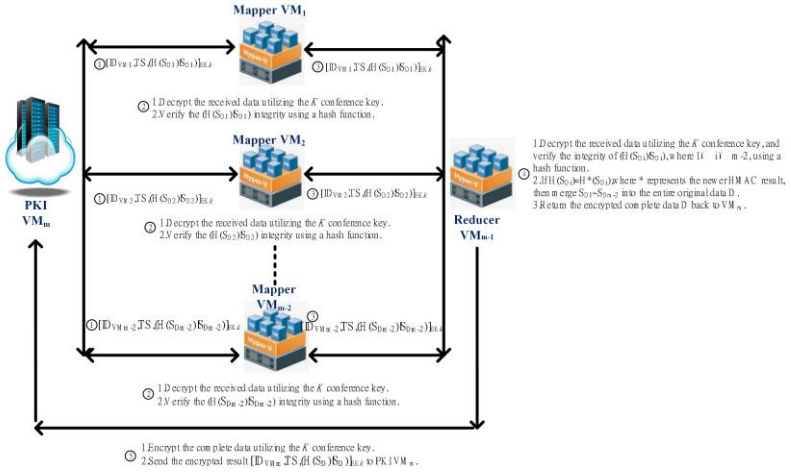


Fig. 7. The secure Map/Reduce operation

4. Secure Data Transmission Analyses and Computing Evaluations

This investigation focuses on the evaluation of the efficiency of the aforementioned models for secure data transmission and conducts several security analyses. In addition, this study also shows that the proposed methods improve the efficiency of secure data transfer, and reduce the recomputing time of blockchain when nodes depart or participate. As well, this study evaluates the convergence time of performing the interpolation polynomial and the verification of multisignature. In addition, our proposed decentralized key management without CA or TA is equipped with a flexible and upgradeable framework. Here is the comprehensive security analysis.

(1) Data integrity: Adding the prev_hash value to blockchain provides a framework for the verification of the integrity of the entire blockchain. If a hacker modifies some past transaction in the previous block N-1, the hash value in the later block N will be invalid, even if the hacker modifies the Merkle tree and the root value in block N-1. Since, the distribution character of blockchain handles such mismatches in hash [27][28].

(2) Avoid single point of failure: Because the data of blockchain is typically stored among many vehicles in a decentralized network, the system and data resists technical breakdowns and malicious attacks very well. Every node within this network can replicate and store copies of the database. This means there is no single point of failure, although an offline single node does not affect network availability or security [29]. On the other hand, many traditional databases are based on one or more servers and are more susceptible to mechanical failures and network offences.

(3) Transparent: In this open system, the private information in the transaction is encrypted, and all vehicles with maintenance functions collectively maintain it. Anyone can query each blockchain data through the open interface, so the whole system information is highly transparent.

(4) Confirmation of identity in cloud computing: Once VM_i receives (C_1, C_2, a_0, a_1,…, a_{M-2}, t) to calculate h(t, C_1) and verify whether the following equation is correct.

$$\frac{(C_2)^s}{(C_1)^{H(t1,C1)}} \equiv ID_M \ mod \ N. \tag{12}$$

If the above equation is accurate, the participant can verify the identity of the PKI VM_m to avoid a request from a counterfeiter.

(5) Better execution efficiency: When a vehicle leaves or joins the network of the CIoVs environment, the Markle root value must be recalculated. Due to the b-tree architecture, a new Markle root value can be obtained by recalculating the branch derived from this node to the root. If there are N vehicle communications and the height of the binary tree is Log_2N, so as long as after Log_2N stages of the operation can get the new Markle root value.

(6) The number of stages for computing the entire blockchain: When the vehicle takes part in or departure from the CIoVs environment, this study evaluates that the number of stages required by blockchain must be recalculated. As shown in Table 2, in the best case, the number of stages for recomputing the entire blockchain is only $\log_2M$ when a vehicle located in the block number N departs from this system, where M is the

amount of vehicles. However, in the worst case, the amount of stages for recomputing the entire blockchain will increase to $N * (\text{Log}_2 M)$ when a vehicle located in the block number 2 leaves. Generally, the number of stages for recomputing the entire blockchain will increase to $(i\text{-}1) * (\text{Log}_2 M)$ when a vehicle located in the block number i leaves.

Table 2. The number of stages for recomputing the entire blockchain

Various case	Best case	General case	Worst case
M vehicles in each block	A leaving/joining vehicle in block N	A leaving/joining vehicle in block i	A leaving/joining vehicle in block 2
The number of stages	$\text{Log}_2 M$	$(i\text{-}1) * (\text{Log}_2 M)$	$N * (\text{Log}_2 M)$

(7) The convergence time of computing the entire blockchain: For the convenience of discussion, under the condition of fixed blockchain number N=4, the convergence time of computing the entire blockchain of this system is discussed under the best case, general case and worst case with the growth in the amount of vehicles. Fig. 8 depicts that as the number of vehicles increases, the convergence time of computing the entire data increases. Especially, in the worst case, the convergence time steeply bumps up, since it must to recalculate N-1 blocks. Additionally, in the best case, this system only needs to recalculate 1 block, and therefore the convergence time increases stably.

(8) The convergence time of calculating the conference key: For the sake of convenient discussion, under the condition of the fixed reducer number is 1. This study evaluates that the convergence time of calculating the conference key K needs to perform the interpolation polynomial to figure out the conference key K. Figure 9 depicts that as the number of the mapper virtual machine M raises, this system has to calculate the M-2 degree of polynomial. As the result, the convergence time of obtaining the conference key K increases as polynomial and exponential growth.

(9) Blockchain mainly generates Merkle tree root through hash function operations, so SHA1 used in this study is compared with various common hash functions for efficient evaluation. We take the file size from 0M bytes to 60M bytes as the input sample, and after 1000 tests the performance is as shown in the Figure 10. The average execution time of the MD5 algorithm for 1000 times is near 230ms, the SHA1 algorithm for 1000 times is near 320ms, and the SHA256 algorithm for 1000 times is near 480ms. In terms of security, SHA256 is obviously the most secure, but it takes considerably longer than the other two [30]. MD5 is comparatively straightforward to generate collisions and be cracked, so SHA1 is the most effective encryption algorithm among the three. In consideration of the need for better computing performance due to the rapid change of vehicles and topologies, this survey employs SHA1 as the hash function.

(10) The verification of multisignature: Any member can verify the signature of the plaintext D according to the only announced public key z. As we know, each vehicle member's signature $\{w_i, S_i\}$ satisfies the following equation.

$$z_i^{D'} = w_i^{w} \alpha^{Si} \bmod p. \tag{13}$$

When the above equation is multiplied for n times, where $i = 1, 2, 3\ldots, n$. This study can infer that the multisignature $\{w, S\}$ is correct as described below.

$$\prod_{i=1}^{n} z_i^{D'} = \prod_{i=1}^{n} w_i{}^{w}\alpha^{Si} \bmod p.$$
$$\prod_{i=1}^{n} (z_i)^{D'} = (\prod_{i=1}^{n} w_i)^{w}\alpha^{S1+S2+\ldots+Si} \bmod p. \quad (14)$$
$$z^{D'} = w^{w}\alpha^{S} \bmod p.$$

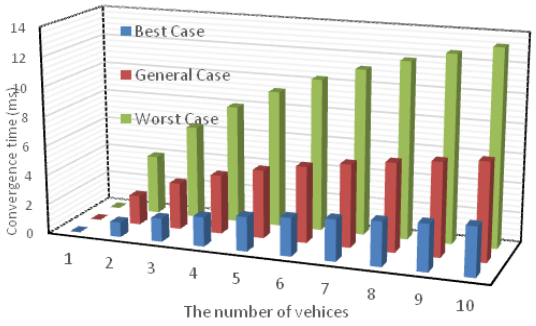


Fig. 8. Under the fixed number of blocks N=4

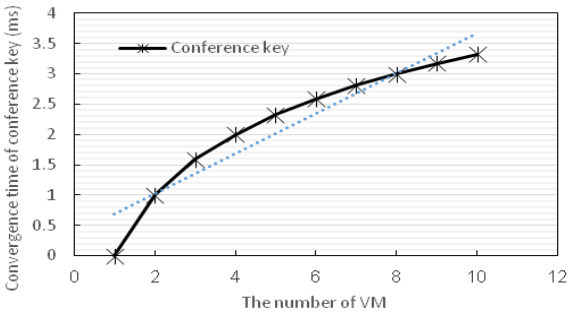


Fig. 9. The convergence time of calculating the conference key

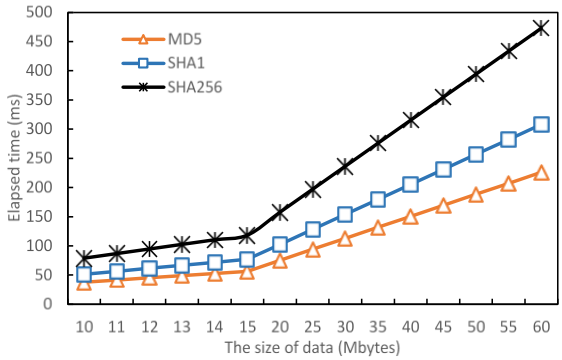


Fig. 10. The elapsed time of generating the Merkle tree root

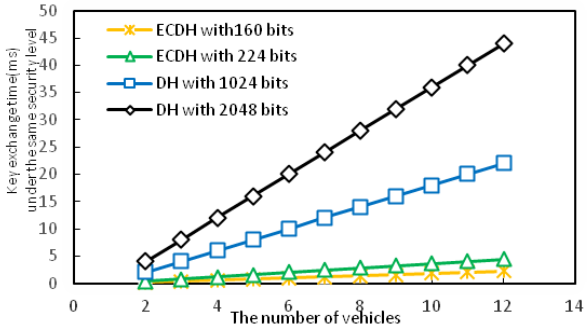


Fig.11. Under the same security level, the key exchange time along with the various number of vehicles

(11) The comparison of DH and ECDH key exchange protocols: This study compares the DH and ECDH key exchange protocols to achieve the same security strength under different key lengths, and the time required by both parties to encipher and decipher the transferred data utilizing the session key generated by DH and ECDH. When the transferred data traverses a different number of vehicles, we compare the time required by pairwise encryption and decryption. Figure 11 depicts that under the same security strength, DH needs 1024 (2048) bits. However, ECDH requires only 160(224) bits. Along the transmission path, the required time for the encryption and decryption of adjacent pairwise vehicles increases as the number of vehicles passing by increases. However, experimental results show that the required time for the encryption and decryption of ECDH is less than that of DH.

5. Conclusion

Since the development of the cloud Internet of vehicles must encounter the problems and requirements of identity authentication and security trust. This paper depicts the security issues among the vehicle, RSUs, network and cloud, and therefore proposes a blockchain strategy that is different from the conventional centralized CA authentication and privacy protection mechanism to ensure the secure data transfer of the cloud Internet of vehicles. The characteristics of this study are as follows. 1. The proposed scheme can reduce computing resources and costs without third-party verification; 2. Decentralization makes it difficult to tamper with transmitted data; 3. Transactions are secure, private and highly efficient, and 4. Data transparency. Finally, we propose a secure Map/Reduce operations by direct user authentication of the conference key agreement to deal with the security computing of the cloud system, so that the application of blockchain technology among the vehicle, RSUs, network and cloud in CIoVs is more complete.

As the cloud Internet of vehicles integrate 5G, artificial intelligence, huge amounts of data, blockchain and other forward-looking technology of emerging industries, there will be a huge security demand for intelligent construction in the near future.

Acknowledgment. This paper is supported by the Ministry of Science and Technology (MOST), Taiwan, under grants MOST 111-2221-E-163-002.

References

1. Anusha, V., Basudeb, B. Sourav, S., Ashok, K. D., Neeraj, K., Youngho, P.: Blockchain-Enabled Certificate-Based Authentication for Vehicle Accident Detection and Notification in Intelligent Transportation Systems. IEEE Sensors Journal 21(14), 15824-15838. (2021)
2. Angtai, L., Guohua, T., Meixia, M., Jianpeng G.: Blockchain-based cross-user data shared auditing. Connection science 34(1), 83-103. (2021)
3. Qilei, R., Ka, L. M., Muqing, L., Bingjie, G., Jieming, M.: Intelligent design and implementation of blockchain and Internet of things-based traffic system. International Journal of Distributed Sensor Networks 15(8). (2019)
4. Uzair, J., Muhammad, N. A., Biplab, S.: A Scalable Protocol for Driving Trust Management in Internet of Vehicles With Blockchain. IEEE Internet of Things Journal 7(12), 11815-11829. (2020)
5. Insaf U., Noor, U. A., Muhammad, A. K., Hizbullah, K., Saru, K.: A Lightweight and Provable Secured Certificateless Signcryption Approach for Crowdsourced IIoT Applications. Symmetry 11(11), 1386. (2019)
6. Pandi, V., Maria, A., Sergei, A. K., & Joel, J. P. C. R.: An Anonymous Batch Authentication and Key Exchange Protocols for 6G Enabled VANETs. IEEE Transactions on Intelligent Transportation Systems 23(2), 1630-1638. (2022)
7. Jianfeng, M., Tao L., Jie C., Zuobin, Y., Jiujun, C.: Attribute-Based Secure Announcement Sharing Among Vehicles Using Blockchain. IEEE Internet of Things Journal 8(13), 10873-10883. (2021)
8. Tian, L., Xuchong, L., Ruhul, A., Wei L., Meng, Y. H.: RETRACTED ARTICLE: Cloud enabled robust authenticated key agreement scheme for telecare medical information system. Connection science 33(4), 1-XX. (2021)
9. Yuting, Z., Zhaozhe, K., Zhaozhe, K.: BCAS: A blockchain-based ciphertext-policy attribute-based encryption scheme for cloud data security sharing. International journal of distributed sensor networks 17(3). (2021)
10. Dong, W., Huanjuan, W., Yuchen, F.: Blockchain-based IoT device identification and management in 5G smart grid. EURASIP Journal on Wireless Communications and Networking 125. (2021)
11. Houshyar, H., Pajooh, Mohammed, A. R., Fakhrul, A., Serge, D.: IoT Big Data provenance scheme using blockchain on Hadoop ecosystem. Journal of Big Data 8(114). (2021)
12. Razi, I., Talal, A. B., Muhammad, A. Khaled, S.: Trust management in social Internet of vehicles: Factors, challenges, blockchain, and fog solutions. International Journal of Distributed Sensor Networks 15(1). (2019)
13. Suaib, A. A. F. M., Mohiuddin, A., Shahen, S. A. F. M., Adnan, A., Kayes, A. S. M., Ahmet, Z.: Blockchain-Based Authentication Protocol for Cooperative Vehicular Ad Hoc Network. Sensors 21(4), 1273. (2021)
14. Xu, W., Xuan, Z., Wei N., Ren, P. L., Guo, Y. J., Xinxin, N., Kangfeng, Z.: Survey on blockchain for Internet of Things. Computer Communications 136, 10-29. (2019)
15. Zeng, W., Hui, H., Yuping, Z., Chenhuang, W.: A secure and efficient data deduplication framework for the internet of things via edge computing and blockchain. Connection Science 34(1), 1999-2025. (2022)
16. Azees, M., Vijayakumar, P., Deborah, L. J., Karuppiah, M., Christo, M. S.: BBAAS: Blockchainbased anonymous authentication scheme for providing secure communication in VANETs. Security and Communication Networks 2021(6679882). (2021)

17. ing, Z., Xiaoliang, W., Qing Y., Wenhui, X., Yapeng, S., Wei, L.: A blockchain-based lightweight authentication and key agreement scheme for internet of vehicles. Connection Science 34(1), 1430-1453. (2022)
18. Bagga, P., Sutrala, A. K., Das, A. K., Vijayakumar, P.: Blockchain-based batch authentication protocol for internet of vehicles. Journal of Systems Architecture 113(8), 101877. (2021)
19. Cui, J., Ouyang, F., Ying, Z., Wei, L., Zhong, H.: Secure and efficient data sharing among vehicles based on consortium blockchain. IEEE Transactions on Intelligent Transportation Systems 23(7), 8857-8867. (2021)
20. Muhammad, F., Sandi, R., Kyung, H. R.: Decentralized Trusted Data Sharing Management on Internet of Vehicle Edge Computing (IoVEC) Networks Using Consortium Blockchain. Sensors 21(7), 2410. (2021)
21. Li, X., Wang, Y., Vijayakumar, P., He, D., Kumar, N., Ma, J.: Blockchain-based mutual-healing group key distribution scheme in unmanned aerial vehicles ad-hoc network. IEEE Transactions on Vehicular Technology 68(11), 11309–11322. (2019)
22. Lu, Z., Wang, Q., Qu, G., Zhang, H., Liu, Z.: A blockchain-based privacy-preserving authentication scheme for VANETs. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27(12), 2792–2801. (2019)
23. Ma, J., Li, T., Cui, J., Ying, Z., Cheng, J.: Attribute-based secure announcement sharing among vehicles using blockchain. IEEE Internet of Things Journal 8(13), 10873–10883. (2021)
24. Hua, Y. L.: Integrate the hierarchical cluster elliptic curve key agreement with multiple secure data transfer modes into wireless sensor networks. Connection Science 34(1), 274-300. (2022)
25. Hua, Y., L., Meng, Y. H.: A dynamic key management and secure data transfer based on m-tree structure with multi-level security framework for Internet of vehicles. Connection science 34(1), 1089-1118. (2022)
26. Lin, H. Y., Hsieh, M. Y., Li, K. C.: Flexible group key management and secure data transmission in mobile device communications using elliptic curve Diffie-Hellman cryptographic system. International Journal of Computational Science and Engineering 12(1), 47-52. (2016)
27. Caixiang, F., Sara, G., Hamzeh K., Petr, M.: Performance Evaluation of Blockchain Systems: A Systematic Survey. IEEE Access 8, 126927-126950. (2020)
28. Wei, L., Lijun, X., Ke, Z., Mingdong, T., Dacheng, He., Kuan, C. L.: Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems. IEEE Internet of Things Journal 9(16), 14741-14751. (2021)
29. Wei, L., Yongkai, F., Kuan, C. L., Dafang, Z., Jean, L. G.: Secure Data Storage and Recovery in Industrial Blockchain Network Environments. IEEE Transactions on Industrial Informatics 16(10), 6543–6552. (2020)
30. Li, Z., Jianbo, X.: Blockchain-based anonymous authentication for traffic reporting in VANETs. Connection Science 34(1), 1038-1065. (2022)

Hua-Yi Lin received the Ph.D. degree in Engineering Science from the National Cheng Kung University, Taiwan, in 2006. He is currently an associate professor in Dept. of Information Management at the China University of Technology, Taiwan.

Received: September 21, 2022; Accepted: November 10, 2022.

Computer Science and Information Systems 20(1):157–173 ttps://doi.org/10.2298/CSIS220927055N

Reinforcement Learning - based Adaptation and Scheduling Methods for Multi-source DASH ⋆

Nghia T. Nguyen[1,2], Long Luu[1,2], Phuong L. Vo[1,2], Thi Thanh Sang Nguyen[1,2], Cuong T. Do[3], and Ngoc Thanh Nguyen[4]

[1] School of Computer and Engineering, International University, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
{ntnghia, vtlphuong, nttsang}@hcmiu.edu.vn
ITITIU18079@student.hcmiu.edu.vn
[3] Department of Computer Engineering, Kyung Hee University, 446-701, Korea
dtcuong@khu.ac.kr, dothecuong@gmail.com
[4] Faculty of Information and Communication Technology,
Wroclaw University of Science and Technology, Poland
ngoc-thanh.nguyen@pwr.edu.pl

Abstract. Dynamic adaptive streaming over HTTP (DASH) has been widely used in video streaming recently. In DASH, the client downloads video chunks in order from a server. The rate adaptation function at the video client enhances the user's quality-of-experience (QoE) by choosing a suitable quality level for each video chunk to download based on the network condition.

Today networks such as content delivery networks, edge caching networks, content-centric networks, *etc.* usually replicate video contents on multiple cache nodes. We study video streaming from multiple sources in this work. In multi-source streaming, video chunks may arrive out of order due to different conditions of the network paths. Hence, to guarantee a high QoE, the video client needs not only rate adaptation, but also chunk scheduling.

Reinforcement learning (RL) has emerged as the state-of-the-art control method in various fields in recent years. This paper proposes two algorithms for streaming from multiple sources: *RL-based adaptation with greedy scheduling* (RLAGS) and *RL-based adaptation and scheduling* (RLAS). We also build a simulation environment for training and evaluation. The efficiency of the proposed algorithms is proved via extensive simulations with real-trace data.

Keywords: multi-source streaming, reinforcement learning, proximal policy optimization, dynamic adaptation streaming over HTTP.

1. Introduction

A significant part of Internet traffic today is video streaming [1]. Dynamic adaptive streaming over HTTP (DASH) is the primary technique to stream a video from a server to a video player. In DASH, videos are encoded in multiple quality levels. Furthermore, videos are

⋆ This is an extended version of the conference paper [20].
This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2020-28-01. Dr. Phuong L. Vo is the corresponding author.

partitioned into video chunks. Each chunk contains media data in a short interval of playback time. Video players request the chunks with suitable quality levels based on the current network condition [2–5]. The downloaded chunks are buffered in the client's memory before being played. Buffer size is the total playing time of the wait-to-be-played chunks. When a new video chunk is successfully downloaded, the buffer size increases by a chunk length. When a chunk is played, the buffer size is decreased by the chunk length. The buffer size has an upper threshold level. When the buffer exceeds the threshold, the client will pause downloading a new chunk, wait for the buffer to decrease below the threshold, and then resume downloading. The client *rebuffers* when the chunk will be played is not in the buffer. Rebuffering causes video freezes.

The rate adaptation function in video clients is essential in providing a high quality-of-experience (QoE) for the user. Various adaptation methods are proposed for DASH. Throughput-based adaptation method chooses the quality level for the next chunk such that it does not exceed the estimated throughput [6,7]. The throughput is usually estimated by the mean or harmonic mean of several last requested chunks. The buffer-based methods observe the buffer level to decide the encoding quality level [8, 9]. Both the throughput-based method and BOLA, a buffer-based method [8], are employed in Dash.js reference client [6]. Some methods combine both these two approaches [10].

On the other hand, several networks today such as content delivery networks, edge caching networks, content-centric networks, *etc.* replicate popular videos at the routers to reduce network congestion and delay. Utilizing multiple sources to stream a video to a user is studied in this paper. When streaming from multiple sources, quality control is much more complicated than streaming from a single source. In multi-source streaming, the quality control includes not only *rate adaptation*, *i.e.*, choosing the quality levels for the chunks, but also *chunk scheduling*, *i.e.*, which chunk indices are requested on each path (see Fig. 1). Due to the difference in the network conditions of the connections, the chunks may arrive at the video client out of order. For example, assume that the maximum buffer size of the client is 3 chunks and there are two paths. With bad scheduling, path 2 is downloading chunk 2 while path 1, with very high throughput, already downloaded chunks 1, 3, 4. Therefore, the buffer is full, however, the video playing is frozen since the client waits for chunk 2.

Some previous works have studied multi-source streaming [11–13]. In [11], MSPlayer can download video content from multiple servers. The authors in [11] consider the chunks with only one quality level, however, the chunk size varies. They focus on the chunk scheduling problem. The client estimates the path quality to request chunk indices and chunk sizes for the paths. In work [12], MP-H2 protocol is designed on top of HTTP/2. MP-H2 splits the video into many chunks, and the client requests chunks over multiple network connections such as wi-fi and cellular. Chunk sizes are calculated based on bandwidth and round-trip-time of the connections. A chunk scheduling algorithm is then used to download the chunks over multiple paths. No adaptation method is proposed in [12]. The work [13] has proposed a bitrate adaptation algorithm for DASH, called DQ-DASH, that allows downloading multiple video chunks from various servers in parallel to enhance QoE. Distributed queueing theory is applied to address the situation when multiple clients send requests to many servers simultaneously. Fair QoE across clients is considered in the model. Different from [11–13], our proposed framework jointly considered rate adaptation and chunk scheduling.

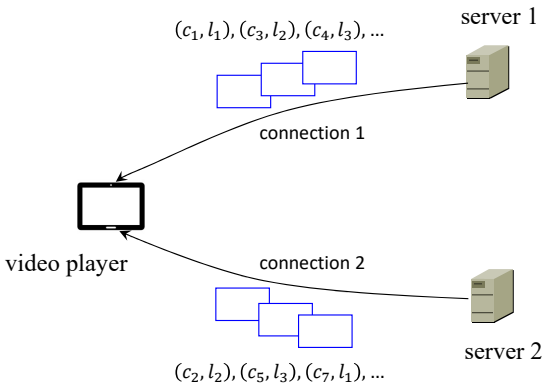


Fig. 1. Multi-source video streaming.

Reinforcement learning (RL) has been widely used in many fields recently. The works [14–16] have applied RL algorithms for single-source adaptive streaming. The actions of RL agents in these works are the quality levels of video chunks. The work [14] applied a Q-learning method for DASH. Buffer and network bandwidth are discretized for the discrete state space. The study [15] applied Asynchronous Advantage Actor-Critic (A3C) algorithm for rate adaptation. The work [16] has proposed D-DASH that applied a Deep Q-learning to choose the quality level for the chunks.

In this work, we also use RL algorithm for rate adaptation and chunk scheduling in streaming from multiple sources. However, there are several challenges we cannot simply extend the RL framework for single-source streaming in [15,16] to multi-source streaming straightforwardly:

- The action space of multi-source streaming must be redesigned to integrate scheduling. An action must include a chunk index and a quality level.
- The RL algorithms need a simulation environment to train the model. In the environment for single-source streaming, when the agent takes action, *i.e.*, downloads a new chunk, the environment immediately returns a reward value associated with that chunk, which is calculated from the utility of the chunk's quality, the quality switch penalty between two consecutive chunks and the rebuffering penalty. The rebuffering penalty is calculated when action is taken. However, in multi-source streaming, where the chunks may arrive to the client out of order, the simulation environment cannot estimate the rebuffering time right when an action is taken.
- The simulation environment for the single-source streaming is open [15]. However, the simulation environment for the multi-source streaming is not available in the literature, as far as we know.

Multipath transmission control protocol (MPTCP) [17–19] also utilizes multiple paths to transmit data from a source to a destination. It is shown that MPTCP achieves high throughput, provides a smooth hand-off, and improves the high availability of TCP connection. However, the MPTCP adoption has been prolonged because of the middlebox problem. Moreover, it requires modifying the kernels of both client and server. Our proposed framework can be applied to stream a video from a single source to a video player

over multiple paths as MPTCP does. However, the MPTCP is a source-control protocol at the transport layer, whereas our proposed multi-source streaming is a client-based control protocol at the application layer. Therefore, the proposed multi-source streaming does not need to modify the kernel as well as overcomes the middlebox problem.

The contributions of our work include:

1. We propose two RL-based frameworks for rate adaptation and chunk scheduling in multi-source streaming called RL-based adaptation with greedy scheduling (RLAGS) and RL-based adaptation and scheduling (RLAS).
2. We build an environment, which is an event-driven simulation that simulates a client downloading chunks from multiple sources and playing the chunks for training and testing.
3. We conduct extensive simulations with real-trace bandwidth to evaluate the performance of the proposed methods. Both RLAGS and RLAS outperform the other baseline methods used with greedy scheduling for multi-source streaming, *i.e.*, throughput-based and BOLA. The source code is available at `https://github.com/ntnghia1908/Master_Thesis`.

This is the extended work of our conference paper [20]. In this paper, RLAS improves the proposed algorithm in [20] by using invalid action masking to avoid duplicate downloads. In addition, we propose RLAGS algorithm with greedy scheduling. We also add more evaluations in various network scenarios. The outline of the paper is as follows. Section I has presented the motivation and related works. Section II describes the RL model applied in rate adaptation and chunk scheduling for video streaming from multiple sources. The simulation environment and results are presented in Section III, and Section IV concludes the work.

2. Reinforcement learning frameworks for DASH

This section describes the RL framework, including reward function, action space, and state space. Two chunk scheduling policies are considered, *i.e.*, greedy and RL-based scheduling, which leads to two proposed algorithms, RLAGS and RLAS, respectively.

2.1. Reward

We apply a similar reward function used in [15,16], which captures utility, switch penalty, and rebuffering penalty. Assume that a time step begins when the client requests a video chunk. The episode ends when the client finishes playing the video.

Reward for single-source streaming: Assume that step t starts when the client requests for chunk t, the reward at step t in single-source adaptive streaming is given by [15, 16] (see Table 1 for the notation descriptions):

$$r_t = q_t - \beta \mid q_t - q_{t-1} \mid -\gamma\phi_t - \delta[\max(0, B^{min} - B_t)]^2, \quad t = 2, \ldots, N, \quad (1)$$

where

Table 1. Main notations

Notations	Descriptions
$B^{\max}$	maximum buffer size (in seconds)
N	number of video chunks
L	number of quality levels in action space
W	number of chunks in action space
$\mathcal{A}$	action space
r_t	reward estimated at step t
s_t	environment state at step t
q_i	utility of quality level i
β	quality-switch coefficient
γ	rebuffering coefficient

- q_t is the utility corresponding to the quality level of chunk t;
- $\mid q_t - q_{t-1} \mid$ penalties the difference in quality levels between two consecutive chunks;
- ϕ_t is rebuffering time is seconds;
- $[\max(0, B^{min} - B_t)]^2$ is an optional penalty that is applied whenever the buffer level is below a threshold B^{min}. This term helps to reduce the risk of rebuffering.

If d_t, *i.e.*, the download time of chunk t, is greater than remaining time in buffer, which is B_t, then rebuffering time ϕ_t is $d_t - B_t$, otherwise, there is no rebuffering. Hence, the rebuffering time associated with chunk t in single-source streaming is given by the following formula

$$\phi_t = \max(0, d_t - B_t). \tag{2}$$

Reward for multi-source streaming: Formula (2) is no longer correct in the multi-source streaming environment since the buffer at the client may not store consecutive chunks. For example, the buffer may have chunks 3, 5, 6, and 7, while chunk 4 has not fully received on the low-throughput path. Therefore, in the multi-source environment, the reward is estimated when playing chunks in a step. Let a step start when the client requests a chunk and end when the client requests a new chunk or reaches the end of the episode. The reward returned at step t in the multi-source streaming environment is given by

$$r_t = \sum_i q_i - \beta \sum_i \mid q_i - q_{i-1} \mid - \gamma \phi_t, \tag{3}$$

where i is any chunk index played, and ϕ_t is the cumulative rebuffering time in step t. The terms $\sum_i q_i$, $\beta \sum_i \mid q_i - q_{i-1} \mid$, and $\gamma \phi_t$ are called *utility*, *switching penalty*, and *rebuffering penalty*, respectively.

2.2. State space

The state s of the proposed reinforcement learning frameworks includes the following components

- vector of network throughput measurements of last 06 video chunks on each path;
- vector of chunk sizes of L quality levels of next W chunks count from playing chunk (length $W \times L$);
- the vector of quality levels of next W chunks counted from the playing chunk, if the chunks have not yet downloaded, their quality level is set to 0;
- current buffer size in seconds;
- number of remaining chunks that have not yet played;
- quality level of the playing chunk; and
- download times of last 06 video chunks on each path.

2.3. Scheduling policies and action spaces

We assume that the request for a new chunk is sent on a path right after the downloading chunk on that path is fully received if the buffer size is under $B^{\max}$. Otherwise, the client will pause sending a new request. Let's consider two scheduling policies, *i.e.*, greedy scheduling and RL-based scheduling, corresponding to two proposed methods, RLAGS and RLAS, respectively.

Greedy scheduling In greedy scheduling, the chunk is requested in order. When the client downloads a new chunk from a source, it requests the chunk index, the smallest index that has not been or is being downloaded. Therefore, RLAGS agent only decides the quality level of the chunk to request. The action space of RLAGS includes the quality levels of video chunks:

$$\mathcal{A}^{\mathrm{RLAGS}} = \{l_i | i = 1, \ldots, L\}, \tag{4}$$

where L is the number of quality levels of video.

For example, in Fig. 2 (upper figure), chunks 1-3 have been played, chunk 4 is playing, and chunk 5 has already been requested. The next request is for chunk 6, with the quality level decided by RLAGS.

RL-based scheduling RLAS method uses RL-based scheduling. When the client requests a new chunk, the RL agent decides both the index and quality level. Assuming that the maximum number of chunks that can be stored in the video buffer is W, the number of quality levels is L. Action space of RLAS method is defined as

$$\mathcal{A}^{\mathrm{RLAS}} = \{(c_i, l_j) | c_i \in [1, W], j \in [1, L]\}. \tag{5}$$

If the playing chunk is c_t and the RL agent takes action $a_t = (c_i, l_j)$ to download chunk on a path at time step t, the client will download chunk index $c_t + c_i$ at quality l_j on this path.

For example, in Fig. 2, if quality levels for each chunk are *low*, *medium*, and *high* ($L = 3$), $W = 4$. Assume that at current time t, the playing chunk is $c_t = 4$ and the RL agent takes action $a_t = (3, 2)$. It means that the agent will download chunk index $4 + 3 = 7$ with quality level 2, which is *medium* quality (see Fig. 2).

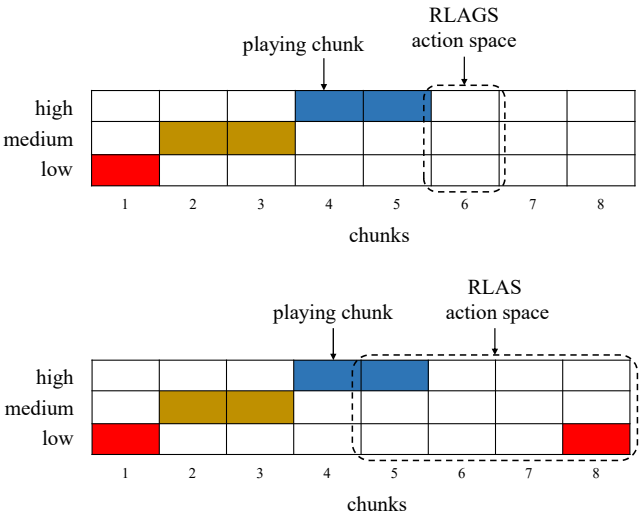


Fig. 2. The action spaces of RLAGS and RLAS. (The shade regions represent the chunks that have been requested.)

Invalid action masking With RLAS, there are invalid actions in some steps. Firstly, the two-dimension action space of RLAS allows the possibility of re-download the same chunk index again if that chunk has not been played. The chunk index already downloaded is considered an invalid action to avoid duplicate downloads. Secondly, since RLAS's action space is a sliding window that shifts forward by one chunk when a new chunk is played, some actions are *invalid* when the number of remaining chunks is less than the window side W. Hence, the valid actions of RLAS are given by

$$\{(c_i, l_j) \mid c_i \in [1, \min[W, N - c_t], c_i \text{ has not been requested}, j \in [1, L]\}, \quad (6)$$

where N is the number of chunks of the video, and chunk c_t is the chunk being played.

There are several approaches to dealing with invalid actions. Two common ones are *invalid action penalty* and *invalid action masking*. With the invalid action penalty approach, the rewards resulting from the invalid actions are set to negative values. With invalid action masking, the action is sampled among the valid actions in each step. These approaches are well investigated and implemented in the work [21]. With policy gradient algorithms, invalid action masking is shown theoretically and empirically that it outperforms the other approaches, particularly with the state-of-the-art proximal policy optimization (PPO) algorithm in the experiments [21]. Therefore we also apply PPO in our evaluations.

2.4. PPO for multisource DASH

PPO is a policy gradient algorithm that uses two networks, actor and critic, like A2C or A3C. The actor network estimates the policy directly from the state. A baseline is sub-

tracted from the return to reduce the variance of a policy gradient algorithm. A common-used baseline is the value function, which is estimated by a critic network. To accelerate the training, PPO algorithm could also use multiple copied environments in parallel, similar to A2C and A3C.

However, PPO is an improvement from A2C/A3C. To prevent the catastrophic drop in the performance of the traditional actor-critic algorithms, PPO constraints the change in policy between two consecutive training steps by introducing a new clipped surrogate objective. PPO has shown a reliable performance and is used in many RL applications. Please see the detail of PPO algorithm in [25].

We utilize Stable Baseline3 [26] library to implement PPO in training and evaluation. Stable Baseline3 includes a set of reliable implementations of deep RL algorithms and is used in many applications. The invalid action masking function is also provided with PPO in Stable Baseline3.

3. Evaluations

3.1. Event-driven environment

We build an environment that simulates the streaming from two sources, emulating the practical scenarios, *e.g.*, a cell phone uses Wi-Fi and 4G to connect to video servers, or a laptop connects via Ethernet and Wi-Fi simultaneously. Scenarios with more than two sources can be easily extended by modifying *reset* function. The simulation environment follows Gym interface to be able to use Stable Baseline3 [26].[5]

The environment emulates a client downloading chunks on two paths parallelly and playing the received chunks. An array-type buffer, which stores downloaded chunk indices, is maintained during an episode. When the client fully receives a chunk, the chunk index is appended to the buffer, and the buffer size increases by a chunk length. The client plays the chunks stored in the buffer sequentially. If a chunk is played, that chunk index is removed from the buffer, and the buffer size increases by a chunk length.

There are four main events, *i.e.*, DOWN, PAUSE, PLAY, REBUFFER. Every event has a timestamp, and the program runs through the events iteratively in time order until the end of the episode. DOWN and PAUSE events are associated with a path index, whereas PLAY and REBUFFER events are not.

- A DOWN event simulates sending a request for a chunk, say c_t, on a path. When the program encounters a DOWN event at time t, at timestamp *t + downtime*, where downtime is the time from sending the request for c_t to fully receiving the chunk, a new DOWN event associated with a new chunk is generated if the buffer size is less than $B^{\max}$. The index and the quality level of the new chunk are decided by RLAGS or RLAS methods. Otherwise, a PAUSE event is generated if the buffer size exceeds $B^{\max}$.
- A PAUSE event simulates pausing the download on a path due to the buffer size exceeding $B^{\max}$. If a PAUSE event is encountered at time t, with timestamp $t + sample$, where sample is a short period (0.05 second in our program), a new PAUSE

[5] The source code of the environment is available at `https://github.com/ntnghia1908/Master_Thesis/blob/main/RLAS/menv_baseline.py`.

event is generated if the buffer size exceeds $B^{\max}$; otherwise, a new DOWN event associated with a new chunk is generated.
- A PLAY event occurs when the client starts playing a chunk, say chunk c_t. After a PLAY event, at time $t + chunk\_length$, a new PLAY event associated with chunk $c_t + 1$ is generated if this chunk is available in the buffer; otherwise, a REBUFFER event is generated.
- A REBUFFER event occurs when the chunk going to be played is not in the buffer. After a REBUFFER event, at time $t + sample$, a new PLAY event associated with chunk $c_t + 1$ is generated if this chunk is fully received; otherwise, a REBUFFER event is generated.

3.2. Simulation settings

We evaluate RLAGS and RLAS with Big Bug Bunny video [4]. There are seven quality levels $300, 700, 1200, 1500, 3000, 6000, 8000$ Kbps ($L = 7$). Assume that the maximum buffer size of the client is $B^{\max} = 30$ seconds and the video chunk length is 4 seconds. The number of chunks in the action space is $W = \lfloor 30/04 \rfloor = 7$, which means that if the agent is playing chunk i, the maximum chunk index stored in the buffer is $i + 7$. We train with the first 60 chunks, which results in 240 seconds per episode. Table 2 shows the parameters of the simulation environment.

Two real-trace datasets are used: a broadband dataset provided by US Federal Communications Commission (FCC) [22] and a 4G LTE Dataset collected from two major Irish mobile operators [24].

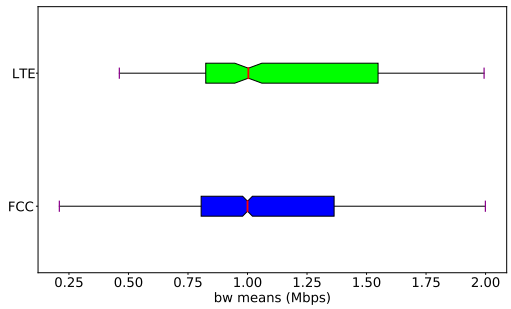


Fig. 3. The distributions of the means of all traces of the datasets.

The **FCC dataset** contains over one million throughput traces in the "download speed" category with a granularity of 10 seconds per sample [23]. (It is 5 seconds before 2016.) The **4G dataset** has 135 traces, with around 15 minutes per trace, at 1-second granularity. The traces are collected from Irish mobile operators with five mobility patterns: static, pedestrian, car, bus, and train [24].

Since the real bitrates of 8,000 Kbps quality level of the almost chunks are less than 4000 Kbps [4], we choose the traces with the average throughputs in $[0.1, 2.0]$ Mbps, in

which 1800 traces in the FCC dataset and 400 traces in the LTE one. Fig. 3 shows the distributions of the average throughputs of the traces from two datasets. We randomly select 80% of traces in each dataset for training, the remaining ones are for testing.

Table 2. Simulation parameters

Environment parameter	Notation	Value
maximum buffer size	$B^{\max}$	30 seconds
number of video chunks	N	60 chunks
number of quality levels in action space	L	7
number of chunks in action space	W	7
quality levels	l_i	$[300, 700, 1200, 1500, 3000, 6000, 8000]$ Kbps
utility	q_i	$\ln(\frac{l_i}{l_1})$
quality-switch coefficient	β	1
rebuffering coefficient	γ	3.3

We compare RLAGS and RLAS methods with two well-known adaptation methods, *i.e.*, throughput-based [2] and BOLA [8] (a buffer-based) methods. These adaptations are originally designed for single-source video streaming. We apply greedy scheduling to extend them to multi-source streaming. In the throughput-based method, the quality of the next download chunk on one path is the highest quality level which is smaller than the harmonic mean of the last six chunks downloaded on that path.

Table 3 lists some tuned hyper-parameters for RLAGS and RLAS. The not-listed hyperparameters are used with the default values provided by Stable Baseline3. We use fully connected neural networks with 64 nodes for each hidden layer. We tuned the number of hidden layers for the algorithms. Round-trip-times of the network connections are uniformly random in $[50, 100]$ ms.

Each proposed algorithm is trained in five runs, $30,000$ episodes each run. Each episode chooses a random trace in the training set and starts at a random point. The throughput trace is circulated if the time from starting point of an episode to the end of the throughput trace is not enough for the time playing the episode. The results are the average values in five runs.

3.3. Results

Fig. 4 shows the convergence of both RLAGS and RLAS algorithms in training with turned parameters given in Table 3. We can see that RLAGS converges faster than RLAS since RLAGS has fewer actions than RLAS, which are only quality levels. However, RLAS yields a higher average reward than RLAGS.

We test the case when one path is a broadband connection, and the other path is an LTE connection. The reward, utility, switch penalty, and rebuffering penalty are given

Table 3. Tuned hyperparameters used in RLAGS and RLAS.

Hyperparameters	Descriptions	Tuning ranges	RLAGS	RLAS
learning rate	learning rate	uniform: [0.0001, 0.001]	0.000125	7.61e-05
batch size	minibatch size	uniform: [59, 590]	411	530
n epochs	number of epoch when optimizing the surrogate loss	values: [10, 20, 30]	10	10
gamma	reward discount factor	values: [0.99, 1.0]	0.99	1
gae lambda	factor for trade-off of bias vs. variance for generalized advantage estimator	values: [0.9, 0.95]	0.9	0.95
clip range	clipping parameter	values: [0.2, 0.3]	0.3	0.2
vf coef	value function coefficient for the loss calculation	uniform: [0.2, 0.5]	0.317708	0.286954
ent coef	entropy coefficient for the loss calculation	values: [0.0, 0.00001, 0.00000001]	0	0
act func	value function coefficient for the loss calculation	values: [128, 256, 512]	256	512
features dim	value function coefficient for the loss calculation	values: [tanh, relu]	relu	tanh
policy net arch layer	number of policy network layer	values: [1, 2, 3, 4]	1	3
policy net arch units	policy network unit.	values: [64, 128, 256, 512]	512	256
value net arch layers	number of value network layer	values: [1, 2, 3, 4]	3	4
value net arch units	value network unit.	values: [64, 128, 256, 512]	512	256

in Table 4. The rewards yielded by RLAGS and RLAS are higher than the reward by throughput-based and BOLA methods. RLAS achieves the highest reward, and RLAGS results in a smaller rebuffering penalty.

We consider the performance of multisource streaming in the case when the difference between two paths increases gradually. Particularly, the mean bandwidth of the first path is from 1.5 Mbps to 2 Mbps and the mean bandwidth of the second path decreases gradually, in $[1.5, 2.0]$ Mbps, in $[1.0, 1.5]$ Mbps, in $0.5, 1.0$ Mbps, and less than 0.5 Mbps.

We can see from Fig. 5 that the rewards of multisource streaming of all the methods decrease gradually. The rewards of RLAS are the highest in most of the cases, which shows the efficiency of the RL-based chunk scheduling. BOLA yields a higher reward

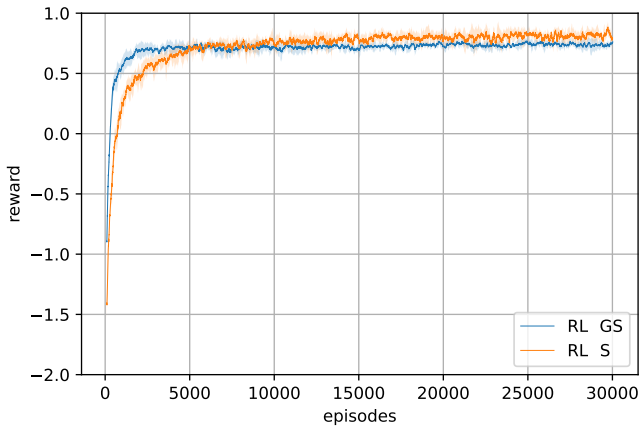


Fig. 4. Convergence of training phases of RLAGS and RLAS methods. The lines and shadows are the means and the standard deviations of the running average rewards of five runs, respectively.

Table 4. Rewards of ABR methods when one path is broadband and another path is LTE connection.

Methods	**reward**	**utility**	**switch penalty**	**rebuffering penalty**
THGHPUT	42.10	68.56	21.40	5.06
BOLA	77.80	129.75	27.26	24.70
RLAGS	88.35±2.04	97.86±2.29	8.53±1.01	0.98±0.52
RLAS	**107.75±1.91**	130.68±5.87	17.51±4.88	5.42±2.62

than RLAGS. However, in the extreme case when the mean bandwidth of two paths is very different, RLAGS and RLAS outperform the traditional methods.

Table 5. Rewards of ABR methods with the mean bandwidths of the first path is in $[1.5, 2.0]$ Mbps and of the second path is less than 0.5 Mbps.

Methods	reward	utility	switch penalty	rebuffering penalty
THGHPUT	17.34	68.20	32.03	18.83
BOLA	-35.78	120.11	19.10	136.78
RLAGS	**66.61±4.94**	92.68±2.66	9.74±1.92	16.33±7.69
RLAS	57.55±3.99	105.69±1.77	17.54±1.49	30.59±3.76

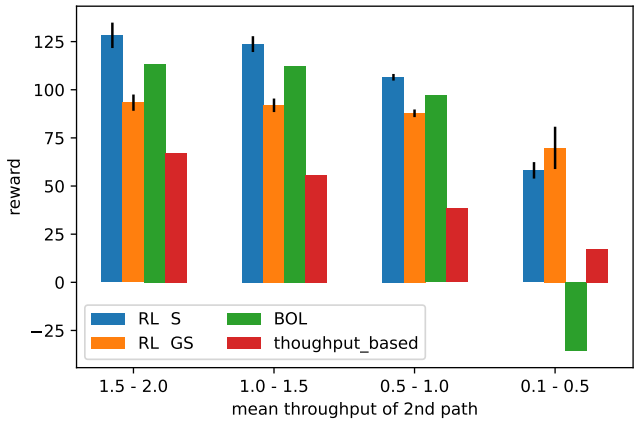


Fig. 5. The test rewards of different adaptation methods when the mean bandwidth of the first path is in $[1.5, 2]$ Mbps and the mean bandwidth of the second path decreases gradually.

Table. 5 shows the performance of the adaptation methods in the extreme case: the average throughput of the first path is in $[1.5, 2.0]$ Mbps, and of the second path is less than 0.5 Mbps. Overall, RLAGS and RLAS outperform BOLA method, and their rewards are much higher than those of the throughput-based method. The reward of RLAGS is a bit higher than RLAS because it has fewer actions in the action space. Hence, the agent may be easier to learn the optimum. The throughput-based method has the least rebuffering; however, it also has the lowest utility. BOLA has the highest utility but also the highest rebuffering penalty. RLAGS balances the objectives: high utility, low number of switches, and small rebuffering penalties.

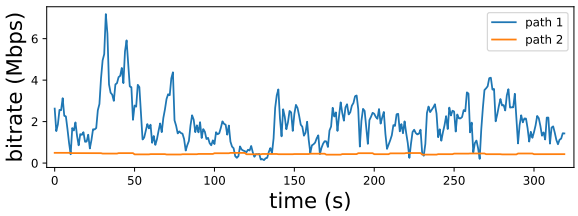


Fig. 6. A sample of throughput traces in the extreme case: the mean bandwidths of the first path is in $[1.5, 2.0]$ Mbps and of the second path is less than 0.5 Mbps.

Fig. 7 shows video quality levels selection and buffer occupancy when the client experiences a pair of throughput traces shown in Fig. 6 with different methods. The video

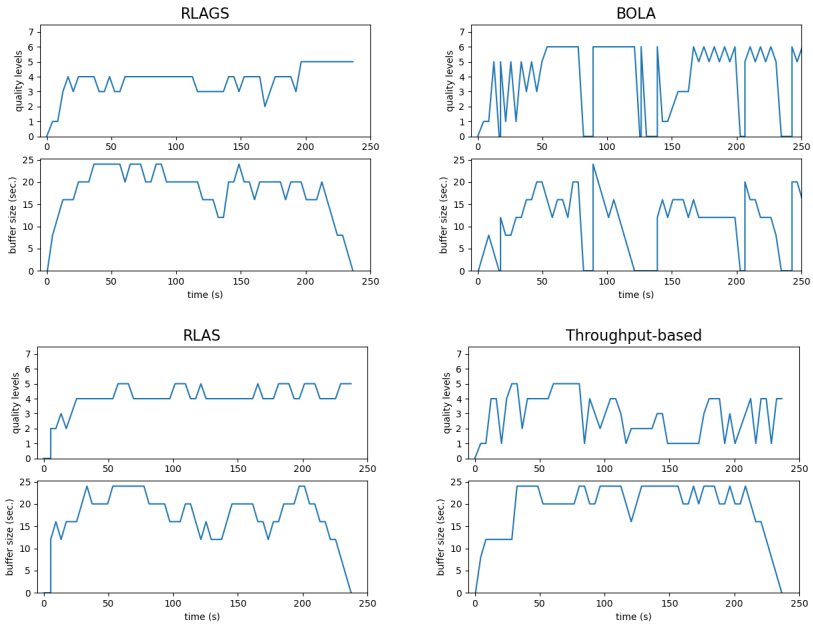


Fig. 7. The quality levels and the buffer occupancy with the sample throughput traces in Fig. 6.

played by the RL-based methods is more stable than by the other methods. We see that the proposed methods have a smarter buffer occupancy so that they can download higher quality levels with fewer switches than other methods.

4. Conclusions

We have proposed two novel adaptation and scheduling methods for video streaming from multiple sources, *i.e.*, RL-based adaptation and greedy scheduling (RLAGS) and RL-based adaptation and scheduling (RLAS). The state space, action space, and reward are defined for the methods. We have also built a GymAI-compatible environment for training and evaluation. Extensive simulations have shown that the proposed methods outperform the baseline methods in terms of the user's QoE. Model-free reinforcement learning algorithms could not work well in transfer learning [27]. If running the model in an untrained environment, the model could yield a low reward. In the future, we will apply model-based algorithms to bitrate adaptation.

References

1. Cisco: Cisco Visual Networking Index: Forecast and Methodology, 2016-2021.
2. T. Stockhammer: Dynamic adaptive streaming over HTTP: standards and design principles. In Proceedings of the second annual ACM conference on Multimedia systems, 133-144. (2011)

3. I. Sodagar: The MPEG-DASH Standard for Multimedia Streaming Over the Internet. IEEE MultiMedia, Vol. 18, Issue 4, 62-67. (2011)
4. S. Lederer, C. Müller and C. Timmerer: Dynamic Adaptive Streaming over HTTP Dataset. In Proceedings of the ACM Multimedia Systems Conference, 22-24. (2012) Online: `https://dash.itec.aau.at/dash-dataset/`.
5. ISO/IEC 23009-1:2014: Dynamic Adaptive Streaming over HTTP (DASH)– part 1: Media Description and Segments format.
6. DASH Reference Client. Accessed: Jun. 28, 2019. [Online]. Available: https://reference.dashif.org/dash.js/
7. J. Jiang, V. Sekar, and H. Zhang: Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE. In Proceedings of CoNEXT. (2012)
8. K. Spiteri, R. Urgaonkar, and R. K. Sitaraman: BOLA: Near-optimal bitrate adaptation for online videos. In Proceedings of 35th Annual IEEE International Conference on Computer Communications (INFOCOM). (2016)
9. T. Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson: A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In Proceedings of the 2014 ACM conference on SIGCOMM, 187-198. (2014)
10. Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran: Probe and adapt: Rate adaptation for HTTP video streaming at scale. IEEE Journal on Selected Areas in Communications, Vol. 32, No. 4, 719-733. (2014)
11. Y.C. Chen, D. Towsley, and R. Khalili: MSPlayer: Multisource and multi-path video streaming. IEEE Journal on Selected Areas in Communications, Vol.34, Issue 8, 2198-2206. (2016)
12. A. Nikravesh, Y. Guo, X. Zhu, F. Qian, and Z. M. Mao: MP-H2: a Client-only Multipath Solution for HTTP/2. In Proceedings of The 25th Annual International Conference on Mobile Computing and Networking, 1-16. (2019)
13. A. Bentaleb, P.K. Yadav, W.T. Ooi, and R. Zimmermann: DQ-DASH: A Queuing Theory Approach to Distributed Adaptive Video Streaming. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), Vol. 16, No. 1, 1-24. (2020)
14. M. Claeys, S. Latre, J. Famaey, and F. De Turck, "Design and evaluation of a self-learning HTTP adaptive video streaming client," *IEEE communications letters*, vol. 18, issue 4, pp. 716–719, 2014.
15. H. Mao, R. Netravali, and M. Alizadeh: Neural adaptive video streaming with pensieve. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication, 197-210. (2017)
16. M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella: D-DASH: A deep Q-learning framework for DASH video streaming. IEEE Transactions on Cognitive Communications and Networking, Vol. 3, Issue 4, 703-718. (2017)
17. D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley: Design, implementation and evaluation of congestion control for multipath TCP. In Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, Vol. 11, 8-8. (2011)
18. C. Raiciu, M. Handley, and D. Wischik: Coupled congestion control for multipath transport protocols, RFC6356. (2011)
19. Phuong Luu Vo, Tuan Anh Le, S. Lee, C. S. Hong, B. Kim, H. Song: mReno: a practical multipath congestion control for communication networks. Computing, Vol. 96, No. 3, 189-205. (2014)
20. Nghia T. Nguyen, Phuong L. Vo, Thi Thanh Sang Nguyen, Quan M. Le, Cuong T. Do, and Ngoc-Thanh Nguyen: A Reinforcement Learning Framework for Multi-source Adaptive Streaming. In Proceedings of International Conference on Computational Collective Intelligence, 416-426. (2021)

21. S. Huang and S. Ontanon: A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. In Proceedings of the Thirty-Fifth International Florida Artificial Intelligence Research Society Conference, (FLAIRS 2022), Florida, USA, May 15-18. (2022)
22. US Federal Communications Commission (FCC). [Online]. Available: https://data.fcc.gov/download/measuring-broadband-america/2019/data-raw-2019-sept.tar.gz
23. Tenth Measuring Broadband America Fixed Broadband Report [Online]. Available: Measuring Fixed Broadband - Tenth Report — Federal Communications Commission (fcc.gov)
24. D. Raca, J.J. Quinlan, A.H. Zahran, C.J. Sreenan: Beyond Throughput: a 4G LTE Dataset with Channel and Context Metrics. In Proceedings of ACM Multimedia Systems Conference (MMSys 2018), Amsterdam, The Netherlands. (2018)
25. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov: Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347. (2017)
26. A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus and N. Dormann: Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research, Vol. 22, No. 268, 1-8. (2021)
27. T. M. Moerland, J. Broekens, and C. M. Jonker: Model-based reinforcement learning: A survey. arXiv preprint arXiv:2006.16712. (2020)

Nghia Trung Nguyen received his bachelor's and master's degrees from Computer Science from International University - Vietnam National University Ho Chi Minh City in 2019 and 2022, respectively. His main research interest is to apply Reinforcement Learning to enhance the efficiency of various applications in computing.

Minh-Long Luu holds a bachelor's degree of Computer Science from International University - Vietnam National University Ho Chi Minh City. His main research areas are Reinforcement Learning and Computer Vision for bitrate adaptation, image classification, and out-of-distribution generalization.

Phuong L. Vo received her B.Eng and M.Eng degrees in electrical-electronics engineering from Ho Chi Minh City University of Technology, Vietnam in 1998, 2002, respectively, and Ph.D. degree at Kyung Hee University, Korea in 2014. Currently, she is an Associate Professor at School of Computer Science and Engineering at International University – VNUHCM. Her research interest is to apply machine learning, optimization, and game theory to contemporary networks.

Thi Thanh Sang Nguyen is a lecturer at the School of Computer Science and Engineering, International University, Vietnam National University, Hochiminh City, Vietnam. She has received her PhD degree in Software Engineering from the University of Technology, Sydney (UTS) in 2013. Her Ph.D. thesis is about Semantic-enhanced Web-page Recommender Systems. She was supervised by A.Prof. Haiyan Lu and Prof. Jie Lu. She received her master's degree in computer engineering from the University of Technology (VNU-HCMC) in 2006. She has more than 20 published research papers in the field of Web mining. Her research interests include Web mining, Semantic Web, knowledge discovery and business intelligence. Her profile on ResearchGate is https://www.researchgate.net/profile/Sang-Nguyen-7, and her publications are on https://dblp.org/pid/55/8981.html.

Cuong T. Do received his BS degree from Hanoi University of Science and Technology and Ph.D degree from Kyung Hee University, in electrical and computer engineering, in 2008 and 2014, respectively. His research interests include Queueing Theory, Game Theory, Machine Learning and their applications in Communication Networks.

Ngoc Thanh Nguyen (Senior Member, IEEE) is currently a Full Professor with the Wroclaw University of Science and Technology, and the Head of Information Systems Department, Faculty of Computer Science and Management. He is the author or coauthor of five monographs and more than 350 journal and conference papers. He has given 22 plenary and keynote speeches for international conferences, and more than 40 invited lectures in many countries. His research interests include collective intelligence, knowledge integration methods, inconsistent knowledge processing, and multi-agent systems.

Received: September 27, 2022; Accepted: November 20, 2022.

Computer Science and Information Systems 20(1):175–194 https://doi.org/10.2298/CSIS211110050J

Analyzing Feature Importance for a Predictive Undergraduate Student Dropout Model

Alberto Jiménez-Macias[1], Pedro Manuel Moreno-Marcos[1], Pedro J. Muñoz-Merino[1], Margarita Ortiz-Rojas[2, 3], Carlos Delgado Kloos[1]

[1] Universidad Carlos III de Madrid, Avda de la Universidad, 30
E-28911, Leganes, Spain
{albjimen, pemoreno, pedmume, cdk}@it.uc3m.es
[2] Escuela Superior Politécnica del Litoral, ESPOL, Campus Gustavo Galindo Km. 30.5
Vía Perimetral EC-090112, Guayaquil, Ecuador
maelorti@espol.edu.ec
[3] Ghent University, Henri Dunantlaan 2,
B-9000, Ghent, Belgium
margaritaelizabeth.ortizrojas@ugent.be

Abstract. Worldwide, one of the main concerns of universities is to reduce the dropout rate. Several initiatives have been taken to avoid this problem; however, it is essential to recognize at-risk students as early as possible. This article is an extension of a previous study that proposed a predictive model to identify students at risk of dropout from the beginning of their university degree. The new contribution is the analysis of the feature importance for dropout segmented by faculty, degree program, and semester in the different predictive models. In addition, we propose a dropout model based on faculty characteristics to try to infer the dropout based on faculty features. We used data of 30,576 students enrolled in a Higher Education Institution ranging from years 2000 to 2020. The findings indicate that the variables related to Grade Point Average(GPA), socioeconomic factor, and a pass rate of courses taken have a more significant impact on the model, regardless of the semester, faculty, or program. Additionally, we found a significant difference in the predictive power between Science, Technology, Engineering, and Mathematics (STEM) and humanistic programs.

Keywords: dropout model, features importance, data mining, learning analytics.

1. Introduction

One of the main issues Higher Education Institutions (HEIs) often face is the high rates of students' dropout [41]. For example, Scheneider [37] reported that about 30% of students in the USA drop out in the first year, and the World Bank reported a dropout rate of about 22% in Latin America [27]. In addition, Schnepf [38] reported dropout rates between 16% and 33% in Europe, although lower rates (11% and 4%) were reported in Asian countries, like Korea and Japan. As high dropout rates can be found in most parts of the world, it is very relevant to analyze how this issue can be detected and how these rates can be decreased.

In order to solve this issue of the dropout rates, it is possible to collect data from students and analyze it using learning analytics. As universities store dozens of records about students, such as students' grades throughout their degree programs, and know whether students completed their studies or dropped out, this information can help anticipate dropout cases and reduce this problem. Mainly, it is possible to detect students at risk and develop predictive models to forecast possible dropouts [8, 39]. Early detection of dropout can be beneficial since this may allow carrying out interventions to address this issue. Some possible interventions can be offering an orientation to students (e.g., counseling sessions to guide students in the courses they should take) [42], offering financial aids or scholarships to students with economic issues [12], and so on. Moreover, apart from specific interventions designed for students, analyses can also provide insights about possible difficulties in how the degree program is organized (e.g., if the workload of a specific semester is unbalanced, that could cause dropout).

In addition, to provide proper and timely support to the students, it is also essential to detect the main factors behind the predictive models. For example, Del Bonifro et al. [15] identified that the number of credits acquired by the students was a significant factor for dropout. However, that number was not available at application time, and thus variables related to performance in high school had to be used. In addition, Abu-Oda and El-Halees [1] discovered that some courses might have a more significant influence on dropout (e.g., students who got a high grade in a specific course had a lower probability of dropout).

In this line, this work aims to conduct a study using several degree programs in an HEI in Ecuador to discover more important variables in students' dropout. This paper aims to address the following main research question: What variables significantly influence students' dropout? To analyze this question, the objectives of this paper are as follows:

Analyze the feature importance for a student dropout predictive model considering all degrees

1. Analyze the feature importance for a student dropout predictive model considering the semesters throughout a degree
2. Analyze the feature importance for a student dropout predictive model considering the faculty level
3. Propose a predictive model based on a model of the University's faculties characteristics to estimate the dropout

This paper is an extension of [20]. In [20], we presented an algorithm for early dropout prediction and resulted in the algorithm's prediction power in different degrees. This paper extends the analysis using feature importance for the different variables involved in the prediction. We aimed to determine the most critical variables and if the differences depend on time, faculty, or degree.

The article is structured as follows. Section 2 presents an overview of the relevant literature. Section 3 describes the dataset and how the predictive model is designed. Section 4 shows the results and discussion, and Section 5 draws conclusions from this analysis and suggests possible directions for future work in this area.

2. Related Work

There have been many contributions focused on dropout prediction. These contributions have been made at two different levels. Some have been done at the course level (i.e., predict who will drop out of a course), while others have been done at the degree program level (i.e., predict who will drop out a degree). Among the former group, there has been research on Massive Open Online Courses (MOOCs), where Moreno-Marcos et al. [29] made a review on prediction in MOOCs and found many different variables were relevant to predict students' performance, including variables related to self-regulated learning, interactions with videos and exercises in the platform. In addition, research has been done in university courses different from MOOCs. For example, Burgos et al. [10] predicted dropout in university courses through Moodle data, and they claimed their models helped reduce dropout by 14%. Moreover, Pereira et al. [34] predicted dropout as a lack of attendance in programming courses.

On the other hand, several works have focused on dropout detection at the degree program level. For example, Luo and Pardos [24] used data from 8 years of course enrollments to predict whether the students would graduate on time. Furthermore, Chen et al. [13] predicted dropout in nine different majors (mostly STEM majors) and found that survival analysis approaches could achieve promising results. When developing these models, one crucial aspect is the anticipation because if models are only accurate at the end of the course, they may not be effective. In this line, Márquez-Vera et al. [28] predicted whether students would continue studying in the following academic year using data from 419 Mexican students and found accurate results within the first 4-6 weeks of the course. Furthermore, Jiménez et al. [22] emphasized the importance of early predictions, and they optimized models to obtain reasonable accuracies of dropout prediction in the university programs after two semesters.

Apart from the anticipation, one of the key aspects to make these predictions are the predictor's variables used to generate the models. In this case, variables are often retrieved from the academic record (e.g., Student Information System). One typical variable is the GPA, combined with other grades. For example, Kang and Wang [23] included both the overall GPA and the term GPA to predict dropout, combined with other variables such as gender, ethnicity, time status (e.g., full-time, half-time), classification (e.g., freshman, sophomore.), and age. They found that while GPA is strongly associated with dropout, other variables could also achieve strong results.

Moreover, Ameri et al. [5] combined several features, including demographics, family background (e.g., parents' educational level), pre-enrollment attributes (e.g., high-school GPA and grades from the admission test), financial attributes, enrollment attributes, and academic attributes. They used GPA, the percentage of passed, dropped, and failed credits, and the credit hours attempts among those academic grades. They concluded that the variables with the highest impact were the high school GPA, the GPA, the percentage of failed credits, and the financial attributes.

Another relevant issue when designing a predictive model is to select the prediction algorithm. For example, Aulck et al. [6] used logistic regression, Random Forest (RF), and k-NN to predict dropout and found better predictors with logistic regression. In addition, Barbosa Manhães et al. [7] used several machine learning algorithms to predict dropout in several undergraduate STEM degree programs in a Brazilian university. They found that multilayered perceptron, logistic regression, Support Vector Machines

(SVMs), and RF were more accurate. Furthermore, Ortigosa et al. [32] applied an early student-dropout prevention system and used it in production. They suggested that tree-based methods, such as RF, can outperform other models (as in [21-31]), but they pointed out that the explicability of the models was essential when they were put in production. Because of that, they initially used RF in a lab environment, but they preferred using the C5.0 decision tree model in the production stage. For that stage, another relevant issue is the generalizability of the models, i.e., ensuring that a model trained with some students is valid for other students.

While some authors have tried to mitigate this issue of generalizability with machine learning techniques, such as assembling [14], it is not feasible to get a one-size-fits-all model, and the instructional conditions should be considered [18]. For example, significant differences can be found when predicting using models trained with different courses or students [30]. Because of that, it is essential to develop separate models for each degree program to keep the context as similar as possible and consider that models may not generalize over time (e.g., when the study plan changes). Moreover, it is vital to analyze the differences of the predictive models depending on the context (e.g., degree program or faculty).

In this context, this work aims to analyze the importance of variables when predicting dropout in different degree programs of a Higher Education Institution. This will better understand how the prominent variables in the models generalize or differ when changing the context. Mainly, the analysis contributes to understanding the importance of variables over time and across degree programs and faculties. Moreover, it provides insights into the essential variables when designing predictive models so that other researchers can adapt predictive systems in their institutions more efficiently.

3. Early Dropout Prediction Model

This section describes the dropout prediction model. The dataset, preprocessing process, and used algorithms are described. Figure 1 from our previous work [20] describes the predictive model; in the transformation phase, we cleaned the provided data. Then, we calculated additional indicators to be used as input in the model. After that, we removed the degree programs that had no graduating students or dropped out because these are new degrees at the university. Next, we calculated the indicators related to the academic history of the students in the selected degrees. Finally, we split the data set, leaving 80% for training and 20% for testing, ran the predictive model on the data, and evaluated the results obtained with the selected metrics.

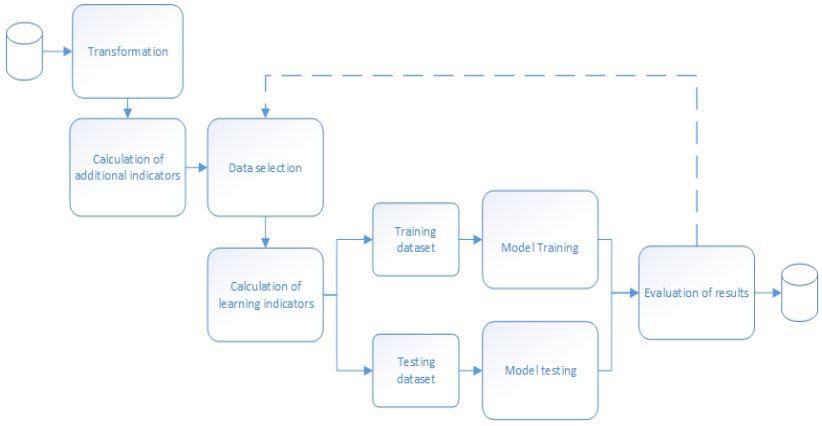


Fig. 1. Predictive model phases [20]

3.1. Dataset and pre-processing

The dataset used for this analysis contains demographic and academic data provided by the university's Information Technology department. This dataset contains academic records of 30,576 students enrolled from 2000 to the first semester of 2020 in 25-degree programs in 8 faculties. Table 1 shows the number of students in the degree programs with the largest number of students, and Table 2 shows the five faculties with the highest number of active students.

The data used as predictors included the following categories: (1) socio-demographic information (for example, employment status, city of residence, marital status, school of origin), (2) financial information (socioeconomic factor), (3) information on study program (e.g., number of credits to complete, course code) and (4) academic performance data (e.g., courses taken, courses ratio, course credits, among others).

During the pre-processing stage, cleaning, changing, and unifying techniques were performed due to the data structure. For the socioeconomic factor variable, we scaled the different student values to a single scale between 0 and 1, where 0 indicates a low socioeconomic level and 1 indicates a high socioeconomic level.

For this study, the academic semesters are identified according to the academic calendar. Particularly, the first and second semesters were called "ordinary semesters" and they normally started in May and October respectively. In addition, there is a third semester, called the "extraordinary semester", which comprises a short period of two months, generally in March and April of each year and this semester is usually used by students for two reasons: to pass courses previously failed or pending registration because the courses do not give flow to others. Students can take a maximum of two courses during the extraordinary semester, although they can decide whether to take them or not. Only ordinary semesters were considered for the predictive models because they had a similar duration of in-class weeks. The students' semesters were sorted chronologically to calculate the variables from each student's first semester in each course.

Table 1. Degree programs with the most active students

Degree Program code	Degree Program name	Faculty	Number of students
CI007	Mechanics	Faculty of Engineering in Mechanics and Production Sciences	792
CI005	Civil Engineering	Faculty of Engineering in Earth Sciences	743
CI013	Computing	Faculty of Engineering in Electricity and Computing	716
LI002	Economy	Faculty of Social and Humanistic Sciences	656
LI007	Business Administration	Faculty of Social and Humanistic Sciences	546
CI001	Industrial Engineering	Faculty of Engineering in Mechanics and Production Sciences	532
CI002	Chemical Engineering	Faculty of Natural Sciences and Mathematics	470
LI006	Production for Media	Faculty of Art, Design and Audiovisual Communication	469
LI005	Graphic Design	Faculty of Art, Design and Audiovisual Communication	457

Table 2. Faculty with most active students

Faculty name	Faculty	Number of students
FIEC	Faculty of Engineering in Electricity and Computing	2582
FCSH	Faculty of Social and Humanistic Sciences	2421
FIMCP	Faculty of Engineering in Mechanics and Production Sciences	2142
FCNM	Faculty of Natural Sciences and Mathematics	1184
FICT	Faculty of Engineering in Earth Sciences	1132
FADCOM	Faculty of Art, Design and Audiovisual Communication	995
FCV	Faculty of Life Sciences	623
FIMCM	Faculty of Maritime Engineering and Marine Sciences	536

After pre-processing, variables related to academic performance were precalculated to be used in the models. The complete list of variables is presented in Table 3. The variable V1 represents the socioeconomic level of the student and family of the student, V2 indicates the total number of times the student enrolled for the second time in a course after having failed it on a previous occasion in his or her current program, V3 indicates the total number of times the student enrolled for the third time in a course after having failed it twice on a previous occasion in his or her current program, V4 indicates the total number of years in which the student has not taken courses, V5 indicates the average of the grades in courses passed and failed registered in the current academic year, excluding cancelled courses, V6 indicates the weighted average of the grades in courses taken considering a penalty according to the number of times taken the same course, V7 indicates the proportion of courses passed by the student in the current undergraduate program, V8 indicates the proportion of courses failed by the student in the current undergraduate program, V9 indicates the proportion of courses with canceled enrollment by the student in the current undergraduate program. Some available sociodemographic variables, such as residence, school of origin, marital status, and employment status, were discarded due to the low correlation between the model output variable. University GPA is calculated by obtaining the average of the final grades among all the courses taken by the student. The final grade for each course is calculated by averaging the two highest grades of the three midterm grades.

Table 3. Learning indicators for the model

Variable ID	Variable	Category	Description	Values
V1	Economic factor	Financial information	Socio-economic factor	0 to 1
V2	Seg mat	Study program	Number of times of second enrollment in a course after failing it the first time	0 to a max number of courses
V3	Ter mat	Study program	Number of times of third enrollment in a course after failing it the second time	0 to a max number of courses
V4	Gap year	Study program	The period in years that the student takes to return to study	0 to maximum not defined (increase in intervals of 0.5 each semester without enrollment)
V5	Average APRP	Academic performance data	GPA of taken courses in the current undergraduate program	0 to maximum possible score (taken courses in other programs are excluded)
V6	Average weighted	Academic performance data	GPA of courses with a number of credits greater than zero, considering a penalty depending on the number of times the student takes the same course	0 to maximum possible score (Penalty: 0,9 for a second time and 0,8 for a third time)
V7	Ratio pass	Academic performance data	The ratio of passed courses by the student	0 to 1
V8	Ratio fail	Academic performance data	The ratio of failed courses by the student	0 to 1
V9	Ratio cancel	Academic performance data	The ratio of the canceled courses by the student	0 to 1

The predicting variable is whether or not a student will drop out in a degree, i.e., it is a categorical variable with only two possible values (0 for dropout and 1 for completion). We establish the dropout criterion when an undergraduate student has not taken any course for quite some time since their last enrollment. Considering the internal rules and guidelines of the university, this period is five years. Therefore, students who had not enrolled for more than five years were detected as dropouts.

In addition, undergraduate students who completed 90% of their college degrees were considered non-dropouts. This rate has also appeared in previous works [3], which showed that individuals with this high level of completion dropout of their college degrees for reasons unrelated to their academic performance.

3.2. Dropout algorithm for each degree

As for the machine learning algorithm, Random Forest classifier (RFC) computation was used [9], and using the RandomForestClassifier method of the sklearn library within the ensemble methods the GridSearchCV method to find the best parameters using Python as the programming language. We evaluated by purchasing the performance using the Area Under the Curve (AUC) metric, similar to [19,40 ,17]. RFC is a remarkable tree-based learning computation known for its low overprediction bias and high accuracy [9].

After pre-processing, the dropout model was trained for each degree program segmented by each semester and faculty meaning that the model used different input data for training depending on the semester or faculty as appropriate. In addition, specific data of students who had already graduated or dropped out were used to test the model. Subsequently, the model was run using active students in each of the first five semesters; for example, the first-semester degree program model was used for all students who had completed a semester in their degrees. The second model was used with students who had completed two semesters, and so on. After the fifth semester, all students who had completed more than five semesters in their undergraduate programs were grouped into a single model, including their interactions. As indicated in [16], regular school dropout occurs between the second and third years of Ecuador studies, and that is why the model focus on the first semesters.

3.3. Dropout algorithm for each Faculty

Using the dataset described in section 3.1, we propose a characterization model for the faculties using student interactions. The purpose of the model is to obtain the following characteristics: *average_weighted*, *ratio_pass*, *type_faculty* as described in Table 4. Table 2 describes the university faculties that are part of the study; the model learned based on the students who have studied in the eight faculties using as inputs the three variables described in Table 4.

Table 4. Learning indicators of faculty for the model

Variable ID	Variable	Description	Values	Type
V1	Average weighted	GPA of courses with a number of credits greater than zero, considering a penalty depending on the number of times the student takes the same course taken in the faculty's programs	0 to max (maximum possible is 10)	Output
V2	Ratio pass	The proportion of courses passed in the faculty	0 to 1	Output
V3	Type faculty	Indicates the type of faculty between STEM and NO_STEM	0 = No_STEM or 1 = STEM	Output

Along with the characteristics model described in the previous paragraph, we propose a global dropout model using as a focus the faculty that is the most global level within the university, a similar perspective to the model proposed in [25] where the author uses global and local data for each degree to calculate dropout. We use the output of the faculty characteristics model described in Table 4 as input to the dropout model and obtain as model output the probability of dropout in the faculties using Random Forest Classifier (RFC) as the algorithm. The model learns based on the faculties described in Table 2, obtaining eight rows to learn what is dropout and eight rows to learn what is not dropout.

4. Results and Discussion

4.1. Prediction Accuracy of the Model

In the first analysis, we averaged the values of each model's metrics using data from ten-degree programs. The results indicated that the different models could accurately predict dropout across all degree programs. Table 5 shows the results for ten randomly selected degree programs out of the 25-degree programs where the model was run, confirming the results as mentioned earlier. Using the AUC metric, the lowest results were obtained for CI004 with 0.80 and CI005 with 0.76. Although the results are not as good compared to the others of this study, there are semesters in both degree programs where the AUC metric is above average. In general, the model obtains good results for the metrics analyzed based on [29], indicating that obtaining an AUC is suitable between 0.8 and 0.9 inclusive. Our model can predict both a student who has completed a semester and a mid-career, similar to the levels obtained in [25,36] above 80%. student analyzed. Therefore, the design of proposed predictive model could be replicated using the same algorithm and variables in other higher education institutions with similar conditions to the one used in the present research.

Table 5 Prediction accuracy for the undergraduate programs between 2000 and first semester 2020

Code of Degree Programs	Degree	AUC average
CI003	Mining Engineering	0.87
CI004	Petroleum Engineering	0.80
CI005	Civil Engineering	0.76
CI008	Statistical Engineering	0.99
CI009	Logistics and Transportation Engineering	0.99
CI013	Computer Engineering	0.99
CI018	Telematics Engineering	0.98
ECCBA	Economy	0.98
INACP	Auditing and Certified Public Accountancy Engineering	0.99
INALL	Food Engineering	0.98

4.2. Feature Importance

This section shows the results obtained after analyzing the feature importance for the different degree programs in the trained model in the different semesters and faculties. The weight value of each variable was obtained through the Mean Decrease metric in Gini coefficient using the Python Shap library. Table 6 presents the frequency of use and the average weight of the variables. All values are used in at least one model within the degree programs. The variables that are used in all models are *average_aprp, rate_pass, economic_factor, average_weighted*.

The results show that the two most important variables in all models are: *average_aprp* and rate_pass. *Economic_factor* is the third most important model, confirming studies considering that this variable influences student dropout [11]; this result can be inferred from the country's economic indicators. The variable *average_weighted* has an average weight of 0.16 due to the relationship with the variable *average_aprp*. Both use the student grades in the different courses taken. The variables *seg_mat* and *ter_mat* have an average weight of 0.05 and 0.07, respectively. The *ratio_cancel* variable has an average weight of 0.02 in the model because students prefer to continue in the course until the end despite failing or dropping out after the first evaluation but without taking the corresponding administrative steps to cancel it. The *rate_fail* variable is only used in 1.1% of all the models (in two models) due to its high relationship with the *rate_pass* variable; it has high importance in the few models used. In conclusion, the ratio variables are correlated among the three, and consequently the low frequency of use of the ratio_cancel and ratio_fail variables. For future research, we recommend using the first four variables whenever possible because of their high predictive power, which was also supported in [2,11].

Table 6. General results of features importance

Variable ID	Variable	Frequency of use	Average weight
V5	Average APRP	100%	0.28
V7	Ratio pass	100%	0.27
V1	Economic factor	100%	0.16
V6	Average weighted	100%	0.16
V2	Seg mat	87.91%	0.07
V4	Gap year	80.22%	0.02
V3	Ter mat	75.82%	0.05
V9	Ratio cancel	39.56%	0.02
V8	Ratio fail	1.1%	0.19

Feature Importance per semester of the models for all the degree programs

To understand the behavior of the models, we averaged the values of the importance of the features in the models for each semester. Table 7 shows the average weights of the models for all the degree programs segmented by semester, the sum of the importance of all the characteristics is one for each semester. The four variables with the highest weights in the predictive model are: *average_aprp, rate_pass, economic_factor, average_weighted* in the first year. The variables *seg_mat, gap_year and ter_mat*, have no value initially because they only have information from one semester, and it is not possible to repeat a course or have years without studying. The *economic_factor* variable loses importance as the semesters of study increase; this could be due to the student's effort to finish his studies despite the economic problems that may occur as he progresses. The variables *average_aprp and ratio_pass* have a constant behavior during the models in the different semesters, due to the significant difference in these variables between graduates and dropouts. On the other hand, the variable *term_mat* has slightly increased weight from the third semester because a student may be taking his third enrollment in some course. The variables *gap_year and ratio_cancel* have little

significant weight in the models, and the variable *ratio_fail* is used only in two models in the model of 4 and 6 or more semesters but had low weight.

As the student progresses in his degree, other variables are included as necessary in the model. For example, when the student is in six semesters or more, the *rate_fail* variable has a significant weight in the model, while the *socioeconomic_factor* variable has a low importance weight. The findings show a correlation between grades and pass rate variables, students in their first semester try to obtain a good grade point average in the courses registered. However, as they advance in their degree program, it is more important to pass the course without having so much weight on the grade obtained in the courses.

Table 7. Results of the average weight of features importance per semester

Semester	Average APRP	Ratio pass	Economic factor	Average weighted	Seg mat	Gap year	Ter mat	Ratio cancel	Ratio fail
1	0.33	0.23	0.24	0.20	0.01	-	-	-	
2	0.29	0.27	0.17	0.17	0.08	0.01	0.01	0.02	-
3	0.27	0.26	0.16	0.16	0.07	0.02	0.05	0.02	-
4	0.26	0.26	0.15	0.15	0.07	0.03	0.06	0.01	0.01
5	0.27	0.27	0.14	0.15	0.08	0.03	0.06	0.01	-
6 or more	0.27	0.34	0.08	0.12	0.08	0.03	0.05	0.02	0.02

Feature Importance per faculty

The behavior between the faculties could be different between the variables in the models. To clarify this hypothesis using data from all semesters, Table 8 shows the average weights of the importance of faculty characteristics. In the faculties shown, the four most important variables for the models are *average_aprp*, *ratio_pass*, *economic_factor*, and *average_weighted*. The variables *average_aprp*, *ratio_pass*, and *average_weighted* have the same trend between each faculty model. Therefore, student performance is an essential variable in the dropout model regardless of the student's faculty. In addition, the *economic_factor* variable presents two groups in its behavior. One is associated with the STEM degree programs (FIEC, FIMCP, FICT) with high values.

The other is related to the humanistic degree programs (FICSH, FCNM, FCV, etc. FIMCM, FADCOM) with low values because students of humanistic degrees usually have a higher economic level. The seg_mat variable is more important in the FCV and FIMCM faculties since they are relatively new and students leave semesters without studying. However, it is not a significant value concerning the rest, but shows a difference with the other faculties. The variable *ter_mat* has a lower weight in the FIMCP and FCV faculties, while the variable *gap_year* has a lower weight in FIEC, FICMP and FIMCM. In the FICSH, FICT, and FADCOM faculties, the variable *ratio_cancel* has a very low weight, almost zero concerning the other variables; we could infer that the students rarely canceled the courses in these faculties. Finally, in all the models, the variable ratio_fail does not have a significant weight. The models trained by faculty showed the same results based on the weight of the four variables compared to the models segmented by semesters. For students of all faculties, STEM or no-STEM

is essential to pass the courses with a good grade. A gender is a variable that does not affect student dropout for this university, however Pilotti et al. [35] found that gender does affect student performance for STEM and non-STEM.

Table 8. Results of the average weight of features importance per faculty.

Faculty name	Average APRP	Ratio pass	Economic factor	Average weighted	Seg mat	Gap year	Ter mat	Ratio cancel	Ratio fail
FIEC	0.245	0.250	0.242	0.137	0.066	0.019	0.057	0.015	-
FICSH	0.261	0.280	0.113	0.165	0.058	0.032	0.073	0.06	-
FIMCP	0.278	0.224	0.216	0.156	0.069	0.017	0.039	0.023	-
FCNM	0.250	0.260	0.180	0.159	0.062	0.032	0.066	0.022	-
FICT	0.285	0.260	0.230	0.127	0.050	0.020	0.045	0.008	-
FADCOM	0.299	0.308	0.135	0.160	0.057	0.026	0.029	0.009	-
FCV	0.252	0.224	0.173	0.162	0.093	0.035	0.102	0.025	-
FIMCM	0.301	0.253	0.172	0.141	0.081	0.016	0.053	0.017	-

We selected four faculties from Table 2 with more students, segmented into two groups: STEM degrees and humanistic degrees. Figure 2 shows a comparison between the feature importance of the FIEC, FICMP faculties corresponding to STEM (FIEC and FIMCMP) and the faculties FCSH, FADCOM corresponding to humanistic degrees (FCSH, FADCOM) faculties. The results showed differences in behavior in some variables; for example, *economic_factor* in STEM schools' degrees had a greater weight than in humanities because the socioeconomic level in humanities faculties was usually higher than in STEM. The variable *ratio_pass* had a greater weight in humanistic faculties to STEM; this indicated that students could pass more courses in humanistic degrees. It should be clarified that the educational contents of the courses are different for the STEM and humanities faculties during the basic formation in degrees. For example, the courses: Calculus, Statistics, Linear Algebra, Programming has different content and evaluation forms for each of the faculties. In the year 2020, changes were made in the other degrees' curricular to unify these courses. All students can see the same content regardless of the faculty during their fundamental training.

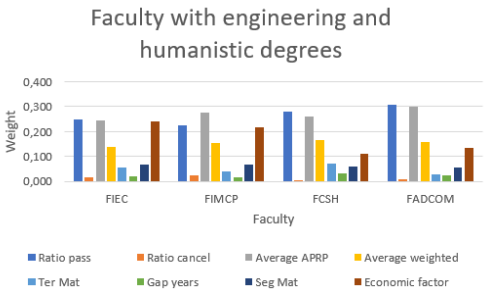


Fig.2. Feature importance between faculty STEM and humanistic degrees**.**

Among the findings, after analyzing the features importance for the different faculties, we found similar behavior in most of them. Thus, we decided to explore feature importance in some degree programs of the faculties to understand the behavior of these variables and identify if the same pattern of the faculties is found.

Feature Importance per degree program

Models predicting better results using characteristics for each degree than global characteristics were similar to [25,36]. We have used this local approach with attributes of each degree, obtaining similar results. In addition, we have performed an analysis in the ten-degree programs with the highest number of active students presented in Table 1, trying to understand the variables' behavior in the different degree programs. Table 9 shows the average weight of the variables in each of the degree programs. The four most important variables for these degree programs were *average_aprp*, *ratio_pass*, *economic_factor*, and *average_weighted*; This finding confirms the faculties' analysis results. The variable *average_aprp* had the lowest weight in the CI002 degree program compared to other programs where the behavior is similar. The variable *ratio_pass* had the lowest weight in the CI001. The *economic_factor* variable had a similar behavior for degrees: CI007 (STEM degrees), LI002, LI007, LI006, LI005 (humanistic degrees), and other behavior for degrees: CI005, CI013, CI001, CI017 (STEM degrees), LI006 (humanistic degree). The LI007 degree program presented a high value in weight for the *average_weighted* variable, significantly different from the other degree programs.

The variable *seg_mat h*ad the lowest value in the degree program CI005 and LI006. The variables *gap_year* and *ter_mat* presented the highest in the CI007 degree program; despite being small, they presented a more significant difference. The variable *ratio_fail* showed little significant weight for degree programs CI007 and LI005. In the faculties analysis, we found that the faculties with a low weight for the *ratio_fail* variable were FICSH, FICT, and FADCOM; the CI007 degree program belongs to the FIMCP faculty; therefore, if we only carried out an analysis by faculties, we could not find this type of findings. Finally, the variable *ratio_fail* had no importance for any model of the ten-degree programs shown. The results allow us to understand better the behavior of the variables in each degree program. These findings cannot be found in a general way if we only analyze the faculties due to the context of each degree program with different behavior on the part of the students.

Table 9. The average weight of feature importance per degree.

Degree	Average APRP	Ratio pass	Economic factor	Average weighted	Seg mat	Gap year	Ter mat	Ratio cancel	Ratio fail
CI007	0.285	0.268	0.177	0.140	0.073	0.064	0.151	0.008	-
CI005	0.258	0.224	0.259	0.148	0.051	0.015	0.055	0.016	-
CI013	0.221	0.265	0.259	0.122	0.073	0.015	0.066	0.020	-
LI002	0.287	0.284	0.134	0.133	0.065	0.024	0.085	0.010	-
LI007	0.253	0.219	0.159	0.217	0.087	0.037	0.052	0.011	-
CI001	0.226	0.177	0.318	0.166	0.076	0.018	0.035	0.014	-
CI002	0.196	0.206	0.320	0.150	0.060	0.012	0.041	0.025	-
LI006	0.278	0.240	0.204	0.163	0.054	0.026	0.040	0.027	-
LI005	0.265	0.299	0.188	0.149	0.063	0.027	0.028	0.005	-
CI017	0.238	0.213	0.287	0.127	0.070	0.014	0.053	0.024	-

We chose the most active students to identify patterns or differences between a STEM degree and a humanistic degree. Figure 3 shows the behavior of the variables in the CI007 degree models as it is the degree with the most active STEM students. The obtained grades and the passing rate were essential variables in all the semesters. The *socioeconomic_factor* was important in the model at the beginning of the degree. Still,

when the student passed the middle of the degree, it had little relevance in the model due to the students' effort to finish the degree. The variable *seg_mat* was important in the second-semester model; we could infer that in the second semester of the degree, students repeated courses, while the weight of the variable *ter_mat* from the third semester maintained a significantly low value indicating that few students reached the third enrollment.

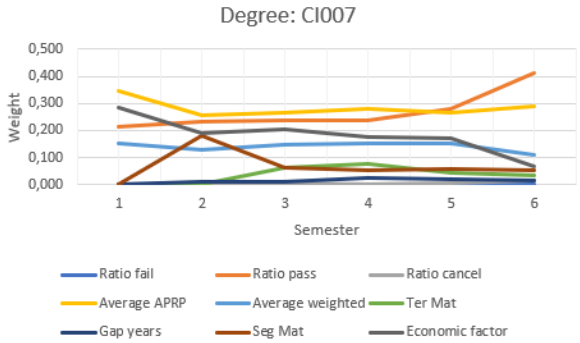


Fig.3. Weight of the variables segmented by the semester of the CI007

Figure 4 shows the behavior of the variables in the models for the LI002 degree because it is the degree with the most active students in the humanistic degree. The two most important variables in the degree model were *average_aprp* and *rate_pass*, exchanging importance after the second year (four semesters) of studies. These variables have a large gap concerning the other variables, with an average weight of less than 0.15 during each semester's different models. In the first semester, the economic level of the student influences the degree of dropout; as the student advances in the levels of the study program, the weight of this variable tends to decrease. The variables *seg_mat*, *gap_year*, and *ter_mat* were important in the first semester because there were only data from one semester studied. The *seg_mat* variable was important from the second semester. There is already data with students studying a course for the second time. Also, with the variable *ter_mat* from the third semester, students were already taking courses for the third time. Between the third and fifth semesters, more students took courses for the third time than for the second time. Finally, the variables *ratio_fail*, *gap_years*, and *ratio_cancel* were barely significant for the degree.

The first difference that we can observe in the analysis of degrees concerning faculties is the *average_arprp* variable. This variable is the most important in the first two years of both degrees, but it is the fourth important variable in the faculties analysis models. The variables *ter_mat* and *seg_mat* have similar behavior to the STEM degree. In contrast, to the humanistic degree, the variable *ter_mat* has a higher weight than *seg_mat*. In this case, humanistic degree students take more courses for the third time than STEM students. While during the middle stage of their degree, the variables have almost similar behavior in both cases.

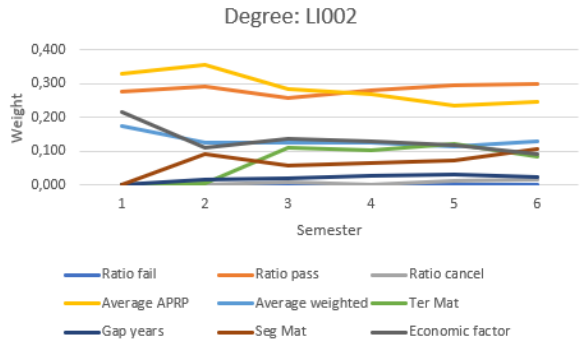


Fig.4. Weight of the variables segmented by the semester of the LI002

4.3. Model of Faculty

Using the model of faculty characteristics described in Section 3.3, we calculated the three variables listed as results in Table 4 for the faculties using the students currently associated with the respective faculties as input. Table 10 shows the results obtained using the proposed model of characteristics. According to the results obtained, the eight faculties can be grouped into 3 clusters as they have similar results for the characteristics. The proposed model has a limitation, so some metrics could not evaluate it [33] because the data used to learn the model are only from 8 faculties. Therefore, we would need data from more faculties from other universities to measure the algorithm's robustness or perform data simulations.

Table 10. Results of model faculty

Faculty ID	Average weighted	Rate pass	Faculty type
F1	7.13	0.6	NO STEM
F2	7.7	0.89	STEM
F3	7.13	0.6	NO STEM
F4	7.13	0.6	NO STEM
F5	7.84	0.92	NO STEM
F6	7.7	0.89	STEM
F7	7.7	0.89	STEM
F8	7.7	0.89	STEM

Using the data described in Table 10, we run the proposed dropout model. Table 11 shows the result obtained using the model for the faculties; we calculated the dropout probability using two different data sets: academic data for the last 20 years and academic data for the last ten years. To find a difference between the behavior of graduates and dropouts in the faculties, we shorten the range of years by dividing the input data used in the model into two groups.

The results show that there is a significant difference between the dropout measurement using as a data set the last 20 and 10 years in Faculties 1,3,4,5 because the students who are currently students have a weighted average and a pass rate closer to the

students who graduated in the last ten years concerning those who graduated in the last 20 years. On the other hand, in Faculties 2,6,7,8, there is a contracting effect in which the increase in dropouts is because the data used as input are nearer to the group of students who have dropped out in the last ten years.

The proposed dropout model cannot be evaluated using some metric [33] because the data used with which the model was trained was very few: dropouts with eight rows and non-dropouts with eight rows. We would need more data from faculties of other universities to evaluate the proposed model and observe the behavior of the proposed variables that characterize the faculties.

Table 11. Results of dropout model faculty

Faculty ID	Dropout (using data from the last 20 years)	Dropout (using data from the last 10 years)
F1	0.83	0.74
F2	0.09	0.21
F3	0.83	0.74
F4	0.83	0.74
F5	0.17	0.04
F6	0.09	0.21
F7	0.09	0.21
F8	0.09	0.21

5. Conclusions

This study analyzed which variables affect the most dropout risk in a predictive model for higher education students. Unlike previous works that limited the analysis to just accuracy or results by a specific degree, our work analyzed the variable's behavior segmented by faculty, degree program, and semester to see whether or not there were any differences. The results indicated that the variables related to GPA, socioeconomic factor, and the pass rate of courses taken have a more significant impact on the model than the first semester of studies. This in-depth analysis identified that STEM programs present a different behavior than humanities programs; for example, socioeconomic status has a more significant influence on STEM models than humanities careers, while the pass rate is more important in humanities careers than in STEM. Additionally, in all models, we found a relationship between variables related to academic performance such as GPA and pass rate of courses taken with student dropout similar to [4].

Generalizing the analysis of variable importance for dropout may induce biases due to the data variability in the different careers. For this reason, we analyze from the most general at the faculty level to each semester in each career, using data from a single higher education institution (HEI) that could not be generalized to the level of all Ecuador in all HEIs due to the context of each one. To complement the analysis, we propose a model for predicting dropout based on teacher characteristics by the students of the respective degrees. Data from one HEI was used; future work will require other HEIs to measure the model's effectiveness and draw conclusions about the variables that influence dropout.

These findings' implications lead to in-depth analysis to avoid generalizations when working with predictive dropout models [25,36]. Even within the same institutions, there can be individual differences that can affect the models.

Regarding the limitations of this study, it should be noted that the recent modification of the study plan for all the degree programs had a significant impact. The students who start their studies with the new student plan will finish their degree program in the next four years and not in five. Therefore, the input data of the dropout predictive model will change significantly concerning the one proposed in the present study. Another limitation is the "extraordinary semester"; optional semesters of study during the vacation semester were not included in this study because they are not mandatory.

As future work, we want further to explore the field of academic analytics with this data to help decision-makers, such as program coordinators, make proper adjustments in the university's programs. For instance, we could perform a more in-depth analysis, exploring the results per cohort and different time frames. Moreover, it would be relevant to compare the results obtained in the present study with models of universities in several countries, analyzing the similarities and differences in the multiple contexts.

Acknowledgment. This work was supported in part by the FEDER/Ministerio de Ciencia, Innovación y Universidades–Agencia Estatal de Investigación, through the Smartlet Project (grant TIN2017-85179-C3-1-R) and the H2O Learn Project (grant PID2020-112584RB-C31); the Madrid Regional Government through the e-Madrid-CM Project (Grant S2018/TCS-4307) which is co-funded by the European Structural Funds (FSE and FEDER); and the European Commission through the LALA project (grant 586120-EPP-1-2017-1-ES-EPPKA2-CBHE-JP).

References

1. Abu-Oda, G.S., El-Halees, A.M.: Data mining in higher education: university student dropout case study. International Journal of Data Mining & Knowledge Management Process5(1), 15 (2015)
2. Al-Noshan, A. A., Al-Hagery, M. A., Al-Hodathi, H. A., & Al-Quraishi, M. S. Performance evaluation and comparison of classification algorithms for students at Qassim University. Int. J. Sci. Res, 8(11), 1277-1282 (2018).
3. Albarracín, P., Daniel, J.: Identificación del perfil de egreso correspondiente a la licenciatura de la carrera de laboratorio clınico e histotecnologico de la Universidad central del ecuador periodo 2017-2022 (2016)
4. Ameen, A. O., Alarape, M. A., & Adewole, K. S. STUDENTS'ACADEMIC PERFORMANCE AND DROPOUT PREDICTION. Malaysian Journal Of Computing, 4(2), 278-303 (2019).
5. Ameri, S., Fard, M.J., Chinnam, R.B., Reddy, C.K.: Survival analysis based framework for early prediction of student dropouts. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 903–912 (2016)
6. Aulck, L., Velagapudi, N., Blumenstock, J., West, J.: Predicting student dropout in higher education. arXiv preprint arXiv:1606.06364 (2016)
7. Barbosa Manhaes, L.M., da Cruz, S.M.S., Zimbrao, G.: Towards automatic prediction of student performance in stem undergraduate degree programs. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing. pp. 247–253(2015)
8. Barbu, M., Vilanova, R., Lopez Vicario, J., Pereira, M.J., Alves, P., Podpora, M.,´Angel Prada, M., Moran, A., Torreburno, A., Marin, S., et al.: Data mining tool for academic data

exploitation: literature review and first architecture proposal.Projecto SPEET-Student Profile for Enhancing Engineering Tutoring (2017)
9. Breiman, L.: Random forests. Machine learning45(1), 5–32 (2001)
10. Burgos, C., Campanario, M.L., de la Pena, D., Lara, J.A., Lizcano, D., Martinez, M.A.: Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. Computers & Electrical Engineering 66, 541–556(2018)
11. Crawford, C.: Socioeconomic differences in university outcomes in the uk: dropout, degree completion and degree class. Tech. rep., IFS Working Papers (2014)
12. Chen, R.: Financial aid and student dropout in higher education: A heterogeneous research approach. In: Higher education, pp. 209–239. Springer (2008)
13. Chen, Y., Johri, A., Rangwala, H.: Running out of stem: a comparative study across tem majors of college students at-risk of dropping out early. In: Proceedings ofthe 8th international conference on learning analytics and knowledge. pp. 270–279(2018)
14. Chung, J.Y., Lee, S.: Dropout early warning systems for high school students using machine learning. Children and Youth Services Review96, 346–353 (2019)
15. Del Bonifro, F., Gabbrielli, M., Lisanti, G., Zingaro, S.P.: Student dropout prediction. In: International Conference on Artificial Intelligence in Education. pp.129–140. Springer (2020)
16. Fabara, E.: Cuadernos del contrato social por la educacion. Cuaderno8, 97–98(2013)
17. Fei, M., & Yeung, D. Y. (2015, November). Temporal models for predicting student dropout in massive open online courses. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW) (pp. 256-263). IEEE
18. Gasevi´c, D., Dawson, S., Rogers, T., Gasevic, D.: Learning analytics should notpromote one size fits all: The effects of instructional conditions in predicting academic success. The Internet and Higher Education 28, 68–84 (2016)
19. Gitinabard, N., Khoshnevisan, F., Lynch, C. F., & Wang, E. Y. Your actions or your associates? Predicting certification and dropout in MOOCs with behavioral and social features. arXiv preprint arXiv:1809.00052. (2018).
20. Heredia-Jimenez, V., Jimenez, A., Ortiz-Rojas, M., Marın, J.I., Moreno-Marcos,P.M., Munoz-Merino, P.J., Kloos, C.D.: An early warning dropout model in higher education degree programs: A case study in ecuador (2020)
21. Howard, E., Meehan, M., Parnell, A.: Contrasting prediction methods for earlywarning systems at undergraduate level. The Internet and Higher Education37,66–75 (2018)
22. Jimenez, F., Paoletti, A., Sanchez, G., Sciavicco, G.: Predicting the risk of academic dropout with temporal multiobjective optimization. IEEE Transactions on Learning Technologies12(2), 225–236 (2019)
23. Kang, K., Wang, S.: Analyze and predict student dropout from online programs. In:Proceedings of the 2nd International Conference on Compute and Data Analysis.pp. 6–12 (2018)
24. Luo, Y., Pardos, Z.: Diagnosing university student subject proficiency and predicting degree completion in vector space. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
25. Manrique, R., Nunes, B. P., Marino, O., Casanova, M. A., & Nurmikko-Fuller, T. An analysis of student representation, representative features and classification algorithms to predict degree dropout. In Proceedings of the 9th International Conference on Learning Analytics & Knowledge (pp. 401-410) (2019).
26. Marcılio, W.E., Eler, D.M.: From explanations to feature selection: assessing shap values as feature selection mechanism. In: 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 340–347. IEEE (2020)
27. Marta Ferreyra, M., Avitabile, C., Botero´Alvarez, J., Haimovich Paz, F., Urz´ua,S.: At a crossroads: higher education in Latin America and the Caribbean. TheWorld Bank (2017)

28. Marquez-Vera, C., Cano, A., Romero, C., Noaman, A.Y.M., Mousa Fardoun, H.,Ventura, S.: Early dropout prediction using data mining: a case study with high school students. Expert Systems33(1), 107–124 (2016)
29. Moreno-Marcos, P.M., Alario-Hoyos, C., Muñoz-Merino, P.J., Kloos, C.D.: Pre-diction in moocs: A review and future research directions. IEEE Transactions on Learning Technologies12(3), 384–401 (2018)
30. Moreno-Marcos, P.M., De Laet, T., Munoz-Merino, P.J., Van Soom, C., Broos, T.,Verbert, K., Delgado Kloos, C.: Generalizing predictive models of admission test success based on online interactions. Sustainability11(18), 4940 (2019)
31. Najdi, L., Er-Raha, B.: A novel predictive modeling system to analyze students a trisk of academic failure. International Journal of Computer Applications156(6),25–30 (2016)
32. Ortigosa, A., Carro, R.M., Bravo-Agapito, J., Lizcano, D., Alcolea, J.J., Blanco,O.: From lab to production: Lessons learnt and real-life challenges of an early student-dropout prevention system. IEEE transactions on learning technologies12(2), 264–277 (2019)
33. Pelanek, R.: Metrics for evaluation of student models. Journal of Educational DataMining7(2), 1–19 (2015)
34. Pereira, F.D., Oliveira, E., Cristea, A., Fernandes, D., Silva, L., Aguiar, G., Alamri,A., Alshehri, M.: Early dropout prediction for programming courses supported by online judges. In: International Conference on Artificial Intelligence in Education.pp. 67–72. Springer (2019)
35. Pilotti, M. A., Abdelsalam, H. M., Anjum, F., Daqqa, I., Muhi, I., Latif, R. M., ... & Al-Ameen, T. A. Predicting Math Performance of Middle Eastern Students: The Role of Dispositions. Education Sciences, 12(5), 314. (2022).
36. Rovira, S., Puertas, E., & Igual, L. Data-driven system to predict academic grades and dropout. PLoS one, 12(2), e0171207 (2017).
37. Schneider, M.: Finishing the first lap: The cost of first year student attrition inamerica's four year colleges and universities. American Institutes for Research (2010)
38. Schnepf, S.V.: Do tertiary dropout students really not succeed in european labour markets? (2014)
39. Suganya, S., Narayani, V.: Analysis of students dropout forecasting using data mining,". In: 3rd Internaational Conference on Lastest Trends in Engineering, Science,Humanities and Management (2017)
40. Tang, C., Ouyang, Y., Rong, W., Zhang, J., & Xiong, Z. Time series model for predicting dropout in massive open online courses. In International Conference on Artificial Intelligence in Education (pp. 353-357). Springer, Cham . (2018).
41. Tinto, V.: Dropout from higher education: A theoretical synthesis of recent research. Review of educational research45(1), 89–125 (1975)
42. Vossensteyn, J.J., Kottmann, A., Jongbloed, B.W., Kaiser, F., Cremonini, L., Sten-saker, B., Hovdhaugen, E., Wollscheid, S.: Dropout and completion in higher education in europe: Main report (2015)

Alberto Jiménez-Macías is a PhD student at Universidad Carlos III de Madrid. He obtained a bachelor's degree in Telematics Engineering and a master's degree in Computer Science at the Escuela Superior Politécnica del Litoral (ESPOL) (Ecuador). He carried out development and research work at the Information Technology Center (CTI-ESPOL) for 8 years. His areas of interest are Learning Analytics, Educational Data Mining and Educational Technology.

Pedro Manuel Moreno Marcos is Visiting Professor in the Department of Telematics Engineering at Universidad Carlos III de Madrid (UC3M). He received his Bachelor in

Telecommunications Technologies Engineering in 2015 as well as his Master Degrees in Telecommunication Engineering and Telematic Engineering, which were both obtained in 2017. All of them (bachelor and masters) were obtained at Universidad Carlos III de Madrid. Moreover, he has obtained several awards, including the Extraordinary Awards in the Bachelor Degree and both Master Degrees, and other awards which recognize his academic achievements and Master Thesis. In 2017, he obtained a FPU fellowship to carry out his PhD, which was finished in July 2020. He also obtained the Outstanding Thesis Award. In January 2021, he obtained the positive evaluations from ANECA for the academic positions of Assistant Professor, Private University Professor, and Associate Professor (as non-civil servant). He worked as Specific Teaching Assistant in the academic year 2020/2021, as Assistant Professor in the academic year 2021/2022, and he has been Visiting Professor since September 2022. Currently, he has made 14 publications in JCR-indexed journals and multiple contributions in other journals and conferences. His areas of research interest include learning analytics, Educational Data Mining and MOOCs (Massive Open Online Courses).

Pedro J. Muñoz-Merino is Full Professor at the Department of Telematics Engineering at Universidad Carlos III de Madrid. His main topic of research is on learning analytics. In 2003, he received his Telecommunication Engineering degree from the Polytechnic University of Valencia, and in 2009 his PhD in Telematics Engineering from the Universidad Carlos III de Madrid. Pedro has published more than 150 papers, including more than 50 in journals indexed in the JCR. Pedro has participated in many research projects at the international and national level as well as with companies, being the Principal Investigator in several of them. Pedro has had different Chair positions in different international conferences related to educational technologies such as General Chair at EC-TEL 2023, Program Chair at EC-TEL 2022, Workshop Chair at LAK 2020, Publication Chair at EDM 2020, Program Chair at II LALA conference 2019, Poster chair at EDM 2017, Demo & poster Chair at EC-TEL 2014.

Margarita Ortiz-Rojas holds a PhD in Educational Sciences from Ghent University. Currently, she is the director of the Center of Educational Services at ESPOL in Ecuador. Her research interests include pedagogical innovations, technology in education, learning analytics, gamification and e-learning".

Carlos Delgado Kloos received the Ph.D. degree in Computer Science from the Technische Universität München and in Telecommunications Engineering from the Universidad Politécnica de Madrid. He is Full Professor of Telematics Engineering at the Universidad Carlos III de Madrid, where he is the Director of the GAST research group, Director of the UNESCO Chair on “Scalable Digital Education for All”, and Vice President for Strategy and Digital Education. He is also the Coordinator of the eMadrid research network on Educational Technology in the Region of Madrid. He is Senior Member of IEEE. He has been the Manager of ICT research projects at the Spanish Ministry and has carried out research stays at several universities such as Harvard, MIT, Munich, and Passau.

Received: November 10, 2021; Accepted: November 25, 2022.

Computer Science and Information Systems 20(1):195–214 https://doi.org/10.2298/CSIS220820053B

Solution for TSP/mTSP with an Improved Parallel Clustering and Elitist ACO

Gozde Karatas Baydogmus

Department of Computer Engineering, Marmara University
Istanbul, 34722
gkaratas@marmara.edu.tr

Abstract. Many problems that were considered complex and unsolvable have started to solve and new technologies have emerged through to the development of GPU technology. Solutions have established for NP-Complete and NP-Hard problems with the acceleration of studies in the field of artificial intelligence, which are very interesting for both mathematicians and computer scientists. The most striking one among such problems is the Traveling Salesman Problem in recent years. This problem has solved by artificial intelligence's metaheuristic algorithms such as Genetic algorithm and Ant Colony optimization. However, researchers are always looking for a better solution. In this study, it is aimed to design a low-cost and optimized algorithm for Traveling Salesman Problem by using GPU parallelization, Machine Learning, and Artificial Intelligence approaches. In this manner, the proposed algorithm consists of three stages; Cluster the points in the given dataset with K-means clustering, find the shortest path with Ant Colony in each of the clusters, and connect each cluster at the closest point to the other. These three stages were carried out by parallel programming. The most obvious difference of the study from those found in the literature is that it performs all calculations on the GPU by using Elitist Ant Colony Optimization. For the experimental results, examinations were carried out on a wide variety of datasets in TSPLIB and it was seen that the proposed parallel KMeans-Elitist Ant Colony approach increased the performance by 30% compared to its counterparts.

Keywords: ACO, Parallel ACO, Parallel Kmeans, TSP.

1. Introduction

Traveling salesman problem (TSP) is an optimization problem, which is one of the Non-Deterministic Polynomial (NP) problems and is similar to problems such as GPS, UAV routing, logistics routing [17, 23]. The basic approach in TSP is to find the shortest path that a salesman will eventually return to the starting point by visiting n cities in a sequence at one time. There are many algorithms in the literature for solving this problem. In particular, optimization algorithms such as Genetic Algorithm, Artificial Bee Colony Optimization, Artificial Ant Colony Optimization (ACO), and Particle Swarm Optimization have been frequently preferred for the solution of this problem in recent years. ACO stands out as an algorithm that is widely applied in traditional path planning which is easy for GPU parallelization among other algorithms. Fig. 1 is an example showing the TSP solution on a graph. In Fig. 1, the figure on the left represents a graph, while the figure on the right offers graphs solution.

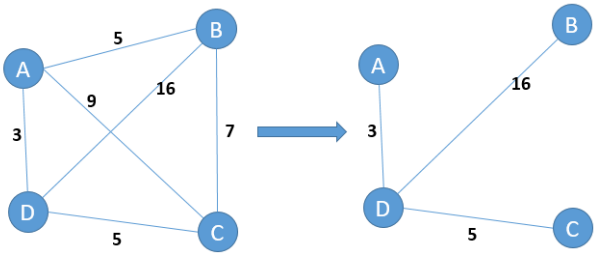


Fig. 1. The example of TSP on a graph without solution

ACO was developed based on the foraging instincts of ants in nature [10–12]. Accordingly, in order to find the shortest path to the food, the ants make use of the pheromone hormone released by the ant on the path it passed while searching for food. This hormone intensifies with each ant passed on and at the same time has the property of evaporation.

Finding a solution by combining clustering and artificial intelligence algorithms has become a favorite of many researchers. Accordingly, the points in the TSP are clustered with the help of the clustering algorithm, so that the closest points to each other are included in different clusters [34]. Then, a connection is made between the clusters created by determining the closest points, and in this way first, the clusters, which are the closest points to each other, are visited and then move to another cluster. The generally preferred algorithm for this is the K-means algorithm [15, 18]. A very easy solution can be found especially for the mTSP problem with this approach.

Parallelization, through the developing technology and GPUs, saves time for solving many complex problems. In this manner, the proposed model in this article designed a parallel approach for the TSP solution. Since the aim is to develop an innovative and successful approach, the literature has been examined in detail and a Parallelized Hybrid algorithm has been proposed considering some deficiencies.

In this study, a new, parallel, and powerful algorithm is proposed for the TSP solution. The strengths of K-means and Elitist ACO algorithms were utilized for this and a new perspective was brought to the studies in the literature. In the proposed method, the following operations carried out:

- The focus of the study is GPU parallelization, so the operations were carried out in parallel whenever possible.
- First of all, various datasets in TSPLIB were selected for the TSP solution and prepared for processing. Particular attention has been paid to the fact that the selected datasets have different numbers of nodes/points.
- Parallel clustering was performed with K-means in the TSP datasets. The K value is determined according to the m value by considering the mTSP problem. Calculation of the distances of other points to the center points determined for clustering was carried out in parallel. In this way, the processing load is reduced by 1/K.
- After clustering, the closest points of K clusters were determined and the clusters were connected to each other from these points.
- Another important part of clustering is the implementation of the Parallel Elitist ACO algorithm. Ants were sent to random points in the dataset for this. The ant will first

wander around in the cluster to which the point it is located, and then move on to another cluster from the connection point. The ant's circulation and the total distance achieved at the end of this circulation were calculated in parallel.
– After all the conditions are provided, a result that can be optimal for TSP is achieved through the parallel hybrid algorithm.

In the following part of the study, detailed information about the related work in Related Work, the methods used and the proposed parallel hybrid algorithm in Proposed Approach are given. In Proposed Approach section, a flowchart of the developed algorithm is given, and then all the steps are explained in detail. Experimental results on selected TSP datasets are shown in Experimental Results section. Discussion and Conclusion section contains the conclusion part of the study.

2. Related Work

In this section, studies which are examines clustering and metaheuristic algorithms in the literature are evaluated. As a result of the literature review, it was discovered that no research study uses both clustering and routing using the GPU parallelization aimed by this study. In this manner, the model proposed in this study brings a new perspective to the literature on GPU parallelization.

In 2022, researchers aimed to design an ACO that would solve TSP by utilizing algorithm improvements and multi-core CPUs and named the algorithm they designed as FACO [30]. Accordingly, at FACO the number of differences between the newly created and selected previous solution is checked and improvements made while maintaining the quality of the existing solution through a more focused search process. The results of the study were examined on many TSPs and it was seen that successful results were obtained with 8-core CPUs and problems with more than 100.000 nodes.

Dihn and others researched a study in 2021 to examine parallel drone scheduling using TSP, where deliveries are split between a truck and a fleet of drones [8]. The solution to the problem is to focus on the known TSP and develop a method. In this context, they proposed a hybrid ACO to solve the problem. The proposed algorithm focuses on a method that represents a TSP solution as a permutation of all data, and then performs the solution by dynamic programming. The hybrid algorithm designed by the researchers is completely based on dynamic programming. For the study, 90 samples were evaluated and it was seen that they gave better results than many studies recommended in the literature.

In 2022, researchers designed an algorithm that takes advantage of ACO to solve, improve overall performance, and shorten solution time in TSP [31]. For this, clustering on ACO parameters, dynamic pheromone evaporation and diversity of solutions in the population were used. In order to observe the working performance, a study was carried out on the problems with the number of nodes ranging from 51 to 2392 and it was observed that the proposed method was successful like the examples in the literature.

Rani et al. proposed an algorithm to determine the travel path in the most optimal way using the Traveling salesman problem and the K-means clustering technique in 2018. The purpose of the research is to develop a web-based application that can help people plan their travels. It proceeds on the assumption that the traveler determines the touristic places they want to visit and the number of days they will stay in the region. The proposed

approach consists of two stages, macro grouping using k-means and micro tour editing using the traveling salesman problem. In the study, operations were carried out in the city of Yogyakarta, one of the touristic cities of Indonesia. As a result of the experiments it was seen that the proposed algorithm works well on a small to a medium number of points [28].

The approach developed by Cheng and Mao in 2007; aims to find the minimum cost path using time Windows for the traveling salesman problem. For this, they improved ant colony optimization, which is very popular among metaheuristic algorithms and is frequently used in solving the traveling salesman problem. In the study, they embedded two local heuristics in the ant colony algorithm to manage time window constraints. As a result of the experimental studies, it was discovered that the proposed algorithm solves the traveling salesman problem more efficiently than the standard ant colony algorithm [6].

The algorithm proposed by Stodola et al. designs three new techniques to reduce the negative effects associated with the ant colony optimization method on the traveling salesman problem, such as improving overall performance and reducing to a local optimum. These techniques are the concept of node clustering, adaptive pheromone evaporation, and diversity of solutions in the population. 30 benchmark data samples from well-known TSPLIB benchmarks were used to evaluate the performance of the proposed method and different comparisons with experimental results were performed. The proposed algorithm outperformed these competing methods available in the literature in most cases [31].

In 2014, Bora and Gupta observed the effect of different distance measurement techniques on clustering using the K-means algorithm. This study stands out, especially because distance calculation is very important in clustering algorithms. Manhattan, Euclid, and Cosine distance calculations were evaluated in the study. In addition, Matlab was preferred to improve the study. The values obtained as a result of this study are very important in terms of determining the algorithm to be selected according to the characteristics of the dataset to be used while clustering [3].

Researchers worked on K-Means and Cross Ant Colony Optimization methods to solve the multiple traveling salesman problem in 2020 [20]. They used the K-Means algorithm to determine the ant colony to determine the tour to find the area that each seller would visit. Experimental results were conducted on three different datasets chosen from the application TSBLIB and a different number of salesmen such as 2, 3, 4, and 8. As a result of the application, it was seen that the proposed algorithm gave better results than the K-Means and Ant colony optimization working alone. In addition, the effect of the number of vendors on the operation of the model was also observed.

In 2009, researchers conducted an optimized routing study for multiple Traveling Salesman Problem [26]. In the study, the K-means clustering algorithm was used to cluster the given cities depending on the number of a salesman. In this way, the m-Traveling Salesman Problem problem is reduced to a simple traveling salesman problem. After clustering, an optimized route is created for each salesman in its allocated cluster using metaheuristic algorithms. For this, ”Tabu Search” and ”Simulated Annealing” metaheuristic algorithms were preferred. In the experimental results, it has been seen that Simulated Annealing gives a shorter path than the Tabu search.

The Traveling salesman problem is a problem that can be summarized as finding the shortest path circulating between certain points by the salesmen. The m-traveling salesman problem is generalized to take into account more than one salesman. Latah conducted

research in 2016 to find a solution to the m-Traveling salesman problem [21]. In the study, the author proposed a solution method by using ant colony optimization and a genetic algorithm. Accordingly, Latah proposed a new model by running K-means clustering algorithm on ant colony optimization. Popular datasets available on TSBLIB were used to test the study. As a result of the study, the author observed that the modified ant colony algorithm he proposed gave better results than the genetic algorithm.

In 2019, researchers proposed a solution to the traveling salesman problem using the Firefly Algorithm (FA) and K-means clustering [16]. This proposed approach consists of three steps: clustering the points in the dataset, finding the optimal path in each cluster, and creating a connection path between the clusters. In the first step, nodes are divided into sub-problems using K-means, in the second step, the optimal path in each cluster is found by FA, and finally, all clusters are reconnected. Experimental results showed that the proposed approach gave better results compared to other algorithms in the literature.

Chang aimed to increase the efficiency of ant colony optimization by using the K-means algorithm to offer a new perspective on the traveling salesman problem [4]. In the study, the city locations were divided into two or more groups according to the use of the K-means algorithm, and then circulation was carried out within these city groups with ant colony optimization. Finally, a connection was made between these clusters at the closest points. Experimental results showed that the proposed method reduces the computational cost by 32%.

3. Proposed Approach

In this section, the algorithms used in the study are given, and then the proposed model is explained in detail.

3.1. Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is a problem that aims to find the shortest tour that passes through each of a certain number of points with known distances only once. It can be solved by calculating the permutations between all points while the number of points is low, but when the number of points increases, the cost of permutations will increase too much, and after a while, it begins to be unsolvable in polynomial time. For this reason, it started to attract the attention of researchers, and many methods have been developed for a fast and effective solution [17, 23].

TSP can be shown by a graph G=(V, E) are V is the set of nodes/points where V = 1,2,3,. . . ,n and E is the set of edges (distance between two points) connecting the nodes in set V. Euclidean distance is mostly used to calculate the path length since TSP aims to find the shortest path to visit all points. This formula is shown by Equation 1.

$$dist(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

where $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ equivalent to the coordinates of the points [1, 14].

3.2. K-Means Clustering

In human life, people do their work by grouping due to their nature. The librarian separates the books according to certain groups while organizing the library or the researcher organizes his/hers belongings according to the season, and groups them by examining the studies related to the subject while researching for his/her publications. Grouping similar objects together have always been in human life.

Computer science, which designs approaches by being influenced by human life, was also affected by this idea and put forward the clustering approach, the infrastructure of which was formed by this idea. Clustering is one of the most common statistical data analysis methods used to obtain a relationship about the structure of the available data [35]. Because we may not always have meaningful data when processing with computers, so knowing how to relate data increases the performance of the model [15, 19, 22]. Clustering, which is an unsupervised learning method, gives better results on such datasets. Clustering algorithms aim to establish relationships between such data. Fig. 2 shows the state of the data before and after clustering.

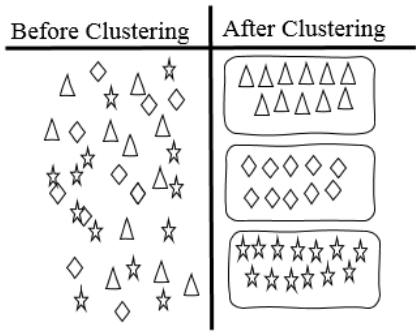


Fig. 2. Clustering

There are two types of clustering algorithms that are performed in Machine Learning; K-means and Hierarchical Clustering. In this work, the K-means algorithm, which is accepted as one of the most used clustering algorithms due to its simplicity and applicability, will be used to determine the relationship between places/locations. Application areas of the K-means algorithm; customer segmentation, game/player analysis, document classification, intrusion/fraud detection, etc. The main problem here is to choose the K value correctly. Because it is necessary to manually (intuitively) assign the K value. There are several methods for determining the appropriate K number, these are; Elbow Method, Average Silhouette Indices, and GAP statistic.

K-means algorithm's main purpose is to determine the center points by minimizing the distance within the cluster. For this, it assumes that the data at hand consists of K clusters and tries to minimize the distance of the points in the clusters to be formed from the cluster mean [18]. The algorithm is based on the Euclidean distance formula given by Equation 2;

$$\sum_{j=1}^{K}\sum_{i=1}^{N}\left\|x_i^j - C_j\right\|^2 \tag{2}$$

In Equation 2, N is the size of the dataset, $X = (x_1, x_2, \ldots, x_n)$ are the points an $C = (c_1, c2_2, \ldots, c_K)$ are centroids in the available dataset. The algorithm performs the following steps;

1. Start with randomly selected K (number of clusters) center points, $\mu_1, \mu_2, ..., \mu_K$.
2. Assign each point in the dataset to the cluster with its closest centroid (based on the distance calculated by the Euclidean distance formula).
3. Calculate the value of the cluster center by averaging all its points. It is checked with Equation 3 and Equation 4 whether the clustering is completed or not.

   Adjust

$$C^{(i)} = argmin_j \left\|x^{(i)} - \mu_j\right\|^2 \tag{3}$$

   For every i, and adjust

$$\mu_j = \frac{\sum_{i=1}^{m} 1\left\{C^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^{m} 1\left\{C^{(i)} = j\right\}} \tag{4}$$

for each j.

Hierarchical clustering was not preferred in this study. As it is known, in hierarchical cluster analysis, similarity and distance calculations between data are updated at every step. This causes it to be slower than the K-means algorithm in terms of time in large data sets. Therefore, K-means was preferred in the study.

3.3. Ant Colony Optimization and Elitist Ant Colony Optimization

Metaheuristic algorithms of artificial intelligence have started to gain attention after the increase in access to information and the researchers' turn of their observations into nature [5, 24, 33]. Many algorithms such as genetic algorithm, Artificial Bee colony, and particle swarm optimization have been developed by examining the existence of living things/matter in nature. The main method used to create a shortest path in this study, the Ant Colony Optimization (ACO) algorithm, was formulated on ants' finding the shortest path to food as a result of their wandering in nature.

ACO was first discovered by Marco Dorigo [10–12]. It has been observed that ants can find the shortest path to food as a result of their circulation to find food, and a related algorithm has been designed. Foraging for food in nature, the ant leaves a hormone called pheromone on its way to and from food. Since this hormone has an evaporating structure, it stays on the road for a certain period of time, the level of pheromone increases with each pass, and the level of pheromone decreases when there is no passage. Since the pheromone level on the road will increase over time because evaporation will be less in a short way, so the pheromone level will be higher, all ants will intuitively start to prefer the path with the highest pheromone level. The formulas developed by Dorigo are given by Equation 5 and Equation 6.

$$\tau_{ij}{}^{k} = (1-\rho)\tau_{ij}{}^{k} + \sum_{k=1}^{n} \Delta\tau_{ij}{}^{k} \tag{5}$$

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l\in\mathbb{N}}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}}, \eta_{ij} = \frac{1}{L_{ij}} \tag{6}$$

where p_{ij}^{k} is the probability of the k^{th} ant moving from node i to node j, τ_{ij} amount of pheromone between nodes i-j, η_{ij} cost function between nodes i and j, α pheromone and β heuristic coefficient, n is ant colony size, ρ is the evaporation coefficient. Equation 5 determines the pheromone level between two points by considering the evaporation coefficient. The probability calculation made by Equation 6 is used to determine which of the path will be selected depending on the current point [7, 27, 32].

Elitist ant colony (EACO) is the first system developed by ACO, inspired by applications in genetic algorithm [9]. According to this approach, the pheromone amounts of the edges of the best round are subjected to a pheromone increase in addition to the standard pheromone update given by Equation 5 in the ACO at each cycle [13]. Elitist ACO's difference from the ACO is that the update of the pheromone trails is carried out by the elitist ants, which can be one or several. Equation 7 and Equation 8 are formulas for calculating Elitist ACO.

$$\tau_{ij}(t) = p.\tau_{ij}(t-1) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} + E.\Delta\tau_{ij}^{bs} \tag{7}$$

$$\tau_{ij}^{bs} = \frac{Q}{L^{*}} \tag{8}$$

In Equation 7 and Equation 8, E is the number of elitist ants which we will call E the elitist coefficient. , L^{*} is the length of the best round, τ_{ij}^{bs} is for the best solution and the amount of pheromone belonging to the edges on the best round is increased by $e\frac{Q}{L^{*}}$.

In EACO the elitist strategy for pheromone updating is also involved and Equation 7 is used for pheromone update. Apart from these, probability calculation and selection are exactly the same as ACO. The flow chart for the proposed algorithm is given in Fig. 3.

3.4. Parallel Clustered Elitist Ant Colony Algorithm for TSP

Graphics Processor Units are converters work between the screen and the processor during the creation of the text and graphics displayed on the computer. It undertakes the tasks of processing and reflecting graphics on many technological devices that you actively use in daily life, especially personal computers, smart phones, workstations, digital screens and game consoles.

In this section, information is given about the developed parallel model proposed in the study. When the literature is examined, it is seen that there is no study suggesting a new parallel approach by using clustering and EACO algorithms together. This study consists of three stages; Clustering with parallel K-means, interconnection between clusters at the closest points/nodes and routing with parallel EACO. In this way, a connection will be established between the shortest nodes between the clusters and the algorithm will find

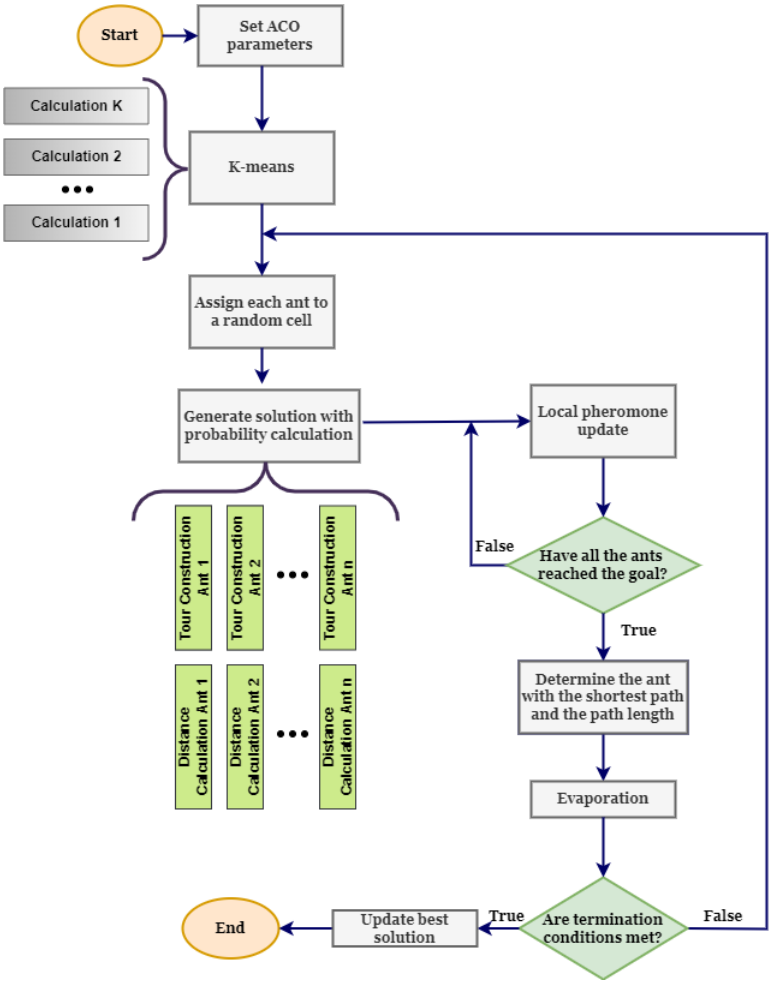


Fig. 3. Flow chart for Proposed Parallel K-EACO

the shortest path in the graph. In this paper, the GPU parallelization is used in calculating the distance of nodes belonging to each center in clustering, while in EACO it is used in determining the tour of the ants. The aim of the proposed study is to design a new optimization model with low computational cost. In the following part, the flow chart of the study is given, and then the step-by-step procedures are explained.

Set ACO Parameters; Determining the parameters in the code is one of the most important parts of the study and also be named Initialization. Critical formulas for both the K-means algorithm and the EACO algorithm were used in the study; these formulas are given in K-Means Clustering and Ant Colony Optimization sections preliminary work done to determine the values that the parameters used in these formulas will take. Especially the number of clusters used, the elitist coefficient in the EACO algorithm, the number of iterations, and colonies are the important ones. Since routing applications using K-means clustering have become popular in recent years, there are quality studies on this subject in the literature. In this manner, [20,26] studies were examined and it was decided to take the K value of m which is the salesman number in mTSP. In this study, the K value was examined separately by taking 2, 4, and 8. In addition, since it is recommended to take the Elitist coefficient as 0.5 in the published books and articles for the Elitist coefficient, the study was carried out in that direction [25]. The researcher who developed this study has done various research and publication for ACO before, so a preliminary study was made for other important parameters and the parameters were determined accordingly. The values assigned to important parameters as a result of preliminary studies and literature searches are given in Table 1. It was seen that the average results were obtained with 100 iterations is good for the experiment and the results were interpreted over this number of iterations.

Table 1. Parameters

Parameter	Value
α	1.0
β	5.0
ρ	0.5
E_{val}	0.5
initial pheromone	0.01
number of iterations	10, 50, 100, 500, 1000
number of nodes (n)	52, 76, 127, 200, 318, 439
colony size	52, 76, 127, 200, 318, 439, 1002,1024

Parallel K-means Algorithm; It is the first important stage of the study. Clustering is an unsupervised learning algorithm that has been used in machine learning algorithm for years. It has been frequently used by researchers in new problems with its popularity. This study also wanted to benefit from the powerful computation of the K-means algorithm and the appropriate algorithm flow for parallelization. Though to K-means, it is aimed to create clusters consisting of the closest nodes to each other and allow the ant to circulate within these clusters, thus reducing both the ant's circulation and the cost/time. Since the main goal of the study is to calculate the shortest path in the shortest time, the K-means algorithm was also run in parallel.

Parallel K-means algorithm starts with the selection of the center nodes. According to Table I, there are 2, 4, and 8. For each K value, the operations mentioned in the following section were carried out separately. After the center nodes are selected, the distances of the other nodes to each center node are calculated in parallel. It means, the distance of each node to the selected center is performed simultaneously in a parallel manner. In this way, when one center is finished, the time lost by calculating the other is eliminated. The distance calculation was carried out with the Equation 2. After calculating the distances of the nodes to the central nodes, the process of determining the elements of the clusters sequentially was carried out. After all, when clusters are created, the central nodes are chosen again, the distances to these center nodes are calculated in parallel and the clustering process is performed sequentially. It is then compared with the previous clustering results. If there is no change, the K-means algorithm is completed, otherwise, the same processes are repeated by choosing the central nodes again.

– *Finding the closest nodes;* Since the aim is to perform a circulation on all the given nodes, when the K-means algorithm is done and the clusters determined, it is necessary to connect the clusters at the closest nodes. An example of this is given in Fig. 4. The Euclid algorithm given by Equation 2 was used to determine the closest nodes between the clusters. For this process, the distance between the points was calculated only between the clusters belonging to the closest center nodes. The calculation of the distances between the nodes was also carried out in parallel. Then, the closest nodes were determined and each cluster was connected as if there was a bridge between them.

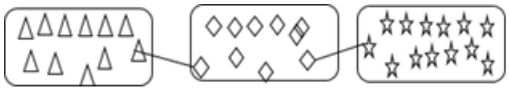


Fig. 4. Connecting Clusters

EACO operations performed with GPU parallelization are shown in Fig. 6.

– *Assign each ant to a random cell;* At this stage, all ants in the colony are randomly assigned to the existing nodes to solve the TSP problem, to start the circulation of finding the shortest path.
– *Generate Solution with Probability Calculation;* This is the second important part of the study. At this stage, three operation performed; Tour construction, finding the closest nodes between clusters, and total tour length of the path. Each of them has been examined separately.
– *Tour Construction;* Another important part of GPU parallelization is the tour construction part. In this part, every ant performs its circulation on the different GPU core. The ant first completes the circulation in the cluster it is in, then moves to the other cluster that closest to the cluster it is in and performs this operation until it completes all the circulations in all clusters. In this way, it performs a circulation between the closest nodes. For this circulation, given steps in Elitist Ant Colony Optimization section which called EACO are followed. Accordingly, the ant assigned to

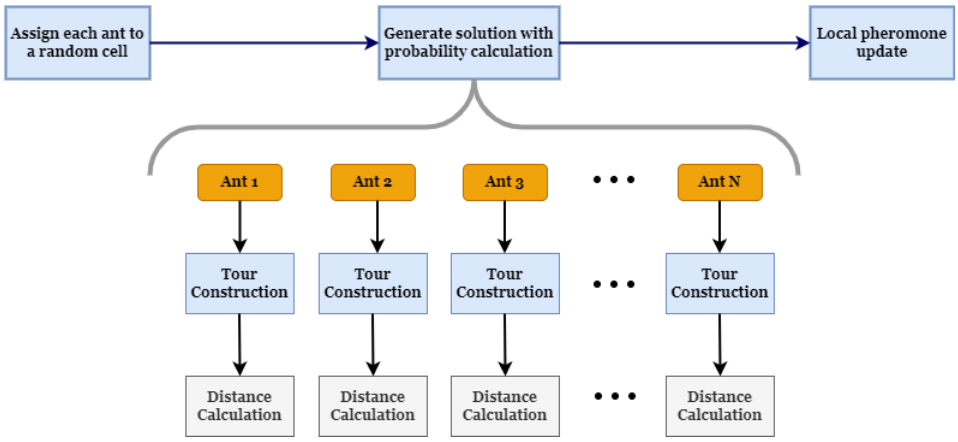


Fig. 5. Parallel Elitist ACO

a random node makes a probability calculation among the nodes it can choose, using the formula given in Equation 6. Then it chooses one of these possibilities with the Roulette Wheel Selection algorithm and moves to the next node. The important part is that the ant first circulates the nodes of the cluster it belongs to, and then moves to the next cluster. Since this process takes place in parallel an ant basis, it provides a great advantage in terms of cost which is time.

– *Distance Calculation;* After tour is completed it is necessary to calculate the fitness function of the ants (in TSP, this is the total path length or the total distance). For this calculation, the Euclidean distance algorithm given in Equation 1 was used and the distance information of each circulation was updated for all ants. Since the aim is to find the shortest path between both the current iteration and all iterations, the length of each circulation is needed. The operations was done with the parallelization on the GPU.

From here on, the following operations were carried out sequentially, meant without parallelization.

Pheromone Update; One of the most important information affecting operations is the pheromone value on the path used while applying algorithms. Through to the pheromone value, the ant decides on the node it will take, so after the circulation on the graph is completed, the pheromones of the paths must be updated again depending on the situation. For the EACO algorithm, calculations were done with Equation 7 and Equation 8.

Determining the shortest path; To determine the shortest path, the circulation paths of all ants are examined, the shortest path is selected and the ant that circulated this path is determined.

Evaporation; Evaporation is an important factor for ants, as the pheromone hormone has a structure that evaporates over time. Accordingly, while applying the EACO algorithm, it is necessary to add the evaporation rate for the work. Equation 5 was used to calculate the evaporation rate of the ants' paths.

4. Experimental Results

In this section, comparison of the developed algorithm with the existing standard method and the accuracy rate of the parallel hybrid algorithm are discussed. For improved Parallel Clustered EACO which is going to be called the K-EACO algorithm, the GPU parallelization which is given in Ant Colony Optimization section is tested and comparisons are made on various TSP libraries ranging in cities from 51 to 1002. Algorithms and parameters were applied as given in the initialization phase which is shown in Table 1. Python programming language have been used for experimental results. Only Standard Python libraries such as Numpy, Pandas, and Numba were used for the proposed algorithm, and the study was created entirely with my codes. Numpy and Pandas libraries are used for accessing dataset, and Numba is used for GPU parallelization. In addition, the ACO algorithm, which was chosen as the comparison algorithm, was also run in parallel. CPU/GPU benchmarks for this algorithm are available in another work by the author. Some of the environmental properties that need to be depicted are shown in Table 2.

Table 2. Environment

Hardware	Features
CPU	Intel(R) Core(Tm) I7-8700 Cpu @3192Mhz, 6 Cores
Op. Sys.	64 bit, Windows 10
Grap.card	NVIDIA GeForce® GTX 1080 Ti Founders Edition 11G
L1/L2/L3 Cache	384 KB/1.5 MB/12.0 MB
RAM	16.00 GB

GPU specifications of the computer can be seen in the Table 3.

Table 3. GPU Specifications

GPU Specifications	
CUDA Cores	3584
Graphics Clock (MHz)	1480
Processor Clock (MHz)	1582
Memory Clock (MHz)	1376
Memory Config	11 GB
Memory Interface Width	352 bit
Memory Interface	GDDR5X
Memory Bandwidth (GB/sec)	11 Gbps

Since the main purpose of this study is to focus on the TSP/mTSP solution, 7 popular datasets in the TSPLIB library were used [29]. These datasets and their optimal solutions are given in Table 4.

Table 4. TSPs and Optimal Solutions

Problem	Number of Cities	Optimal Solution
berlin52	52	7542
eli76	76	538
bier127	127	118282
kroA200	200	29368
lin318	318	42029
pr439	439	107217
pr1002	1002	118282

In the study, ant's circulation time which is the time taken to find the shortest path, and error rate were examined as evaluation criteria. The formula given in Equation 9 was applied for the error rate.

$$ErrorRate = \frac{d - d'}{d'} \tag{9}$$

which $d^{'}$ is the best solution length, d is the solution found by the algorithm. One of the important evaluations in the study is the value of K to be used in the K-means algorithm, that is, the number of clusters. When the literature is examined, the studies on this subject are investigated and seen that the value of K should be 2, 4, and 8 in most of them. In this context, the results of the serial and parallel average execution time (in millisecond) of the K-means algorithm according to the determined number of K values are given in Table 5.

Table 5. Sequential-Parallel K-means Time Comparison

K	Time (ms)	
	Sequential K-means	Parallel K-means
2	0.0150	0.0001
4	0.0245	0.0001
8	0.0309	0.0001

It was seen that clustering worked much faster with GPU parallelization when the results were evaluated. After the parallel K-means algorithm was completed, these clusters were connected at their closest points at each other. The results of the determination of the points and matching operations are added to the runtime results of the developed algorithm.

In the following part, comparisons were made choosing the K value of 8. Table 6 compares the times for finding shortest path of the existing ACO algorithm with the advanced parallel K-EACO algorithm. In Table 6, the evaluations are shown on a different row for each node.

Since the important thing in this study is parallelization, the results of the use of clustering and EACO without parallelization should be compared with those obtained as

Table 6. ACO and K-EACO Time Comparison

Number of Nodes	GPU Time (ms)	
	ACO	K-EACO
52	30.4	16.09
76	48.72	30.88
127	145.16	88.27
200	389.86	197.94
318	1125.96	601.42
439	2489.45	1709.61
1002	22056.11	17746.19

a result of working with GPU. Table 7 shows the running times of Sequential K-EACO (seq.) and Parallel K-EACO (par.) algorithms.

Table 7. Seq. and Par. Time Comparison

Number of Nodes	CPU vs GPU Time (ms)	
	Seq.	Par.
52	32.48	16.09
76	57.73	30.88
127	157.66	88.27
200	420.66	197.9
318	1197.12	601.42
439	2597.80	1709.61
1002	22734.96	17746.19

Fig. 6 shows the running time of the parallel and sequential model. When the figure is examined, it is seen that the parallel working model works much faster. The actual running time of the sequential K-EACO (seq.) and parallel K-EACO (par.) algorithms when the K value is 8 shown in Fig. 6. sequential K-EACO algorithm is applied in parallel, it does not give results as fast as K-EACO. The reason for this is that the ant move faster through in a set of points close to each other.

Another criterion as important as time is the error rate. Fig. 7 compares the error rates achieved by applying sequential K-EACO and improved parallel K-EACO to individual dataset. It is seen that the error rates are very close to each other, but the error rate is much lower with the proposed algorithm when the figure is examined.

Fig. 8 shows the shortest path graph reached as a result of circulating with sequential K-EACO the lin318 dataset, which has an average data number, and by running it with the advanced K-EACO algorithm in Fig. 9.

When all the given results and figures are examined, it is seen that the proposed improved algorithm gives much better results both in terms of error rate and time. Although the ACO algorithm was run in parallel, it could not pass the proposed algorithm, because it reduces the error rate and reduces the circulation time by circulating between related points through to clustering.

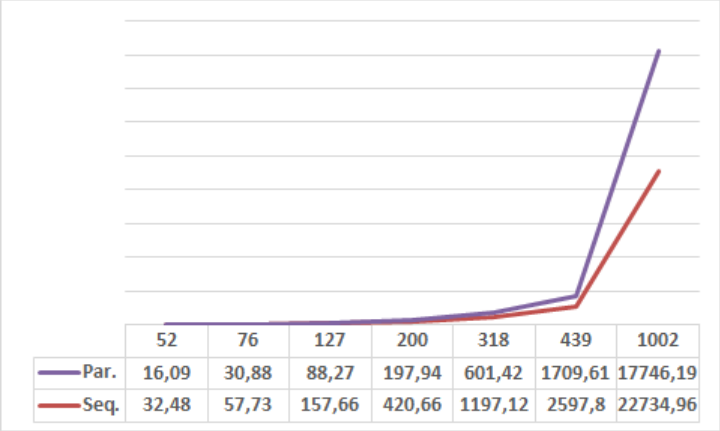


Fig. 6. Time Comparison for seq. and par.

5. Discussion and Conclusion

In the study, it is aimed to obtain a reliable algorithm that works faster by using the power of clustering in establishing relationships between data. In addition, since the performance of artificial intelligence algorithms in solving real world problems has increased in recent years, combining them with different algorithms provides more successful results. For this reason, an optimization algorithm that gives fast and successful results is designed by taking the strengths of both algorithms.

Shortest path-finding algorithms have started to draw attention due to the popularization of internet of things devices and the need for GPS use in almost every field. In this manner, TSP is a problem that attracts the attention of researchers and successful GPS suggestions can be made if a solution is found. There are many studies in the literature to solve this problem, but in these studies, there is no method that applies K-means and EACO by using GPU parallelization like in this article. As a result of the preliminary studies, it has been seen that the parallelization on the GPU works much faster than the CPU and the shortest path on the ACO is found more optimal solution through clustering. Therefore, parallelization in the study was carried out in both K-means and EACO algorithms.

Experimental results were carried out on GPU using various TSP datasets available in TSBLIB. Results on the CPU were not included in the study because it was previously published as another academic study [2]. As a result of the study, it has been seen that the proposed new method has low time complexity and very successful in finding the optimal result, and it is 30% efficient This study will give an idea to other researchers in terms of providing a perspective on the EACO method and showing that it works efficiently with GPU, especially for researchers looking for a new study area. Other optimization algorithms of artificial intelligence can also be considered for the study. In the literature, it has been seen that the genetic algorithm is very successful in PEP studies. Therefore, the use of ACO and Genetic algorithm together can be evaluated in future studies. The research on this subject continues.

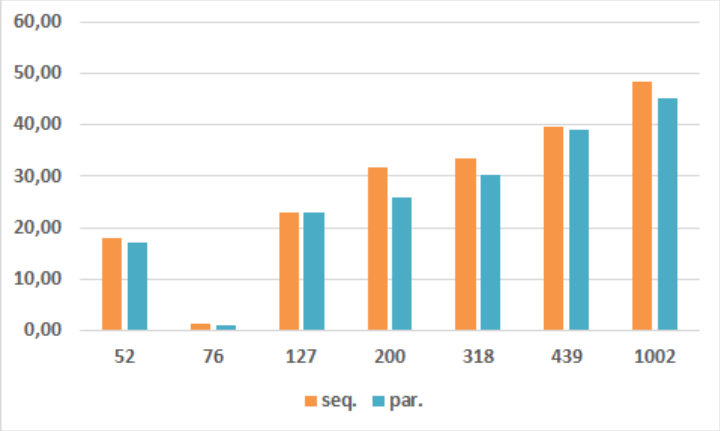


Fig. 7. Error Rate Comparison for seq. and par.

References

1. Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The traveling salesman problem. In: The Traveling Salesman Problem. Princeton university press (2011)
2. Baydogmus, G.K.: A parallelization based ant colony optimization for travelling salesman problem. In: 2022 1st International Conference on Information System & Information Technology (ICISIT). pp. 166–169. IEEE (2022)
3. Bora, M., Jyoti, D., Gupta, D., Kumar, A.: Effect of different distance measures on the performance of k-means algorithm: an experimental study in matlab. arXiv preprint arXiv:1405.7471 (2014)
4. Chang, Y.C.: Using k-means clustering to improve the efficiency of ant colony optimization for the traveling salesman problem. In: 2017 IEEE international conference on systems, man, and cybernetics (SMC). pp. 379–384. IEEE (2017)
5. Chen, M.Y., Rubio, J.d.J., Sangaiah, A.K.: Guest editorial-pattern recognition, optimization, neural computing and applications in smart city. Computer Science and Information Systems 18(4), 0–0 (2021)
6. Cheng, C.B., Mao, C.P.: A modified ant colony system for solving the travelling salesman problem with time windows. Mathematical and Computer Modelling 46(9-10), 1225–1235 (2007)
7. Deng, W., Xu, J., Zhao, H.: An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. IEEE access 7, 20281–20292 (2019)
8. Dinh, Q.T., Do, D.D., Hà, M.H.: Ants can solve the parallel drone scheduling traveling salesman problem. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 14–21 (2021)
9. Dorigo, M.: Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale. Ph.D. thesis, PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (1992)
10. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE computational intelligence magazine 1(4), 28–39 (2006)
11. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical computer science 344(2-3), 243–278 (2005)
12. Dorigo, M., Stützle, T.: Ant colony optimization: overview and recent advances. Handbook of metaheuristics pp. 311–351 (2019)

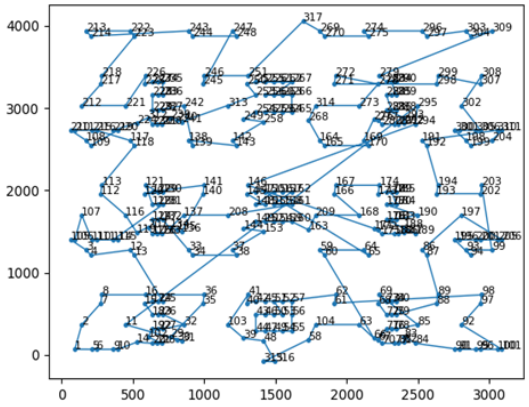


Fig. 8. lin318 with solution sequential K-EACO

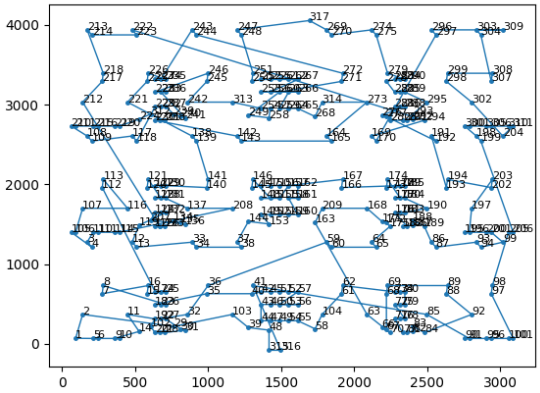


Fig. 9. lin318 with solution parallel K-EACO

13. ESEN, H., Söyler, H., KESKİNTÜRK, T.: Global karınca koloni algoritmasının simetrik ve simetrik olmayan gezgin satıcı problemlerine uygulanması
14. Gutin, G., Punnen, A.P.: The traveling salesman problem and its variations, vol. 12. Springer Science & Business Media (2006)
15. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) 28(1), 100–108 (1979)
16. Jaradat, A., Diabat, W., et al.: Solving traveling salesman problem using firefly algorithm and k-means clustering. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). pp. 586–589. IEEE (2019)
17. Jünger, M., Reinelt, G., Rinaldi, G.: The traveling salesman problem. Handbooks in operations research and management science 7, 225–330 (1995)
18. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE transactions on pattern analysis and machine intelligence 24(7), 881–892 (2002)

19. Kodinariya, T.M., Makwana, P.R.: Review on determining number of cluster in k-means clustering. International Journal 1(6), 90–95 (2013)
20. Kusumahardhini, N., Hertono, G., Handari, B.: Implementation of k-means and crossover ant colony optimization algorithm on multiple traveling salesman problem. In: Journal of Physics: Conference Series. vol. 1442, p. 012035. IOP Publishing (2020)
21. Latah, M.: Solving multiple tsp problem by k-means and crossover based modified aco algorithm. International Journal of Engineering Research and Technology 5(02) (2016)
22. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern recognition 36(2), 451–461 (2003)
23. Lin, S.: Computer solutions of the traveling salesman problem. Bell System Technical Journal 44(10), 2245–2269 (1965)
24. Liu, Z., Jiang, G.: Optimization of intelligent heating ventilation air conditioning system in urban building based on bim and artificial intelligence technology. Computer Science and Information Systems 18(4), 1379–1394 (2021)
25. Merkle, D., Middendorf, M., Schmeck, H.: Ant colony optimization for resource-constrained project scheduling. IEEE transactions on evolutionary computation 6(4), 333–346 (2002)
26. Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., Parthiban, P.: Optimization of non-linear multiple traveling salesman problem using k-means clustering, shrink wrap algorithm and meta-heuristics. International Journal of Nonlinear Science 8(4), 480–487 (2009)
27. Paniri, M., Dowlatshahi, M.B., Nezamabadi-Pour, H.: Mlaco: A multi-label feature selection algorithm based on ant colony optimization. Knowledge-Based Systems 192, 105285 (2020)
28. Rani, S., Kholidah, K.N., Huda, S.N.: A development of travel itinerary planning application using traveling salesman problem and k-means clustering approach. In: Proceedings of the 2018 7th International Conference on Software and Computer Applications. pp. 327–331 (2018)
29. Reinelt, G.: Tsplib—a traveling salesman problem library. ORSA journal on computing 3(4), 376–384 (1991)
30. Skinderowicz, R.: Improving ant colony optimization efficiency for solving large tsp instances. Applied Soft Computing 120, 108653 (2022)
31. Stodola, P., Otřísal, P., Hasilová, K.: Adaptive ant colony optimization with node clustering applied to the travelling salesman problem. Swarm and Evolutionary Computation 70, 101056 (2022)
32. Xiao, J., Li, C., Zhou, J.: Minimization of energy consumption for routing in high-density wireless sensor networks based on adaptive elite ant colony optimization. Journal of Sensors 2021 (2021)
33. Yang, R., Li, D.: Adaptive wavelet transform based on artificial fish swarm optimization and fuzzy c-means method for noisy image segmentation. Computer Science and Information Systems (00), 39–39 (2022)
34. Yildiz, K., Çamurcu, A.Y., Dogan, B.: Comparison of dimension reduction techniques on high dimensional datasets. Int. Arab J. Inf. Technol. 15(2), 256–262 (2018)
35. Yıldız, K., Çamurcu, Y., Doğan, B.: Veri madenciliğinde temel bileşenler analizi ve negatifsiz matris çarpanlarına ayırma tekniklerinin karşılaştırmalı analizi. Akademik Bilişim 10, 248 (2010)

Gozde Karatas Baydogmus was born in Istanbul, Turkey, in 1991. She received the bachelor's degree from the Mathematics and Computer Science Department, Istanbul Kültür University, in 2009, the M.S. degree from the Computer Engineering Department, Istanbul Kültür University, in 2013 and Ph.D. degree the Computer Engineering Department from Marmara University. She continues to work in the field of computer security. In 2015, she completed her master thesis on NoSql Database Testing in Istanbul Kültür University and Ph.D thesis on Intrusion Detection Systems in Marmara University. During her

master studies, she worked on distributed databases. She worked as a Research Assistant with the Department of Mathematics and Computer Science, Istanbul Kültür University. She is working as a Assistant Professor as Marmara University in Computer Engineering Department. Her research interests include computer networks and security, machine learning, deep learning, cryptography, python programming, statistics, and graph theory.

Received: August 20, 2022; Accepted: December 01, 2022.

Computer Science and Information Systems 20(1):215–228 https://doi.org/10.2298/CSIS220825054T

Sternum Age Estimation with Dual Channel Fusion CNN Model

Fuat Türk[1,*], Mustafa Kaya[2], Burak Mert Akhan [2], Sümeyra Çayiröz[2], Erhan Ilgit[2]

[1]Computer Engineering, Cankiri Karatekin University, Cankiri, Turkey
fuatturk@karatekin.edu.tr
[2] Gazi University School of Medicine, Ankara, Turkey

Abstract. Although age determination by radiographs of the hand and wrist before the age of 18 is an area where there is a lot of radiological knowledge and many studies are carried out, studies on age determination for adults are limited. Studies on adult age determination through sternum multidetector computed tomography (MDCT) images using artificial intelligence algorithms are much fewer. The reason for the very few studies on adult age determination is that most of the changes observed in the human skeleton with age are outside the limits of what can be perceived by the human eye. In this context, with the dual-channel Convolutional Neural Network (CNN) we developed, we were able to predict the age groups defined as 20-35, 35-50, 51-65, and over 65 with 73% accuracy over sternum MDCT images. Our study shows that fusion modeling with dual-channel convolutional neural networks and using more than one image from the same patient is more successful. Fusion models will make adult age determination, which is often a problem in forensic medicine, more accurate.

Keywords: Sternum age; deep fusion CNN, CNN age estimation, dual channel fusion CNN, sternum with CNN.

1. Introduction

Classical age estimation methods usually involve procedures that are detected through images such as the face or finger-wrist bone. The basic principle is to detect the features/findings that occur with aging through images and to estimate age from these features/findings. General machine learning algorithms can be used to estimate age [1]. The accuracy of age estimation over images expressed by machine learning depends on manually designed features and learning algorithms used. A crucial point to note at this point is the fact that a person's biological and skeletal maturity is related to bone age rather than chronological age [2]. Bone age can be used to determine chronological age when information is not available in underdeveloped countries where the age determination of children is not registered [3]. In addition, bone age is used as an auxiliary element in the diagnosis of various diseases [4,5]. Despite the known importance of bone age, studies are mostly limited to the age of 18. The changes

* Corresponding author

observed in the skeletal system after the age of 18 are insufficient to make a clear age determination. It is not possible to accurately estimate the digit of a person's chronological age. Age determination in adults can be made with a wide range ranging from ten to twenty years. More precise data is needed in forensic cases where the determination of medical age estimation is extremely important. From this point of view, age determination through the sternum gains importance. However, the fact that there are very few differences in the sternum in individuals who have completed their adulthood, and the inability to make a precise estimation even from the point of view of the radiologist, have caused these studies to be avoided. Therefore, studies have been carried out mostly on height and gender estimation over the sternum [6].

Recently, with the rapid development of Convolutional Neural Networks (CNN), CNNs have been successful in classification problems consisting of close data and have brought the engineering and medical fields together at joint working points. . There are almost no studies on how to construct a highly accurate age estimation model from sternum MDCT images with deep learning methods. With the Dual Channel Fusion CNN (DCF-CNN) model we propose, we carry out the study by transferring coronal and sagittal MDCT images of the sternum to our network via two channels.

The summary of this study and its contribution to science is given below:

1. We offer the opportunity to examine deep learning methods by dividing an original, unused, sternum dataset into groups 20-35, 36-50, 51-65, and over 65 years old.
2. Classic fusion CNN models can have 2,3,4 channels, but they can work by giving the same image as input over and over. Here, we propose a new fusion dual channel model approach by giving coronal and sagittal MDCT images of the same patient as two inputs.
3. Based on the success of the proposed model even in cases where the sternum image differences are very small, we can say that high accuracy values can be achieved by incorporating images with multiple different inputs into the system separately in future studies. This inference is extremely important for difficult deep learning problems.
4. Considering that age estimation studies with the sternum are almost non-existent with deep learning, we should state that we have brought a new perspective with sternum images and directed future studies.
5. We performed the simultaneous extraction and merging of different input images of a system. We demonstrated the use of multiple deep learning models and hyperparameters during this process. As a result, we propose a wide application area by saving deep learning studies from monotonous models.

The study is organized as follows: Sect. 2 summarizes the Literatur survey on the subject. The methodology of the study is given in Sect. 3. The search results are shown in Sect. 4. and Sect. 5 summarizes the conclusion and future works of the study.

2. Literature Survey

Monum et al. reported that conventional radiographic assessment of ossification of the sternum and rib tips did not yield effective results for age estimation of cadavers. This study examined Computed Tomography (CT) images to determine age at death in the Japanese population. In the study performed on 320 chest plate images, the accuracy of

the model was tested on 26 male and 24 female subjects, and the accuracy in age-decad estimation was found to be 57.69% and 70.83%, respectively [7].

In the study by Singh and Pathak, 8 nonmetric features were examined on 343 sternum images collected from autopsy cases. They observed that the mesosternal foramen is present, especially in men and elderly subjects, and the frequency of lateral projection in the manubrium sterni decreases with aging. It has been suggested that the fusion of sternal elements (manubrium sterni, corpus sterni, and xiphoid process) shows a variable pattern and is therefore not a reliable criterion. Scores of nonmetric features were obtained to the correct gender category in 73.8% and the correct age decades in 70.0% in the logistic regression analysis [8].

Bacci et al. performed an anthropological study on the sternum to estimate the age at death. In this study they mentioned that the sternum is an overlooked element in terms of adult age estimation. Also, fusion phases of 461 sternums from a black South African population were observed to match the individual's actual age with different phases of synostosis of the manubriosternal and sternoxiphoidal connections. The results show that both young (25 years old) and elderly individuals can have the complete fusion of the sternum, while some sternums remain unfused throughout adult life. Overall, logistic regression results showed 62.5% (male = 63.9%; female = 61.8%) accuracy (62.5%) for age determination [9].

Mohammed Ali et al. carried out a study on age and gender estimation in the Egyptian population using MDCT images of the sternum. The validity of the logistic regression equation for gender estimation was calculated as accuracy (88.3%), specificity (90.5%) and sensitivity (85.7%). They stated that the manubriosternal and sternoxiphoidal joint fusions were highly variable according to age, and the general logistic regression results showed a low accuracy rate [10].

Silajiya et al. carried out a study for age estimation on sternum X-Ray images. The authors examined 109 sternum bones and evaluated the fusion of the manubrium sterni and xiphisternum with the sternum body by X-Ray. According to this study, the age of fusion of the xiphisterum and the sternum body in men is 42, and the age of fusion of the xiphisterum and sternum body in women is 44. In the male population, manubriosternal fusion begins at age 50 and is completed after age 59. It has been reported that cases with partial manubriosternal fusion in women increase after the age of 54, and complete union after the age of 64. However, these ages are not exact and belong to a small series studied [11].

Zhang et al. For age estimation based on sternum MDCT images, it examined images of 512 documented individuals (254 females and 258 males) aged 20 to 85 years. In this study, the amount of cartilage costal calcification was taken as a basis for age determination and the Gradient Boosting Regression (GBR), Support Vector Machine (SVM) and Decision Tree Regression (DTR) machine learning models were tested. Outcomes were estimated at 88% for men and 77% for women [12].

When the literature is evaluated as a whole, it is seen that the studies that are more successful in age estimation are those based on the amount of calcification in the cartilaginous component of the ribs. Studies based on the fusion of the manubrium sterni and processus xiphoideus with the corpus sterni were able to estimate age with less accuracy.

3. Methodology

3.1. Dataset and Image Preprocessing

The data set used in this study was meticulously prepared by Radiologist Doctor Mustafa Kaya and his team, with the approval of the ethics committee of Gazi University. MDCT images of the study were obtained from patients who underwent thoracic MDCT for any reason between 01/01/2021 and 30/06/2022 in the Department of Radiology, Faculty of Medicine, Gazi University. MDCT examinations were performed with a third generation 192 section, dual tube MDCT scanner (Somatom Force, Siemens Healthcare, Erlangen, Germany) using the following parameters. Tube output 120 kV, pitch 0.9, detector collimation 0.6 mm, reconstruction section thickness 1 mm. The HU value of the sternum medullary, the width of the joint space between the corpus sterni and the manubrium sterni, and the amount of calcification of the anterior cartiloginous components of the ribs 1-7 were evaluated. Patients over 20 years of age who had not undergone mediastinotomy were included in the study. Investigations with movement/respiratory artifacts, artifacts caused by metallic stabilization bodies belonging to previous operations, and artifacts of port or cardiac pacemakers were excluded from the evaluation. In the study, MDCT images of a total of 300 patients, 75 in each class, including four different age groups, 20-35, 36-50, 51-65, and over 65, were used. Information about data distribution is shown in Figure1.

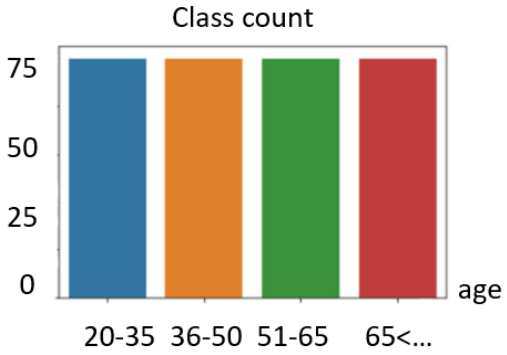


Fig. 1. Sternum dataset distribution graph.

The hypotheses for age determination are given in the following three titles.

1. As the age increases, the joint distance between the corpus sterni and the manubrium sterni narrows.
2. As the age increases, the medullary Hounsfield Unit (HU) density of the sternum decreases.
3. As the age increases, the amount of calcification of the cartilage costae, which forms the costosternal joints, increases.

To evaluate these hypotheses, the following measurements were made.

1. The joint distance between the corpus sterni and the manubrium sterni was measured from the 2D sagittal sternum MDCT image.

2. To rule out the effect of possible degenerative effects on the manubriosternal joint, medullary HU values were measured with a 5 mm diameter ROI (Region of interest) from the manubrium stern approximately 1 cm superior to the manubriosternal joint and from the corpus stern approximately 1 cm superior to the xiphoid process proximal line.
3. Multipanar reconstruction (MPR) was performed in the coronal plane, and Maximum Intensity Projection (MIP) images of 15 mm thickness were obtained from the coronal section including the second costosternal joint. It was aimed to evaluate the amount of calcification in the cartilage component of the bilateral ribs 1-7 on the defined MIP images. The measurements are explained in detail in Tables 1, 2, and 3.

Table 1. Values of manubriosternal joint space by age groups

Age range	Manubriosternal-joint spacing median value (mm)	Manubriosternal-joint spacing mean ± SD (mm)
20-35	2.4	2.1 ± 1.3
36-50	2.1	2.0 ± 1.1
51-65	2.0	2.0 ± 1.3
65<...	1.5	1.5 ± 1.3

Table 2. Manubrium sterni HU values

Age range	ManubriumHU (mean±SD)	ManubriumHU(median)	ManubriumHU percentile< %5	ManubriumHU percentile>%95
20-35	214 ± 75	218	73	333
36-50	161 ± 58	169	29	243
51-65	129 ± 66	129	5	239
65<...	99 ± 74	108	-27	227

Table 3. Korpus sterni HU values

Age range	KorpusHU (mean ±SD)	KorpusHU (median)	KorpusHU percentile <%5	ManubriumHU percentile >%95
20-35	128 ± 69	124	29	244
36-50	76 ± 57	82	-18	175
51-65	59 ± 69	54	-46	174
65<...	27 ± 62	33	-27	124

Existing images were edited with the help of the Pillow library within the Python Imaging Library by making Crop, Contrast, Brightness, and Filter improvements, respectively. Due to the high similarity between patient classes, additional image processing, and data augmentation techniques were not particularly preferred. The data obtained from the image processing results are shown in Figure 2.

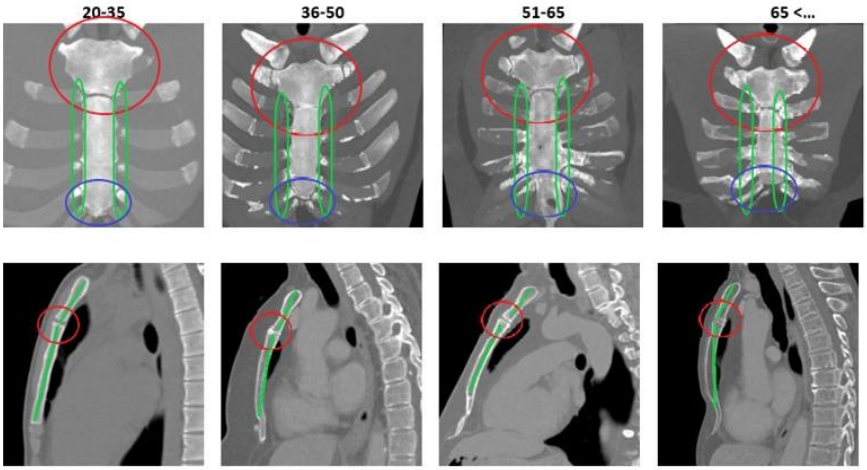


Fig. 2. Sternal age group images after image processing. Top row: Red ring: Configuration changes in the manubrium stern with age, green ellipse: Increasing calcification of the costosternal joints with age, Blue ring: Configuration changes observed in the xiphoid process. Bottom row: Manubriosternal narrowing and increase in subchondral sclerosis with age.

3.2. Classic CNN Classification

Classification with CNN is the most widely used architecture in difficult classification problems such as histopathological typing of tumors, in the differentiation of benign or malignant masses on radiological images [13-15]. Medical Imaging studies often attract attention as studies with limited data. To overcome the difficulties in such studies, a pre-trained network with a different dataset can be used by adapting it to the classification task. These networks, developed , produce promising results in various medical imaging studies but the results are always open to discussion. In this case, the best way to follow would be to design CNN models that can recognize the network model well even under difficult classification conditions. Figure 3 shows a basic CNN architecture.

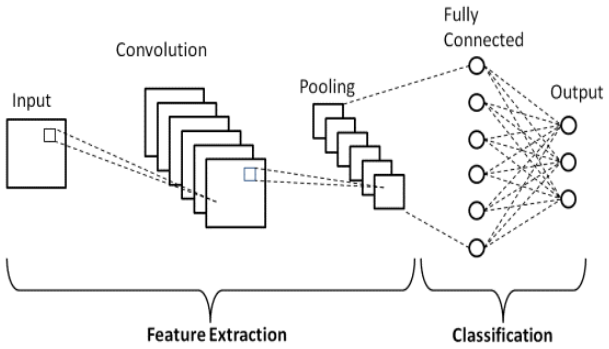


Fig. 3. Basic CNN architecture

The basic network structure starts with the image, where the image is filtered to include maps in the convolution layer. It is then condensed with pool layers. Subsequently, higher-level features are extracted in fully connected layers with a corresponding weight. These achievable features are processed to classify according to output categories corresponding to the original input. [16,17].

3.3. Proposed DCF-CNN architecture

With Fusion CNN architectures, it is aimed to learn architecture better by giving the same scene as an input again and again. The ultimate goal here can also be considered as capturing more features with fewer modalities [18,19].

Coding the basic information for the architectural structure allows a learning level without a large amount of data, so the use of a small-scale dataset provides successful results in terms of performance [20].

The fusion model design shown in the figure below is inspired by the Hybrid V-Net model and fusion deep neural network architectures [21]. The proposed Fusion based CNN architecture is shown in Figure 4.

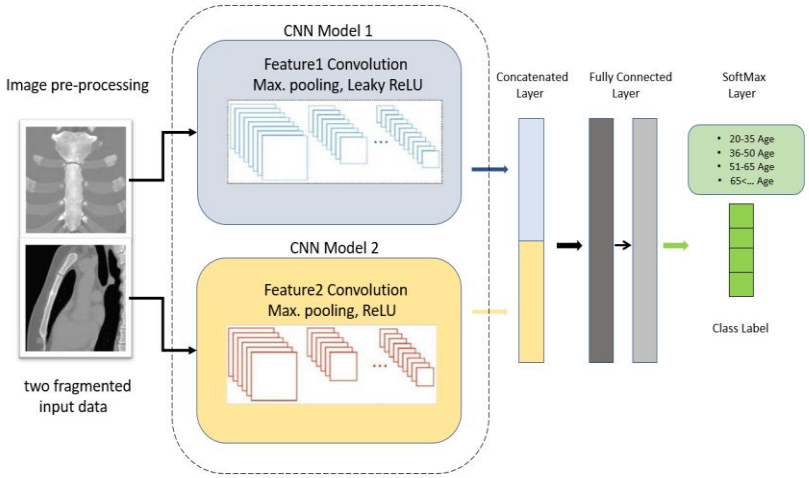


Fig. 4. Proposed DCF-CNN architecture

This work presents a DCF-CNN architecture to generate many local structures with various filter sizes. The input image size for the proposed DCF-CNN architecture is 150 × 150 × 3. The main entrance is divided into two roads. These include the CNN Model1 and CNN Model2 architectures. Login information is given parallel to these paths. The proposed architectures include convolutional filter size layers of 5 × 5 and 3 × 3, respectively. More local features are obtained with these filters of different sizes. In both Model paths, maximum pooling and Batch Normalization (BN) operations are performed after the convolution operations. One of the obvious differences between the models was the preferred activation functions. Leaky ReLU activation was used as it gave better results for CNN Model1. Thus, we can say that we can achieve more

optimization of the network by eliminating the negativities arising from large gradients. The Leaky ReLU activation Function is shown in Equation 1. Although the coefficient "a" expressed here is a small value, it was preferred as 0.01 for our model [22].

$$f(x) = \begin{cases} x & if\ x > 0, \\ ax & otherwise \end{cases} \quad \textbf{(1)}$$

For CNN Model2, the ReLU activation function is preferred and is shown in Equation 2. It should be noted that values less than zero are neglected and it is aimed to optimize accordingly [23].

$$f(x) = \max(0, x) \quad \textbf{(2)}$$

Another difference between the two architectures is the optimization method. Just as Adam is essentially an RMSprop with momentum, so Nadam is Adam with Nesterov momentum. Adam is an extension of gradient descent that adds the first and second moments of the gradient and automatically adapts a learning rate for each optimized parameter. Nadam, on the other hand, is a momentum extension where the update is performed using the gradient of the predicted update in the parameter instead of the actual current variable value. This has the effect of slowing down the search when optimizing is found rather than overdone in some cases [24,25]. For this reason, Adam for the first model and the Nadam optimizer for the second model was preferred because they gave better results. To better understand the implementation phase of the proposed models, the layer structures, activation functions, and dropout blocks are shown in Table 4. It should be noted here that two different architectural features were combined during classification after the extraction stage and turned into a fully connected single-layer structure. This structure allows the same input class to capture more features with more than one image, rather than combining two images with a single image.

Table 4. CNN Model1 and CNN Model2 structures

	CNN Model 1	CNN Model2
Input Image	150*150*3	150*150*3
Filters size	256/128/64/32/16	256/128/64/32/16
Kernel size	(5,5)/(5,5)/(3,3)/(3,3)/(3,3)	(5,5)/(5,5)/(3,3)/(3,3)/(3,3)
Pool. Layer	Max. Pool (2,2)	Max. Pool (2,2)
Optimizer	Adam Optimizer	Nadam Optimizer
Activation	Leaky ReLU	ReLU
Concat. Layer	256*2	
Fully Con.	512/256/4	
Output	Softmax, 4 class	

3.4. Performance Evaluation Metrics

Accuracy, Recall, Precision, and F-measure are the main metrics for measuring the performance of classification algorithms. Accuracy describes the overall performance of the proposed model and is calculated as shown in Equation 3 [26].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \textbf{(3)}$$

Precision shows how many of the Positive predicted values are Positive and is calculated as shown in Equation 4.

$$Precision = \frac{TP}{TP+FP} \quad \textbf{(4)}$$

A recall is a measure of how many transactions that should have been predicted as Positive were predicted as Positive and is calculated as shown in Equation 5.

$$Recall = \frac{TP}{TP+FN} \quad \textbf{(5)}$$

The F1-Score value shows the harmonic mean of the Precision and Recall values and is calculated as shown in Equation 6.

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall} \quad \textbf{(6)}$$

3.5. Dataset Distribution Operations

Figure 5 shows the block diagram of how the distribution is made. The point to note here is that two different images of patients with the same Identification Number (ID) are given sequentially as input to the two proposed models. To ensure the distribution, the ID information of the data set given to the first model was transferred to the second model.

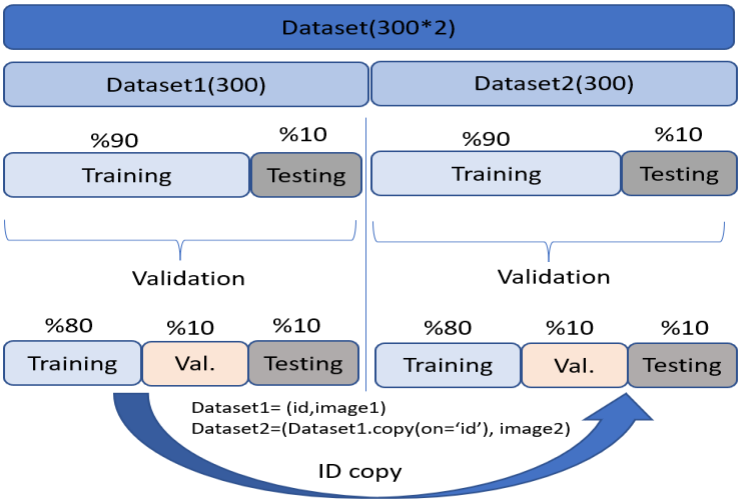


Fig. 5. Dataset Separation Process

4. Results and Discussion

The results shown in this section show the data obtained as a result of the train, test, and validation stages and the distribution of the data set. Fusion Model and CNN Models were run separately with 100 epochs. Each model is trained as designed with the same

hyperparameters. In Figure 6, accuracy and loss curves for the proposed Fusion CNN model after training and test results are shown. Based on these curves, we can easily say that the training and test results are consistent. However, the learning process could not go higher after a point.

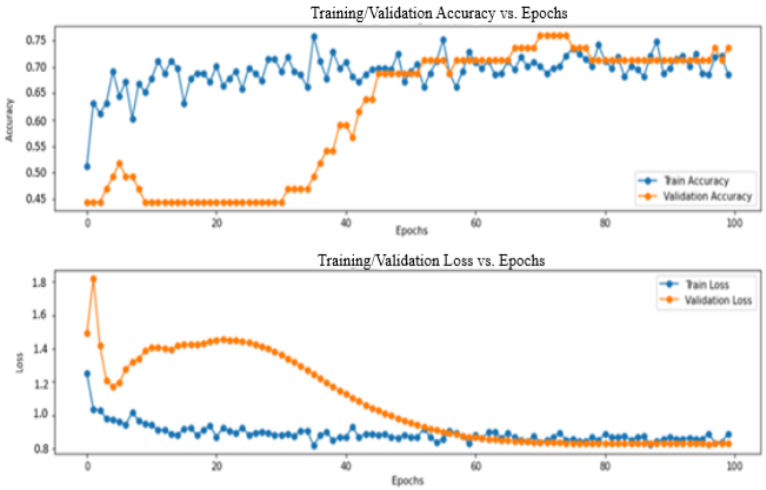


Fig. 6. Train and validation accuracy/loss graph for DCF-CNN

Table 5 shows the performance metrics after the test results of the values with all three models together. When the results were examined, CNN Model1 was run only on coronal images for the sternum and achieved 65% accuracy, while CNN Model2 only achieved 61% accuracy on sagittal images. The dual-channel Fusion CNN Model we recommend, on the other hand, took coronal and sagittal images together as input and achieved an accuracy rate of 73%.

Table 5. CNN models and proposed DCF-CNN of metrics values

Model	Accuracy	Precision	Recall	F1-Score
CNN Model1	0.65	0,66	0.67	0.66
CNN Model2	0.61	0.62	0.62	0,61
DCF-CNN Model	0.73	0.73	0.74	0.74

When the confusion matrix table in Figure 7 is examined, we can say that there are deviations in the results of the estimation values, especially due to the close differences between the ages of 36-50 and 51-65. However, the high similarities between the image classes and the existence of exceptional cases that can be observed in all age groups make age estimation very difficult for radiologists.

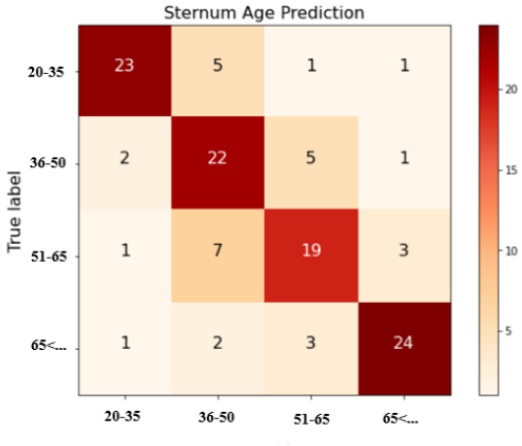


Fig. 7. Sternum age confusion matrix graph for DCF-CNN

In Figure 8, examples where the models are mislabeled for both images are given. Based on these images, it is seen that there are false labels not only in close classes but also in classes with high age differences. Looking at the images, it can be seen how difficult the classification task is. It is understood that only very small differences are decisive. We thought that the decrease in sternum medullary HU values with age might be a correction factor in the estimated age and actual age mismatches, but our results did not support this. Measured sternum meduller HU values were consistent with the estimated age, not the actual age. This made us think that the osteoporotic process in the sternum that develops with age is not a parameter independent of changes in the shape of the sternum bone or calcifications in the cartilage costa. The osteoporotic process is almost always related to bone shape changes with age, and probably the main cause of bone shape changes is the osteoporotic process. In addition, meduller HU measurements on the sternum for age groups are valuable and HU values below 5% percentile can be used as reference values for the diagnosis of osteoporosis.

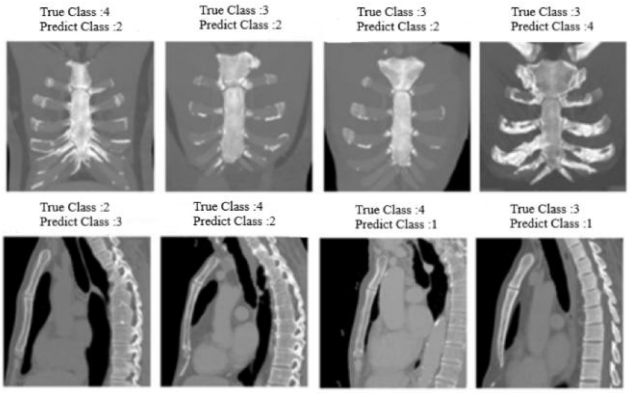


Fig. 8. Sternum mislabel images, Class 1: 20-35 age, Class 2: 36-50 age, Class 3: 51-65 age, Class 4 : 65<… (DCF-CNN test result)

The most successful study for age determination on sternum MDCT images is the study published by Zhang et al. in 2018 [12]. Although manubriosternal joint space and sternum medullary HU values were also included in the evaluation for a more precise age determination in our study, we could not determine age with precision in this study. The reason for this may be that individuals with chronic diseases that will affect age determination were not excluded from our dataset. In addition, while Zhang et al used the images obtained with the volume rendering technique (VRT) for age determination, we preferred 2D MIP images. With a high probability, VRT images are more successful in age determination than 2D MIP images.

5. Conclusions

Sternum age estimation studies carried out to date include classical biostatistics studies based on logistic regression analyzes based on the human eye, or machine learning studies that do not suggest a new model. With the DCF-CNN model we proposed, we were able to predict with 73% accuracy the original data set, which we divided into four diverse groups (with a 15-year interval) on a challenging subject with natural limitations such as adult age determination. The fusion modeling, we developed on two separate images of the sternum in our study increased our success compared to the predictions made with a single image. Based on this situation, we present a new approach model that will make a difference, especially in medical images. We can state that the success of the system can be increased by using images of the same patient with unique features related to each other. Even other than the medical image, two or more images with the same features can be passed through multiple channels and the accuracy can be increased by combining common features. In this context, we can say that more efficient results can be obtained by using different optimization methods, activation functions and hyperparameters at the same time.

As a future perspective, more advanced fusion models can be used on other bones used in adult age determination, especially pelvis bones, and studies are needed in this direction.

References

1. Xing, J., Li, K., Hu, W., et al. Diagnosing deep learning models for high accuracy age estimation from a single image. Pattern Recognition, Vol 66, 106–116. (2017) https://doi.org/10.1016/J.PATCOG.2017.01.005
2. Mirwald, RL., Dominic, A., Baxter-Jones G., Bailey DA. An assessment of maturity from anthropometric measurements Pediatric Bone Mineral Accrual Study View project Growth and Maturation in Sport and Exercise View project (2002) https://doi.org/10.1097/00005768-200204000-00020
3. Setel, P.W., Macfarlane, S.B., Szreter, S, et al. Who Counts? 1 A scandal of invisibility: making everyone count by counting everyone. Vol 370, 1569–77, (2007). https://doi.org/10.1016/S0140
4. Satoh, M. Bone age: assessment methods and clinical applications. Clinical Pediatric Endocrinology, Vol 24, 143–152, (2015). https://doi.org/10.1297/CPE.24.143

5. Nguyen, Q.H, Nguyen, B.P, Nguyen, M.T., et al. Bone age assessment and sex determination using transfer learning. Expert Systems with Applications, Vol 200, 116926, (2022). https://doi.org/10.1016/J.ESWA.2022.116926
6. Saraf, A., Kanchan, T., Krishan, K., et al. Estimation of stature from sternum-Exploring the quadratic models, (2018). https://doi.org/10.1016/j.jflm.2018.04.004
7. Monum, T., Makino, Y, Prasitwattanaseree ,S, et al. Age estimation from ossification of the sternum and true ribs using 3D post-mortem CT images in a Japanese population, (2019). https://doi.org/10.1016/j.legalmed.2019.101663
8. Singh,J., Pathak, RK. Forensic Anthropology Population Data Sex and age-related non-metric variation of the human sternum in a Northwest Indian postmortem sample: A pilot study, (2013). https://doi.org/10.1016/j.forsciint.2013.02.002
9. Bacci, N, Nchabeleng, EK., Billings, BK Forensic Anthropology Population Data Forensic age-at-death estimation from the sternum in a black South African population, (2017). https://doi.org/10.1016/j.forsciint.2017.11.002
10. Ismail, M., Ali, M., Mosallam, W., et al Sternum as an indicator for sex and age estimation using multidetector computed tomography in an Egyptian population. Forensic Imaging Vol, 26, 200457, (2021). https://doi.org/10.1016/j.fri.2021.200457
11. Radiological Age Estimation From Sternum.: EBSCOhost. Available: https://web.p.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=0&sid=0b6e5d68-86f6-44f4-aaa6-9b3a8c6e9271%40redis, (2022).
12. Zhang, K, Fan, F, Tu, M, et al. The role of multislice computed tomography of the costal cartilage in adult age estimation. International Journal of Legal Medicine, Vol 132, 791–798, (2018). https://doi.org/10.1007/S00414-017-1646-Y/TABLES/3
13. Abdullahi, A., Bawazeer, K., Alotaibai, S, et al. Pretrained Convolutional Neural Networks for Cancer Genome Classification; Pretrained Convolutional Neural Networks for Cancer Genome Classification, (2020).
14. Sun, Y., Zhu, S., Ma, K, et al. Identification of 12 cancer types through genome deep learning. Scientific Reports 2019, Vol 9, No. 1, 1–9, (2019). https://doi.org/10.1038/s41598-019-53989-3
15. Li, S., Xu, P., Li, B, et al Predicting lung nodule malignancies by combining deep convolutional neural network and handcrafted features. Physics in Medicine & Biology Vol. 64, 175012, (2019).. https://doi.org/10.1088/1361-6560/AB326A
16. Ardila, D., Kiraly, AP., Bharadwaj S, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography, (2019). https://doi.org/10.1038/s41591-019-0447-x
17. Mutasa, S., Sun, S, Ha, R. Understanding artificial intelligence-based radiology studies: CNN architecture. Clinical Imaging, Vol, 80, 72–76, (2021). https://doi.org/10.1016/J.CLINIMAG.2021.06.033
18. Türk, F., Lüy, M., Barışçı, N. Kidney and Renal Tumor Segmentation Using a Hybrid V-Net-Based Model. Mathematics 2020, Vol 8, 1772 (2020). https://doi.org/10.3390/MATH8101772
19. Zhang, Y., Morel, O., Blanchon, M., et al. Exploration of Deep Learning-based Multimodal Fusion for Semantic Road Scene Segmentation. VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Vol 5, 336–343, (2019). https://doi.org/10.5220/0007360403360343
20. Mezina, I., Qingjie, L., Yunhong, W. Very High-Resolution Images Classification by Fusing Deep Convolutional Neural Networks, The 5th International Conference on Advanced Computer Science Applications and Technologies ,(2017). https://doi.org/10.23977/ACSAT.2017.1022
21. Tenenbaum, J.B, Freeman, W.T. Separating style and content with bilinear models. Neural Computation, Vol, 12, 1247–1283, (2000). https://doi.org/10.1162/089976600300015349
22. Maniatopoulos, A., Mitianoudis, N. Learnable Leaky ReLU (LeLeLU): An Alternative Accuracy-Optimized Activation Function, (2021). https://doi.org/10.3390/info12120513

23. Eckle, K., Schmidt-Hieber, J. A comparison of deep networks with ReLU activation function and linear spline-type methods. Neural Networks, Vol 110, 232–242, (2019). https://doi.org/10.1016/J.NEUNET.2018.11.005
24. Khaire, U.M., Dhanalakshmi, R. High-dimensional microarray dataset classification using an improved adam optimizer (Adam). Journal of Ambient Intelligence and Humanized Computing, Vol 11, 5187–5204, (2020). https://doi.org/10.1007/s12652-020-01832-3
25. Sharma, J., Soni, S., Paliwal, P., et al. A novel long-term solar photovoltaic power forecasting approach using LSTM with Nadam optimizer: A case study of India, (2022). https://doi.org/10.1002/ese3.1178
26. Nisha, J.P., Gopi, V., Palanisamy, P. Automated colorectal polyp detection based on image enhancement and dual-path CNN architecture. Biomedical Signal Processing and Control Vol, 73, 103465, (2022). https://doi.org/10.1016/J.BSPC.2021.103465

Fuat Türk B.Sc. graduated from Gazi University Computer Technology Department in 2004. He received his PhD degree from K ırı kkale University Computer Engineering Department in 2020. He is working as an Assistant Professor in the Department of Computer Engineering at Çankırı Karatekin Univers ity. His areas of interest are biomedical informatics, artificial intelligence and software.

Mustafa Kaya, MD. graduated from Ankara University Medical School in 1996. He finished radiology residency at the Ankara University Medical School in 2002. He is working as Lecturer in the Department of Radiology at Gazi University Medical School. He is an expert in thoracic and abdominal Multi detector CT imaging.

Burak Mert Akhan, MD. graduated from Hacettepe University, Faculty of Medicine in 2017. She is working as a Resident Doctor in the Department of Radiology at Gazi University Faculty of Medicine since 2019. His areas of interest are interventional radiology and artificial intelligence.

Sumeyra Çayıröz, MD. graduated from Istanbul University, Istanbul Faculty of Medicine in 2020. She is working as a Resident Doctor in the Depa rtment of Radiology at Gazi University Faculty of Medicine since 2021.

Erhan Ilgıt MD. graduated from Ankara University Medical School in 1986. He finished Radiology residency at the Gazi University Medical School in 1991. He became associate professor in 1993 and full professor in 2011. He serves as the Director of the Radiology D epartment at Gazi University Medical School.

Received: August 25, 2022; Accepted: December 15, 2022.

Computer Science and Information Systems 20(1):229–249 https://doi.org/10.2298/CSIS211112048P

Self-Service Kits to Scale Knowledge to Autonomous Teams – Concept, Application and Limitations

Alexander Poth[1], Mario Kottke[1], and Andreas Riel[2]

[1] Volkswagen AG, Berliner Ring 2
D-38436 Wolfsburg, Germany
{alexander.poth, mario.kottke}@volkswagen.de
[2] Grenoble Alpes University, Grenoble INP, G-SCOP Laboratory
F-38031 Grenoble Cedex 1, France
andreas.riel@grenoble-inp.fr

Abstract. In large organizations, it is not trivial to spread knowledge to all teams. Often, individual teams need to handle similar topics and re-invent the wheel. Another scenario is that a group of people with a common role (for example "guild" in Spotify model) has to distill their practices to make them shareable. Trainings should have empower participants so to apply the learnings easily in their daily businesses. To realize this, the proposed Self-Service Kit (SSK) approach can be used in the context of a holistic methodology that fosters team autonomy while leveraging knowledge spread and sharing throughout a large organization. Such a methodology is presented and instantiated in an enterprise context in facing the mentioned challenges.

Keywords: computer science, information systems, agile, learning organization, efiS® framework.

1. Introduction

The VUCA challenges (Vulnerability, Uncertainty, Complexity and Ambiguity) [1] of our modern economic, ecologic and social context makes that more and more enterprises are confronted with the need to initiate and drive forward their agile transformations in order to be able to control change and react to change as fast as possible. Although agile frameworks, methods and practices have been largely published and widely adopted already, the company-specific introduction and scaling of whichever framework is always a huge challenge requiring enormous investments in terms of time, effort, and more often than not, also failure. One of the key reasons at the origin of this is the difficulty of making agile change propagate in the organization at a vast scale, in particular in large organizational settings. Departmental borders and, more generally, organizational silos [2] in classical hierarchical corporate organizations lead to the fact that agile transformation efforts frequently fail reaching the critical mass of entities required to make the change happen on a large scale. Overcoming these boundaries requires specific strategies that typically rely on the presence and active involvement of agile trainers and coaches in a huge number of departments [3]. These coaches have the mission to introduce agile mindsets and practices in a way that is most appropriate to the

current organizational culture and practices of that very unit. While this approach is mostly effective on a local (department/team) scale, it often fails to efficiently scale up higher, cross-departmental levels. Furthermore, the number of coaches required in order to adopt this approach in large organizations is in general hardly compatible (at least limited by the scalability of the coaches) with the resources that companies can or want to invest in transformation efforts, which will translate to tangible economic benefits only in the mid- and long-term [4].

This article investigates an alternative approach to this challenge that relies on team autonomy at its ultimate level for scaling. Its core concept is to provide teams with digital agile "transition kits" that they have developed and keep extending and improving themselves [5]. Agile coaches accompany this process, however at a far more distant and selective scale than in the classical approach. This concept essentially relies on teams knowing themselves best their needs and desired interpretation of agile frameworks and methods. Only initially coaches can just support them in identifying those, and they push for the creation and evolution of electronic agile self-service kits (SSKs) inspired by the concrete needs of the product/service teams, and moderated by experienced competence and knowledge leaders in the organization. The authors put this SSK-driven approach in the context of the large-scale [6] agile setups especially the efiS® framework they created and implemented in the Volkswagen Group IT. With respect to size, global outreach, cultural mindset, and diversity this global enterprise providing several IT services to the Volkswagen AG poses challenges to agile transformation that are largely comparable to those companies in other sectors are facing when adopting agile practices. The key point this paper is making lies in the demonstration of the potential of the SSK-based agile transformation approach through concrete success stories at the Volkswagen AG.

To this aim, this article is organized as follows: section 2 elaborates on related work in the field of large-scale agile frameworks and organizational learning-based transformation approaches. Section 3 explains the methodology applied to coming up with efiS® and the SSK's constituting this framework. Section 4 presents four use cases implemented at the Volkswagen AG in order to prove the practical relevance, efficiency and effectivity of SSKs in the large-scale setting of the Volkswagen Group IT. Section 5 points out the limitations of the SSK approach and its initial instantiations at the Volkswagen Group IT most notably with respect to the validity and transferability of the insights and evaluation results to other corporate environments. Section 6 concludes by summarizing the key contributions that this paper attempts to make both to the academic and practical community and includes the outlook which describes next steps.

2. Related Work

Works on the challenges that traditional organizations are confronted with when they undergo agile transformation are numerous like [7]. Many of those challenges rely on the fact that in agile environments, people are key success factors for the outcomes [8]. Taking this essential aspect into account adequately grows in difficulty with the size of the organization. In [9] we investigate widely adopted large-scale agile frameworks such as SAFe®, LeSS, Scrum@Scale™, and others, with respect to the ways they integrate

facilities for team autonomy, self-governance, and knowledge scaling. Our conclusion from this investigation was that although all three aspects are addressed to some extent in each framework, there is no consistent focus on autonomy in any of them, and self-service approaches are hardly addressed. However, as autonomous teams are key elements in agile organizations for product development [10], self-service approaches for knowledge scaling in large organizations is a subject that has been identified as relevant in several contexts, and considered as non-trivial [11]. Among them, Problem-Based Learning (PBL) [12] relies on collecting and documenting particular problems and their solution approaches. If applied by the teams themselves, and enriched by guidance [13], the gathered knowledge can be compiled in SSK's.

Given the fact that in agile teams, the social component is in general well developed [14], knowledge sharing support through social network sites [15] and communities of practice (CoP) [16] are especially appropriate, since agile teams are based on the specific competencies of their individuals [17]. Furthermore, autonomy and self-organizing teams come together and need cross-functionality, which is based on sharing of knowledge [18] that is available both within and outside the teams.

Another important driver for self-service approaches is the increasing geographical distribution of agile organizations [19] and teleworking [20]. Self-paced distant learning based on goal-based scenarios [21] is more and more relevant, combined with blended learning [22]. Self-paced and asynchronous formats [23][24] are extended by synchronous online formats for facilitating tight collaboration [25].

In corporate environments, Web-based training (WBT) approaches are increasingly common [26]. They transfer information; however they fail guiding knowledge spread. Giving an adequate design context to the knowledge spread approach is therefore a key success factor [23], since it is a prerequisite to enable learners to re-contextualize the provided learning content. Labs used for practical instruction-guided training combined with e-learning constitute a blended learning method [27]. Such labs provide guidance, however they represent only a limited set of real-world scenarios. SSK's inspired by and deployed in real-world scenarios and created by Communities of Practice can overcome these limitations, however care has to be taken in the problem identification and solution guidance to avoid significant failures [28] leading to harm [29] either by misguidance, misuse or even by accident.

To the best of our knowledge, combining the knowledge concentration and dynamics of Communities of Practice with the rapid and convenient spread of web-based of corporate SSK's has not been investigated in literature so far, although it promises interesting scaling power.

3. Methods

3.1. Research Goals

The goal of this research is to come up with an autonomy-focused approach to facilitate agile transitions for large-scale enterprise settings without the need of extensive team

coaching. The focus of the proposed approach is to fosters team autonomy to a maximum while assuring the fast and deep propagation of agile mindsets and practices across the entire company. In this context, knowledge scaling is an important building block which is addressed by SSK's. To shape the term transition and transformation in this context, we define transformation with a defined target state which can be reached and then it is done. A transition is more used in a Cybernetics [30] thinking. In this case agility is like a variable and the variable can change the state to any value. There is no final state defined – it leads to a continuous step-by-step transition to more agility. Depending on the teams transition progress they have an agile maturity which leads to the ability to act autonomous. However, the team specific "maximum" depends on the team maturity and their products [31]. The autonomy enables the teams to define and shape their delivery procedures within their value streams – the autonomy leads to createablity [32]. For this, the knowledge-scaling focused SSK approach shall be extended to be easy to use, adaptable to the specific needs and maturities of organizational units and their teams, and be deployable in any agile or non-agile framework. Particular design constraints are given by the need for the minimization of frequent and broad agile coach interventions in individual teams for the efficiency reasons explained in the introduction to this article. Furthermore, the approach shall be based on the SSK paradigm defined in [33]. The concept of the SSK to offer knowledge like learnings which become knowledge or later on specific Intellectual Property (IP) of an organization to the autonomous teams. The instantiation of this knowledge can differ from the information which is shared. To support this a SSK defines a kind of common structure and a basic set of relevant meta data. Furthermore, the detailed SSK implementation is open to address the consumers behavior. To foster loosely coupled autonomous teams the knowledge have to be available in an asynchronous way – else a more intensive coupling for synchronization between the teams for knowledge transfer is needed. With the SSK concept teams have to possibility to become prosumers. In their early stages with lower maturity they consume the SSK with the knowledge or IP shared from others. With their growing maturity teams can become to producers of knowledge with their learnings. They can extend existing SSKs are build new SSKs if they have opened a new knowledge domain. The objective of SSKs is that they are like 24*7 available, independent and self-contained knowledge units.

Based on the fundamental research objectives specifically defined for the SSK transition approach in [34], we aim at building, distributing and relating SSKs to each other such that they help constitute a holistic aid in adopting (agile) practices for specific topics and subject areas that are of particular importance for the specific organization.

Based on the identified aspects and observations requirements for a holistic SSK approach can be derived. More specifically, we define the Key Requirements (KR) to the SSK-based agile transition framework as follows:

1. SSKs as defined in [33] shall be applicable for "stand-alone" knowledge topics in a way that topic experts initiate the SSK creation and foster their continuous improvement and share the learnings with other teams. SSK application through teams shall happen in a "team-pull", rather than in a "coach-push" effort wherever possible. These SSKs shall drive the building up of knowledge in a particular subject area on a large organization scale.

2. SSKs shall also serve as a means of delivery for building blocks of holistic agile frameworks as efiS®. Compared to 1), this requirement implies an increased need for complementary design of individual SSK's and consistency among them.
3. SSKs shall also be designed in a way that hands-on trainings can be built from them. The integration of agile transition means (such as SSKs in this case) in trainings not only fosters efficiency, but it also allows collecting direct feedbacks on the understandability, relevance and needs for improvement for these SSKs.
4. Given 1)-3), SSKs shall also serve as part of knowledge management instruments within the organization. If applied on a large scale on several core knowledge topics of the company in bidirectional form (i.e., employees get knowledge from them, but also feed in new knowledge into them), they can be seen as a living representation of the diversity and actuality of core knowledge elements. Knowing about the actuality requires life cycle management facilities built into SSKs.

3.2. Research Context Design

The Volkswagen Group IT is the field of research. The Group IT is like a virtual organization which includes the IT capabilities of the brands. Additionally, the brands act autonomous within their business areas. So, it is possible to group e.g. specific technology expertise logical in communities. However, all legal entities can organize and decide about their projects etc. within their autonomy. Agile transitions are made in different ways. First it can come bottom up – a team decides to work aligned to an agile approach they find suitable for their setup and context. In this case the team can self-organize this transition journey or get help by agile coaches. The second approach is more top-down. In this case e.g. a program management decides to establish an agile setup. Here, the approach is predefined and the teams have to align with the program management decision. Often facilitation is given by the program management to the teams to become part or the programs transition journey. Coaches often facilitate transformations to reach pre-defined goals in a time-framed period. Then the enabled teams go ahead autonomous within their agile transition. As the demand about specific agile knowledge can vary significantly over time, approaches are needed to deliver the demanded knowledge just-in-time. Human coaches are not able to scale in/out with the demand-peaks. To handle the volatile demand other approaches are needed. One facilitation for the dynamic demands are self-services.

This complex and heterogeneous research environment makes it difficult to define synthetic investigation setups which are representative to ensure a-priori that outcomes can be transferred to the real world. This leads to selecting a research approach which can handle real-world environments and deliver its outcomes direct into the demanding organization or team. The objective is to work directly with a set of users to ensure that all real-world effects are addressed and the outcome produces value for them. In a second step, the "general availability" scales the outcome.

3.3. Methodology

The methodology fits to the research environment and the research question (RQ). The RQ is how SSKs concept can be used on a broader base to address typical scenarios of knowledge sharing in enterprises? This leads to the hypothesis that the current SSK concept needs selected extensions to fit the set of typical knowledge sharing use cases of enterprises. All new concepts to extend the existing SSK approach are introduced in this work. The existing capabilities and features of SSKs are described in [20]. To identify the use cases different experts from the Agile Center of Excellence (ACE) and Test & Quality Assurance (TQA) were requested to show typical cases in which knowledge is needed. This can be used to derive the use cases and specific demands about SSKs. Consequently, the methodology chosen to develop the efiS® framework – an agile framework for enterprises - and one of its core building block SSK [33] is Action Research (AR) [35]. Oriented on the AR approach the specific AC for this work is derived. The experimental field included several organizational entities within the Volkswagen Group IT. In their double-roles as researchers and agile coaches, the authors have chosen the approach of semi-formal interviews with different product/service teams in order to accompany them on their ways of creating and instantiating SSKs for different purposes linked with the agile transition. The extended SSK concept was developed within a working group of ACE and TQA members. This was useful also to get fast feedbacks in the daily work of coaching and facilitation of teams and organization. The prosumer approach of SSKs is based on the collaborative agile mindset by design. In order to assure the complementary nature of the SSKs, an SSK assured the uniform core structure and the facilitation of SSK use and understanding. An important further structural element is the common life-cycle developed for SSKs, which comprises a fundamental state-model which can be adapted and extended if needed. Some meta-data attributes were also defined from the outset facilitating the management, structuring, and linking of SSKs. Some of these attributes are mandatory (e.g. last revision, the name of the owner/maintainer), others are optional and/or context-specific (e.g. the maturity of the content, dependencies of documents, other SSKs etc.).

The action research approach within the corporate environment of the Volkswagen Group IT intrinsically included the validation of the developed SSK approach. The latter's more general validity and applicability has been considered consistently through all phases by making sure that any concepts chosen to build the efiS® framework and its SSKs are not specific neither to the company nor to the industrial sector it is active in. The validation measures and observations were conducted in real product or service environments. Figure 1 shows the flow which combines the research (analyze potentials and evaluate solution) and development (identify scope and develop solution) of the SSK concept. The iterative approach is started by the initial demand and terminated by the released solution. The termination condition for this specific AR flow is the "sufficient fit" between demand and the current solution, by identifying no further significant improvement potential which justifies another development iteration. All steps of the flow for the SSK development are mainly conducted by the SSK working group, only the step evaluation solution was mainly done by teams who have to use the potential released solution in the future. In this step the working group was "observer" to learn also about usage, adaption issues etc.

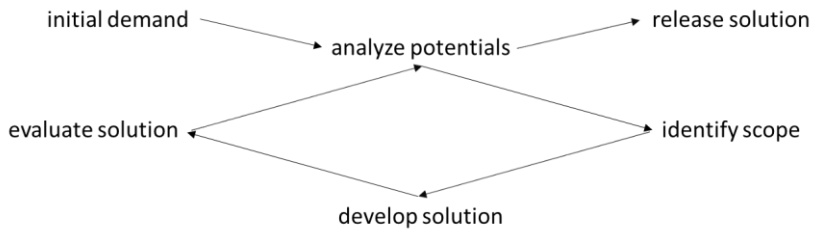


Fig. 1. The iterative research and development flow with the entry and exit points.

3.4. Design of model capabilities

Derived Use Cases. In order to meet the research goals, we decided to define representative use cases for SSKs with a high practical relevance. This helps to avoid over-engineering of the SSK approach extension for scaling agile practices and knowledge. Such use cases have been derived from typical enterprise settings. The following use cases are addressed:

- Stand-alone SSK (Research Goal 1): to package knowledge of one topic to make it available to other people to address KR1
- Network of SSKs (Research Goal 2): to relate SSKs to address KR2 with each other in the context of
  - Frameworks with more or less dependencies between their "building-blocks": establish a harmonized set of SSKs for topic-specific building blocks to build complex frameworks to realize additional effects beyond the specific building blocks on the system level of the framework; this requires a harmonized release of all involved SSK of the network for assuring consistency.
  - Knowledge base with maturity "indicator" for the practice: extends the life-cycle of SSK to additional attributes. Beyond the versions semantic aspects to manage for example maturity of the SSK content.
- SSK as training "take-away" package (Research Goal 3): bundle specific topics to work with them in the training on examples and later on re-use them in the real-world context to address KR3.

All 3 research goals together contribute to address KR4. These use cases require additional attributes and meta data for SSKs to model the demanded characteristics for the enterprise setting. The additional information attributes have to be designed to be optional or additional to keep the compatibility to the SSK approach v1.0 as proposed in [33]. The additional attributes define v1.1 with the extended modeling capabilities. For the attributes and their values context specific options have to be identified like pre-defined values usable as tags or references like links.

Demanded as optional meta data, the dependency and the maturity description is needed. To keep the application of the SSK approach as simple and intuitive as possible, the design results of the v1.1 have to be lean and easy to understand.

Dependency handling. These data have to be designed to handle references to other SSKs. Three different types of dependencies are needed to handle a network of dependent SSKs:

- No dependency – no action needed (trivial case).
- Uni-directional dependency – needs checks to evaluate the "source" update to a new version.
- Bi-directional dependency – on every update of both sides, a check is needed.

Existing formal approach for dependency modeling like [36]. However, typical users do not have the formal modeling approach in mind, if they want to express a relation. This leads to investigating more intuitive approaches. Mark the dependency on the SSK visible for everybody to make it transparent. This can be realized as an attribute to the SSK metadata like for e.g. "dependency to:". Then, the one or more dependencies are listed. An intuitive way to do this are e.g. hyper-links, which are well-know and understood by mostly all people.

In a second step, identify the semantic level of the dependency to avoid SSK dependencies where possible by smart SSK design. A strategy is the indication of a semantic dependency, but avoid to have content which is subject to change in the future. The result is that the overall network is described, but changes driven by the dependency are rare or completely avoidable. These are weak dependencies. An example is to have a dependency identified and then only link to the "master description" and avoid partly repletion of the master description as a copy which has to reworked in case of changes in the master description.

In case of established sematic dependencies, ensure validations for new versions to enable blocked releases of the dependent SSKs if needed. Blocked release packages have strong dependencies. In case of semantic dependency to the content of the master description the version of the master description has to be part of link to avoid inconsistency. To make this kind of dependency transparent, it is useful to add the version of the dependent SSK to the dependency attribute. So, everybody can independently identify on their own cases where potentially inconsistent versions are used. As a positive side effect of this information, it helps to initiate updates to newer versions of dependent SSKs, because users can have local outdated copies of SSKs.

The presented approach with links for dependency modeling and optional version binding offers an easy to understand approach with a lot of options to model different kind of dependencies.

However, this dependency handling approach does not help to identify dependencies in a (semi-)automated way – this is still the work and responsibility of the SSK team.

Maturity Description. Modeling of a maturity life-cycle for SSKs can be realized by a state diagram. For most cases the state diagram is straight forward, like: concept → evaluated → established → decommissioned. Any SSK version can have only one state. This makes it easy to realize, each state as a "tag" in the simplest modeling way. The current maturity state respective the representation as tag is associated with the SSK. The approach of using "tags" has the benefit that it also can be used to model more complex maturity life-cycles. A more complex case is for example the case in which different domains have to be represented by a set of states like: evaluated in domain x, established in domain y and decommissioned in domain z. The important point is that each domain can model its life-cycle flow independent with the tags. This can lead to n domains * m states (Figure 2). In the worst-case, n*m tags have to be assigned to the

maturity attribute. However, in practice, this is untypical (due to a very simple design of the SSK maturity description) and is only a theoretical issue to have a large amount of tags. The approach to describe the maturity with tags is useful for a trivial maturity life-cycle staging, but it also allows building more complex stage-models if needed. This makes it adaptable to many application scenarios.

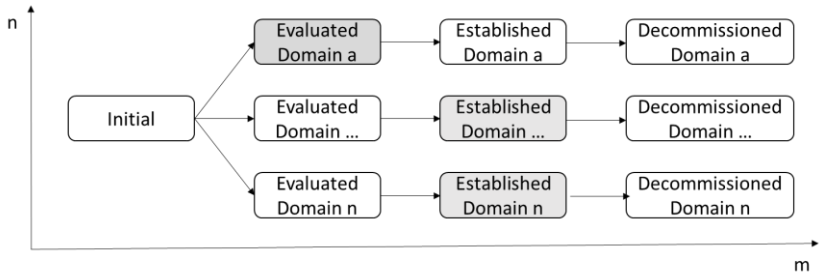


Fig. 2 Example of a complex set of tags to model maturity with business domain orientation – each domain can have an individual usage "state" represented by the grey tagging.

4. Evaluation

The evaluation presents examples from the AR setup of different use cases of the Volkswagen Group IT to demonstrate the potentials and applications of SSKs. The use case specific demand to SSKs for the new proposed meta data attributes are presented, too. The four case studies are structured in a way as to explain the objectives of each use case, and the role of SSKs in their implementation. Then, the learnings and results for each use case are presented. While we did not undertake agile maturity evaluations at the beginning and end of a certain evaluation period, we directed the evaluation focus on how the SSK's were perceived, accepted, and integrated by the teams under evaluation. From that perspective, agile teams reporting a felt usefulness of the SSK's, as well as contributing to the latter's extension and improvement, leads to a positive evaluation. Each sub-chapter addresses one of the four KR and all together contribute to the RQ. Each sub-chapter presents the needed extensions to reach its KR. This confirms the hypothesis, that with only a few selected extensions use cases in real enterprise setups can be solved with the SSK concept.

4.1. Case study A: "Stand-alone" knowledge topic

Use Case: Use the SSK approach to deliver a topic like Quality Assurance (QA) for Machine Learning (ML). The presented evAIa approach [34] is a representative example for this use case. The evAIa approach is used to identify adequate safeguarding actions for ML based products and services. This identification should be realized by autonomous teams – this motivates an SSK for the evAIa approach. The SSK can be

built on the basic structure initially presented in [33]. No additional attributes are needed. Figure 3 shows a part – an overview - of the SSK. The content of the SSK is directly related to the content of the evAIa publication. A special aspect is the reference to the PQR approach [37] to identify product specific quality risks. This is an additional aid to ensure the right focus for the evAIa application. It is referenced as a kind of useful pre-condition. However, there is no direct dependency to the PQR SSK established. This example shows how two SSKs one for the evAIa and one for the PQR method can have a relationship by reference without having a direct dependency.

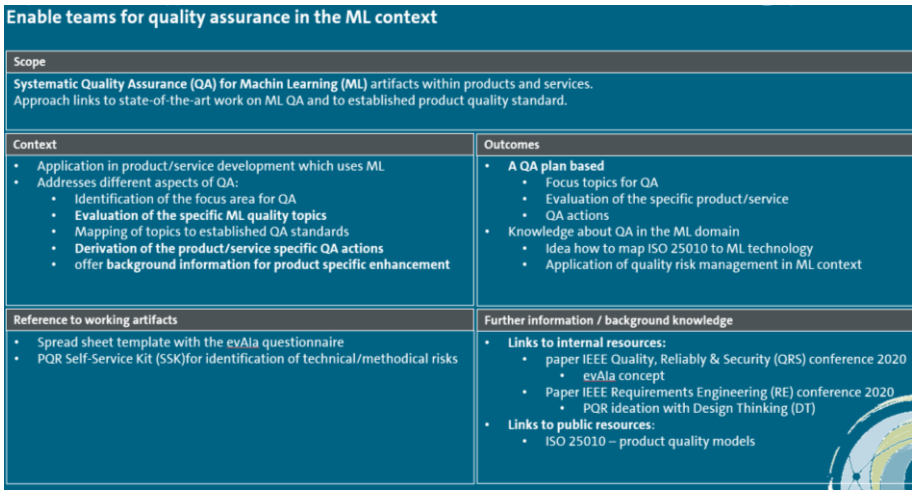


Fig. 3. Extract from the evAIa approach SSK.

Setup: The evAIa approach was developed in a cross-brand team from Audi, Cariad and Volkswagen [34]. By design the documentation was made with the SSK concept. The evaluation took place in ML teams of different brands. The feedbacks were used to enhance the SSK and the questionnaire of evAIa. The authors of the SSK observed the usage by the teams. The iteration phase terminates as no significant improvement potentials were identified. This termination criteria is based on the feedback and the observations which indicate an intuitive and fluent application of the SSK as a specific definition for "ease to use". Key feedback included a clear facilitation of QA approaches to Machine Learning based software. Teams reported being more comfortable in the selection of appropriate QA methods through the guidance by the evAIa SSK. This contributes to an increased level of team autonomy with respect to this selection.

Learning: SSKs are easy to use and open for individual refinement of non-defined methodologies and model aspects like the pre-condition handling of the evAIa approach with PQR. Applicable in insolated setups without the complexity to separate a semantic content into more SSKs.

Results: The SSK development kit 1.0 as facilitation tool for other SSKs was designed to build independent SSKs. Dependencies are not explicitly managed. They can be handled in many individual ways. The evAIa example shows this with the "pre-condition" approach.

Based on SSK development kit. Adaption: no

4.2. Case study B: Building blocks for a combined knowledge package

Use Case: Use the SSK approach as a generic template for building blocks of a network of more or less dependent knowledge "elements". An example for a semantic network of knowledge elements is a framework. The presented evaluation uses the efiS® framework [38] as a real-world case. The efiS® framework delivers itself and its building blocks with SSKs. The core objective of this use case is to show how different building blocks can be combined to a more holistic and dependent knowledge package. The presented case study focuses on the efiS® framework - the holistic knowledge package. The four pillars of the framework are comprised amongst others of aTWQ [39], TTM [32], PQR [36], LoD [38] and SSK. This example focusses only on these selected building blocks to show how the combination of different building blocks described by SSKs can be assembled to a network with dependencies to realize a holistic knowledge package. The aTWQ (agile Team Work Quality) approach supports evaluating the generic team maturity. It is based on an agile team maturity model in the sense of [40] and a team-autonomous maturity evaluation approach. The TTM (Technical Team Maturity) is a technology-focused extension helping to direct team maturity evaluation on their mastery of technologies (e.g. cloud technology). The PQR (Product Quality Risk) approach supports identifying product/service specific quality risks to mitigate them adequately during the development. The LoD (Level of Done) approach supports focusing on regulation and standard requirements relevant to the specific product/service domain. The LoD layer approach is an optional LoD extension fostering reuse of domain specific sets of regulations. Figure 4 presents an overview of the efiS® framework and a mapping of some of its building blocks to the four pillars (empowerment, focus, integrate and scale) of the autonomous value streams. The goal of the SSK dependency model is to offer a consistent set of SSKs as a knowledge package of the efiS® framework to autonomous organizations and teams.

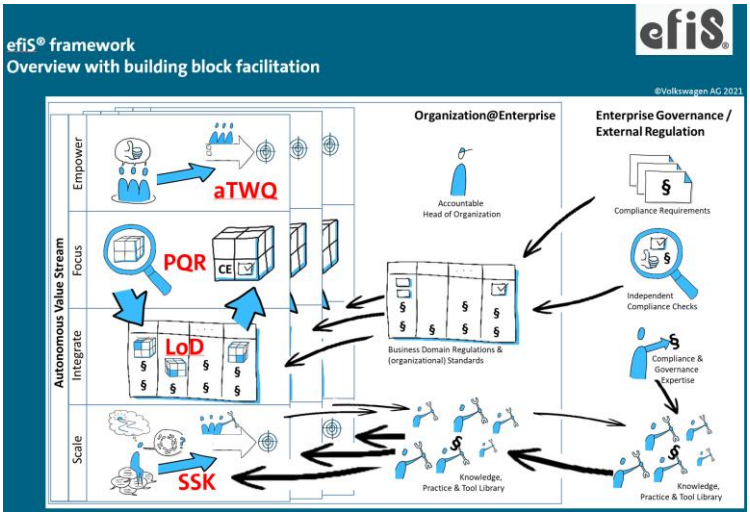


Fig. 4. The efiS® framework with its 4 pillars and selected building blocks (red) mapped to pillars.

Setup: The efiS® framework is assembled by building blocks. The building blocks were developed by expert teams around the building block topics – the SSK developer. Often at least one of the SSK developers becomes SSK maintainer over its life-cycle. From their view each building block is independent. However, the efiS® framework creates by the smart assembly a holistic framework. The work was to identify the level of dependencies between the building blocks and the framework. This was made by experts of the efiS® framework. The resulted SSK network usage was observed in a large SAFe® instantiation with 10 teams. The iteration phase terminates as no significant improvement potentials were identified.

Model the dependencies. In the case of the efiS® framework, the empowerment drives the autonomy of the team. This influences the capability to modify the delivery procedures. This can impact the LoD and the PQR handling, too. These are unidirectional dependencies.

The integration with a LoD defines the compliance aspects of the product development and specific domain knowledge of the organization for the delivery procedures. The LoD directly impacts the product development and influences the deliverables like products and services according to Conway's law [41]. Conway's law says that "systems image their design groups" also the relation between organization and product/delivery architecture/structure. This is a unidirectional dependency.

The focus on products with PQR identifies product/service specific aspects. Some of them have to be handled via procedures and actions handled by the LoD. This leads to a strong interaction of the PQR and LoD. This is a bi-directional dependency.

The scaling with SSK empowers the team with knowledge in most cases. In some cases, new knowledge is scaled by the team into the organization by adding or creating SSKs. This dependency is additive not changing structures. This is a bi-directional dependency.

Figure 5 shows the described dependencies.

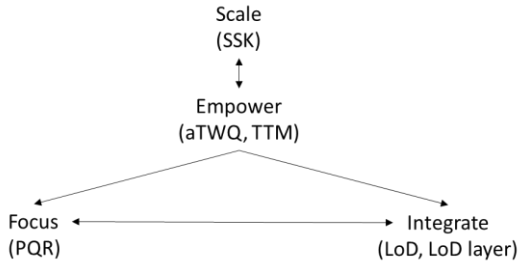


Fig. 5. Dependencies between the efiS® framework pillars.

Avoid content dependencies. Try to describe SSK content without content dependencies. Find the level of real impact/interaction of content – try to avoid the interaction to have the SSKs independent were possible. This also applies to SSKs within the same efiS® framework pillar. An example are the aTWQ and TTM approach. Both are designed to be orthogonal to avoid influencing each other. One handles the generic agile team capabilities the other specific technology capabilities. This makes it possible to have independent SSKs for aTWQ and TTM.

Via the scale pillar, knowledge comes into the team and empowers it. However, the content that is delivered can have dependencies on a semantic level. As described for the empower pillar, it is the work of the SSK designers to split the content to independent SSKs. This has been achieved for the SSKs of the efiS® framework. For other content, this has not been assured and therefore has to be handled in the team – especially in case of contrary "recommendations". The same is in case of contributing or building new SSKs by empowered teams. This insight should be mentioned as a recommendation in the SSK development kit that this insight should be handled during the SSK development. In some case this can lead to refactoring existing SSKs and new releases of aligned SSKs.

The empower pillar is driven by the knowledge delivered by SSKs to the teams e.g. the SSK for aTWQ or TTM. Furthermore, the team specific maturity profile, which typically grows over time, influences the team autonomy. This autonomy can have impact to the focus and integrate pillar, because the autonomy can lead to optimizations by the teams. The semantic level of impact is the adoption of the integrate pillar. By design of the LoD SSK this semantic dependency is handled by offering only an example which can be adopted. The same semantic level is with the focus pillar and the SSK for PQR approach, which is designed to be adoptable.

The focus pillar with its PQR SSK can have impact to the integrate pillar by adding product specific actions to the LoD. To handle this semantic dependency, the LoD SSK is by design open for the kind of extensions.

The integration pillar with its compliance handling impacts the product with the derived tasks and actions. The LoD SSK by design handles this interaction from and to the product focus pillar of the efiS® framework. The efiS® framework indicates this with the arrows between the pillars (Figure 4). This semantic dependency is modeled by the efiS® framework and is a design constraint for the SSK. This modeling is motivated as it is a fundamental part of the ISO 9001 requirements with "8.1 The organization shall plan, implement and control the processes, .., needed to meet requirements of relevant interested parties and the quality objectives for the provision of products and services …to realize opportunities and mitigate risks…" and "8.5.5 As applicable, the organization shall meet requirements for post-delivery activities associated with the products and services. In determining the extent of post-delivery activities that are required, the organization shall consider: a) the risks associated with the products and services;" This makes it easy for all teams to ensure that the efiS® framework implementation fits to the extracted examples as representatives for typical demands of the ISO 9001 and similar derived quality requirements of other standards and regulation which often are inspired by the ISO 9001 (or at least do not contradict to the ISO 9001). This makes it explicit that not the LoD implementation alone handles the specific product/service risks adequately.

Check for release packages. As it was avoidable to establish semantic and content dependency for all pillars and their SSKs, no release packages have to be defined. However, the LoD and PQR SSKs have to be controlled every change to keep this level of "weak" dependency without release packages.

Learnings: The efiS® framework SSK team realized that the LoD and the optional layer approach can be described in one SSK to avoid redundancies between highly dependent SSKs for the case that they are delivered by a LoD and LoD-layer SSK. This leads to a refactoring to one SSK which contains the LoD and LoD layer approach.

Furthermore, this learning was integrated in the generic SSK development kit to remember SSK developer to avoid dependencies wherever possible. All other investigated dependencies were handled intuitively adequate in the efiS® framework SSK setup.

Results: The instantiation of the efiS® framework based on the SSKs is possible. The authors have been accompanying programs which are continuously adopting the efiS® framework. As an example, one program adopted the efiS® framework with its core building blocks with approximately 3 Program Increments (PI) of their SAFe® instantiation without explicit agile coaching by ACE efiS® facilitators or efiS® trainings. In each release train, selected building blocks were established. Equipped only with the SSK and some broad agile knowledge, they started with the efiS® framework SSKs to understand the underlying concepts. They identified that the initial setup of the LoD will take its time due to the stakeholder feedback iterations in principal, however also other factors linked with the adoption of new practices. Then they progressed with the product risk evaluation to get "scope". In a next step, they initiated with scoped and "open mind" teams with the aTWQ and TTM. A weekly 15 minutes call was established to talk about "issues". Sometimes this call was omitted, or an extra review/feedback session was added for outcomes like the LoD. Overall, this accompaniment was less than 1h per week and was a "nice to have" because the basic adoption of the efiS® framework was made based on the SSK. The advanced specific feedback and coaching advices/recommendation are outside the SSK scope. The systematic dependency refinement approach helps to go a step towards the long-term goal: self-explanatory of efiS® based on a basic set of SSKs. The presented simplified use case example is applicable also to networks of more SSKs. The entire efiS® framework is currently described by 7 SSKs. The relation of the SSKs builds a logical flywheel in which the whole framework is more than the sum of the building block parts.

Based on SSK development kit. Adaption: semantic dependency needs synchronized versions.

4.3. Case study C: Training Take-away

Use case. The integration of SSKs into the practical parts or exercise lessons of trainings ensures that the learnings are "packaged" into a useable setup for real world application. This is motivated by the revised Bloom's taxonomy [42] and the competency driven curriculums of trainings. This has been applied in the efiS® trainings and TaaS trainings for the Volkswagen Group Academy and the IT Academy, respectively. All the mentioned SSKs existed before the trainings were designed. This constraint led to the point that the trainings had to "reuse" an as-it-is SSK. The implication is that the case for the practical lessons needed to be designed to fit to the training curriculum and be a good reprehensive case for the SSK. The effort of the case design in practice requires less effort than to develop the practical lessons from scratch. This positive side-effect fosters the reuse of SSK for trainings, too. An issue is that the life-cycle of the SSK is independent from the training life-cycle. Always using the newest version of the SSK, may require some adjustment of the trainings exercise case. However, in practice, a smart design of the exercise case oriented on the recommendations of the dependency section avoids this in most cases. The evaluation in two TaaS and two efiS® framework

trainings were positive examples for easy integration, low creation effort (compared to building them from scratch), as well as a high practical relevance. The added value for the trainees for the understanding and adoption of the learned concepts into their daily practical work environment was reported as high. A useful side effect of the "lab character" of the trainings is that one can observe the trainees in their practical SSK applications during the trainings. This is helpful to get inspirations and feedback for improving the SSK for better usability.

Setup: Typically, the trainings are developed and maintained by different persons than the SSKs. At least the point in time in which they are developed are different, because trainings typically follow the demand. However, training demand comes after the "content" is existing and not reverse. Parts of the training content can be described in SSKs. In this case the trainers used feedbacks of the trainees about the usability of the SSKs in the exercises. The iteration phase terminates as no significant improvement potentials were identified.

Learnings: It is possible to integrate existing SSKs into trainings. For such trainings, case studies have to be designed which use the SSKs in the learning context of the training. With this setup, it is possible to train people how to work with SSK and how to get most value out of them. Furthermore, it can be used as case study to observe the application of SSKs to get ideas to improve them.

Results: The integration of SSKs into trainings is a step forward to blended-learning. The SSK can be used during the training or during preparation or "home-work". After the training, the SSK can directly be used in the trainees' organizations in their specific project contexts. This helps bundling training content for real world application without additional transfer work by the trainees. This is a big step forward to ensure that training content is relevant and usable for trainees.

Based on SSK development kit. Adaption: no

4.4. Case study D: Body of Knowledge

Use case. Expert teams use SSKs to structure and offer practices with their groups. To indicate the maturity of practices, an additional maturity-attribute is used. It is a collection of expert experience which grows in maturity with application by others (validation and enhancement over time) – from a successful implementation to a best practice. The attribute is realized as a set of tags. The tag values of this maturity attribute have the following significance:

- Explore: practices under development – for experiment in a limited risk environment
- Verify: practices with a disruptive potential – for early adopters
- Open mind: emerging practices on the way to be main-stream
- Must have: used for practices which are recommended as state of the art or main-stream
- Deprecated: practice is not recommended to use anymore

The IT Quality Manager community uses this maturity attribute to build a body of knowledge for their communities. This approach enables all to contribute and use the practices. Depending on their maturity and application context, the "portfolio" of practices is more or less huge. The moving to a new maturity stage of practices is made

in the cyclic community meetings. This makes it transparent to everybody what is new or what has been changed. The intention behind this approach is that the experts can build their own set of practices for their daily businesses and keep it up to date within the learnings of the entire community. Every community member can participate and contribute to practices with new insights and learnings form their product and service domains of the entire Volkswagen Group IT.

Setup: The experts in the IT Quality Manager community are prosumers of the SSKs. So, they can decide what is missing and what should be enhanced to fit their quality expectation for their Body of Knowledge. The iteration phase terminates as no significant improvement potentials were identified.

Learnings: The beginning was not easy, because it was new for the community to be themselves the authors of the practices they want to spread and establish within the organization. This needs time and examples to show what should be in the practice collection and in which maturity state. However, the pressure to add practices was "build" with the strategy that there was no central team defining and maintaining the practice collection like often done by central expert or governance teams. This turned out to be an important lever for team autonomy and self-governance, while fostering mutual collaboration and knowledge sharing among the teams at the same time

Results: After initial practice collection during setup a slowly growing set of practices for the IT Quality Manager community. However, slow growth it not a negative indicator, because a practice set should not be an encyclopedia. It should rather be selection of practices that are compatible with the working culture of the specific organization. The important point is that this case study shows that a community can work with SSK's to build their domain-specific body of knowledge.

Based on SSK development kit. Adaption: semantic life-cycle requires additional attributes

4.5. Generic Topic: Source and Updates of SSK

Table 1. Meta data for SSKs to model advanced use cases

Attribute	Content
Dependency	Dependency with other SSK and optional info weak/strong
Maturity	Tag(s) for state
Source	URL(s)
Version	x.y.z
Maintainer	Email(s)

The SSK approach creates the significant challenge of keeping up to date the different SSK versions create by the users in different project teams and organizational entities. A recommendation is to establish one official source to avoid many update-sources. However, this only supports the SSK maintainer team. As a meta data of SSKs, the source tag is a useful information for users, to know where they can check for updates. The version helps to see the type of update if the notation x.y.z is used. X is the major version indicating non-backward compatible changes. Additionally, it is useful to have a history/change-log in the SSK to show what was really changed. Y is the minor version

used for extensions or enhancements and z for bug fixes. The producer or maintainer is given as contact person.

Table 1 presents the proposed meta data attributes which were useful during the holistic evaluation of advanced use cases for SSKs.

5. Limitations

In order to point out the limitations of the proposed approach, we distinguish between the SSK approach itself as defined by its methodology, and the evaluation setup with its constraints.

5.1. SSK Approach

The SSK approach is not a formal modeling method. This is a limitation by design, but needed to enable most people in agile organizations to produce and maintain SSKs without deep formal description knowledge. The minimal formal modeling extensions are a benefit in more complex use cases, but they are not automatically identified like a SSK dependency and model checked which can lead to errors and failures during the SSK life-cycle of creating, updating and deleting, if it is no longer useful.

5.2. Evaluation Setup

The case study evaluations cannot cover all possible real-world scenarios. The selected scenarios are from a real-world enterprise context, but they are only indicators that "typical" setups get benefits from the presented approach. Another limitation in the setup is that by design the distribution to autonomous teams makes measuring difficult. Metrics can only partly cover samples of observed SSK applications like trainings. Furthermore, samples are not randomized, since training participants represent a specific group of people who are interested in the training topic. These people have the skills and knowledge "required" for the training, and may not represent the entire organization. Additionally, the researchers act as designer and observers of the application of the designed artefacts. This can lead to some selective observations. Selective observation can lose sights of some potential interesting improvement aspects or issues by the application. Due to this, there might also be some bias in the evaluation of the presented approach's effectivity and efficiency. The corporate context made that the authors could not avoid this setting. However, they did their best to let the teams ("clients" of this approach) judge, even if not necessarily based upon strictly quantifiable criteria.

6. Conclusion and Outlook

The paper focuses on the definition of requirements for extending SSKs to becoming a means of fostering the agile transition in large company settings, especially in distributed on-line (virtual) organization settings like during the Covid pandemic. These enhanced SSKs open a new dimension of application and adoption scenarios in practice and provides new insights from an academic perspective. More precisely, this work shows the practical feasibility of fostering large-scale agile transitions with a self-service approach that relies on team autonomy both in terms of its design and its deployment and evolution by the experimental application of SSKs in the wide range of evaluation setup. SSKs providing the fundamental basis of this approach, we have shown that by extending the life cycle model of such SSKs and integrating the consistent handling of SSK dependencies. The SSKs can be effective drivers for the implementation of a holistic agile practices framework. We have shown this in the form of practical use cases in the complex setting of the Volkswagen Group IT. These use cases covered stand-alone application of subject knowledge SSKs, related and complementary SSKs constituting a knowledge framework, as well as a bundle of SSKs instantiated in different organizational units and representing a body of knowledge. We have also shown that classical trainings can complement the self-service approach most effectively when they are based on the SSKs themselves.

Practically, we have shown advanced design options of SSKs for different use cases that lead to more self-explanatory SSKs that reduce training/coaching efforts on large-scale agile transition journeys. New business cases for SSKs have been made possible through dependency modeling and SSK life-cycle management. So far, we have not yet shown the specific, quantifiable influence of SSK application on VW Group IT's way from a specific point A to a specific point B in their agile transition journey. Our objective was rather to show that SSK's can and already did give a significant contribution to increasing the autonomy and knowledge-sharing of distributed, diverse teams in the large organization.

Possible next steps shall move further in the direction of self-explanatory SSKs. However, it is not trivial to ensure that without adding more attributes while keeping complexity of the approach low, which is important to ensure that mostly everybody can still build and use SSKs. Furthermore, the measurement based on generic KPIs and the management of distributed SSKs is still an open topic which needs more investigation in the future. Here, the focus could be to establish representative sampling or "sending home" some meta data, but this requires some kind of (temporally) online-connection. This would also enable something like an (auto-)update-function to ensure that only the latest versions of an SSK are used.

References

1. LEADERSHIP SKILLS & STRATEGIES V U C A world [online]. Available: https://www.vuca-world.org/ (current March 2022)
2. Serrat O. (2017) Bridging Organizational Silos. In: Knowledge Solutions. Springer, Singapore. https://doi.org/10.1007/978-981-10-0983-9_77. (current March 2022)

3. Kowalczyk, M., Marcinkowski, B., Przybylek, A.: Scaled agile framework: Dealing with software process-related challenges of a financial group with the action research approach. In: Journal of Software: Evolution and Process, (2022) →in review
4. R. M. Parizi, T. J. Gandomani and M. Z. Nafchi, Hidden facilitators of agile transition: Agile coaches and agile champions, *8th. Malaysian Software Engineering Conference (MySEC)*, 246-250, doi: 10.1109/MySec.2014.6986022. (2014)
5. Poth A., Kottke M., Riel A.: Scaling Agile – A Large Enterprise View on Delivering and Ensuring Sustainable Transitions. In: Przybyłek A., Morales-Trujillo M. (eds) Advances in Agile and User-Centred Software Engineering. LASD 2019, MIDI 2019. Lecture Notes in Business Information Processing, vol 376. Springer, Cham. https://doi.org/10.1007/978-3-030-37534-8_1 (2019)
6. Moe, N. B., Olsson, H. H., & Dingsøyr, T.: Trends in large-scale agile development: a summary of the 4th workshop at XP2016. In Proceedings of the Scientific Workshop Proceedings of XP2016 (1-4). (2016)
7. Dikert, K., Paasivaara, M., & Lassenius, C.: Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software, 119, 87-108. (2016)
8. Przybylek, A., Albecka, M., Springer, O., Kowalski, W.: Game-based Sprint retrospectives: multiple action research. In: Empirical Software Engineering 27, 1, (2021), [Online]. Available: https://doi.org/10.1007/s10664-021-10043-z (current March 2022)
9. Poth A., Nunweiler E.: Develop Sustainable Software with a Lean ISO 14001 Setup Facilitated by the efiS® Framework. In: Przybyłek A., Jarzębowicz A., Luković I., Ng Y.Y. (eds) Lean and Agile Software Development. LASD 2022. Lecture Notes in Business Information Processing, vol 438. Springer, Cham. https://doi.org/10.1007/978-3-030-94238-0_6 (2022)
10. Patanakul P, Jiyao C, Lynn G.S.: Autonomous teams and new product development. Journal of Product Innovation Management 29(5). 734-750. (2012)
11. Lee LL.: Knowledge sharing metrics for large organizations. Knowledge Management: Classic and Contemporary Works, The MIT Press, 403-419. (2000)
12. Hung W, Jonassen DH, Liu R.: Problem-based learning. Handbook of research on educational communications and technology 3(1). 485-506. (2008)
13. Hmelo-Silver CE, Barrows HS.: Goals and strategies of a problem-based learning facilitator. Interdisciplinary Journal of Problem-Based Learning, 1(1): 21-39. (2006)
14. Chau T, Maurer F, Melnik G. Knowledge sharing: Agile methods vs. tayloristic methods. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, 302-307. (2003)
15. Ellison NB, Gibbs JL, Weber MS.: The use of enterprise social network sites for knowledge sharing in distributed organizations: The role of organizational affordances. American Behavioural Scientist, 59(1). 103-123. (2015)
16. Paasivaara, M. and Lassenius, C.: Communities of practice in a large distributed agile software development organization - Case Ericsson, Information and Software Technology, vol. 56, pp. 1556-1577 (2016)
17. Cockburn A, Highsmith J.: Agile software development, the people factor. Computer, 34(11). 131-133. (2001)
18. Hoda R, Murugesan LK.: Multi-level agile project management challenges: A self-organizing team perspective. Journal of Systems and Software (117). 245-257. (2016)
19. Dorairaj S, Noble J, Malik P.: Understanding team dynamics in distributed Agile software development. International conference on agile software development, Proceedings, Springer, Berlin, Heidelberg, 47-61. (2012)
20. Poth, A., Kottke, M. and Riel, A.,: The implementation of a digital service approach to fostering team autonomy, distant collaboration, and knowledge scaling in large enterprises. Human Systems Management. 1-16. (2020)

21. Schank RC, Berman TR, Macpherson KA.: Learning by doing. Instructional-design theories and models: A new paradigm of instructional theory, 2(2). 161-181. (1999)
22. Hoic-Bozic N, Holenko Dlab M, Mornar V. Recommended system and web 2.0 tools to enhance a blended learning model. IEEE Transactions on Education 59(1). 39-44. (2015)
23. Hoic-Bozic N, Mornar V, Boticki I.: A blended learning approach to course design and implementation. IEEE Transactions on Education 52(1). 19-30. (2009)
24. Latchman H, Salzmann C, Gillet D, Bouzekri H. Information technology enhanced learning in distance and conventional education. IEEE Transactions on Education 42(4). 247-254. (1999)
25. Singh H. Building effective blended learning programs. Educational Technology 43(6). 51-54. (2003)
26. Williams SW.: Instructional Design Factors and the Effectiveness of Web-Based Training/Instruction. In: The Cyril O. Houle Scholars in Adult and Continuing Education Program Global Research Perspectives: Volume II. Compiled by Cervero RM, Courtenay BC, Monaghan CH, 132-145. (2002)
27. Dukhanov A, Karpova M, Bochenina K.: Design virtual learning labs for courses in computational science with use of cloud computing technologies. Procedia Computer Science 29. 2472-2482. (2014)
28. Raspotnig C, Opdahl A.: Comparing risk identification techniques for safety and security requirements. Journal of Systems and Software, 86(4). 1124-1151. (2013)
29. IEC 61508.: Functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission, 2nd ed. (2008)
30. Ashby W. Ross: Introduction to Cybernetics. Chapman & Hall, London. (1956)
31. Poth A., Jacobsen J., Riel A. : A Systematic Approach to Agile Development in Highly Regulated Environments. In: Paasivaara M., Kruchten P. (eds) Agile Processes in Software Engineering and Extreme Programming – Workshops. XP 2020. Lecture Notes in Business Information Processing, vol 396. Springer, Cham. https://doi.org/10.1007/978-3-030-58858-8_12. (2020)
32. Poth, A., Kottke, M. and Riel, A.: Measuring team work quality in large-scale agile organizations. IET Software, John Wiley & Sons, Inc. (2021)
33. Poth, A., Kottke, M. and Riel, A.,: The implementation of a digital service approach to fostering team autonomy, distant collaboration, and knowledge scaling in large enterprises. Human Systems Management. 1-16. (2020)
34. Poth, A., Kottke, M. and Riel, A.,: September. Scaling agile on large enterprise level with self-service kits to support autonomous teams. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS) (pp. 731-737). IEEE. (2020)
35. Järvinen, P.: Action research is similar to design science. Quality & Quantity, 41(1), pp.37-54 (2007)
36. Chaohui Z.: A knowledge base with dependencies | IEEE Conference Publication. IEEE Xplore (2014). [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6980868 (Current March 2022
37. Poth, A. and Riel, A.,: August. Quality requirements elicitation by ideation of product quality risks with design thinking. In 2020 IEEE 28th International Requirements Engineering Conference (RE). 238-249. IEEE. (2020)
38. Poth, A., Kottke, M., Heimann, C. and Riel, A.: The EFIS framework for leveraging agile organizations within large enterprise, XP'21 (2021)
39. Poth, A., Kottke, M. and Riel, A.: Agile Team Work Quality in the Context of Agile Transformations–A Case Study in Large-Scaling Environments, European Conference on Software Process Improvement. 232-243. (2020)
40. J. Becker, R. Knackstedt, J. Poeppelbuss. Developing maturity models for IT management. Bus. Inf. Syst. Eng., 1. 213-222. (2009)

41. Conway M. E.: How Do Committees invent?. (1968). [Online]. Available: https://hashingit.com/elements/research-resources/1968-04-committees.pdf (Current March 2022)
42. Bloom BS, Krathwohl DR, Masia BB.: Bloom taxonomy of educational objectives. Allyn and Bacon, Pearson Education, (1984)

Alexander Poth studied computer engineering at Technical University Berlin. He is IT Quality Manager in the Group IT at Volkswagen AG. He is product owner of Testing as a Service (TaaS) and cares about many other interesting things like the Quality innovation NETwork (QiNET) of Volkswagen to ensure that IT service/product development and delivery can fit the quality expectations.

Mario Kottke studied Business economics and computer science at Martin-Luther-University Halle. He is an Agile Coach at Volkswagen AG. His working field is the first contact for projects who want to work agile and consulting during the start. His experience is 4 years working as a scrum master and 3 years as an agile coach.

Andreas Riel received an MSc (Dipl.-Ing.) in Telematics from Graz University of Technology, Austria, a PhD in Meachtronics from Vienna University of Technology, Austria, as well as a habilitation (HDR) in Industrial Engineering from Grenoble Alps University, France. He is an independent coach, consultant and trainer in Systems Engineering, Quality Engineering, Innovation Management, as well as Digitalization and Industry 4.0 to major European industry, especially automotive. He is also a part-time Professor at the Institute of Engineering of Grenoble Alps University.

Received: November 12, 2021; Accepted: July 29, 2022.

Computer Science and Information Systems 20(1):251–276 https://doi.org/10.2298/CSIS211001008H

Multi-constrained Network Occupancy Optimization ⋆

Amar Halilovic[1], Nedim Zaimovic[2], Tiberiu Seceleanu[3], and Hamid Feyzmahdavian[4]

[1] Ulm University, Ulm, Germany
{amar.halilovic}@uni-ulm.de
[2] ALTEN AB, Västerås, Sweden
nedim.zaimovic@alten.se
[3] Mälardalen University, Västerås, Sweden
tiberiu.seceleanu@mdh.se
[4] ABB AB, Västerås, Sweden
hamid.feyzmahdavian@se.abb.com

Abstract. The greater the number of devices on a network, the higher load in the network, the more chance of a collision occurring, and the longer it takes to transmit a message. The size of load can be identified by measuring the network occupancy, hence it is desirable to minimize the latter. In this paper, we present an approach for network occupancy minimization by optimizing the packing process while satisfying multiple constraints. We formulate the minimization problem as a bin packing problem and we implement a modification of the Best-Fit Decreasing algorithm to find the optimal solution. The approach considers grouping signals that are sent to different destinations in the same package. The analysis is done on a medium-sized plant model, and different topologies are tested. The results show that the proposed solution lowers the network occupancy compared to a reference case.

Keywords: industrial control networking, packing algorithms, network performance.

1. Introduction

Increasingly different types of information are required in industrial processes and engineering systems, resulting in the diverse and widespread use of networks. Today, there are about 25 billion Internet of Things (IoT) devices deployed in the world, and that number has increased for about 20 billion in the last five years [31]. Thus, we can expect communications-capable machines to be the most common type of device in the future, leading to a considerable growth of the data volume. A significant increase of the network traffic and network load may limit and even decrease the network speed while increasing the data loss. Therefore, powerful data communication is necessary to handle this growth.

Network occupancy can show how large is the network load. Although network speeds are getting higher, it is always desirable to minimize the network occupancy, as data volumes and the number of related critical features are also increasing. Thus, network optimization plays a vital role in many applications that have certain speed and reliability requirements. Such requirements refer to accurate delivery of packages to their destination, in some guaranteed time frame.

⋆ Extension of the article published in the 7th Conference on the Engineering of Computer Based Systems (ECBS 2021), May 26–27, 2021, Novi Sad, Serbia.

Problem formulation. Our work is triggered by a move from the classical *Control-Centric System Model* towards a *Network-Centric Control Model* in the context of industrial automation systems (IAS). An IAS is concerned with the acquisition, delivery, and processing of input signals from sensors to controllers and delivery of the controller signals to actuators. Such a distributed system is usually deployed on the Open Platform Communications Unified Architecture (OPC UA) [1] protocol, running on top of Ethernet network technology. We discuss here about the part of the model between sensors and controllers. The reverse communication, plus additional - management related communication - can be covered by an extension of the proposed solution.

The sensors provide the process data to the control system. Their signals are aggregated in *Field Control Interface* devices (FCI) and grouped in larger, as possible, disjunctive datasets. FCIs in the network have registers for storing signal values. Each dataset has an *ideal* frequency at which the FCI assembles them into packets. However, for most of the signals, values packed at different than ideal time moments may still provide for good process management. On the other side of the network, there are controllers responsible for specific tasks. For each task, a dataset of signals is assigned to the process. These datasets overlap with datasets of signals sourcing at potentially different FCIs.

Further, *task allocation* is a problem that can be identified in multiple domains, and acknowledgment of the related complexity has long been on researchers' agendas. From a technical perspective, task allocation is considered an "essential problem" in the area of parallel computing [26], resurfacing with the advent of multi-core processors [15], and today also being "one of the most fundamental classes of problems in robotics" [4]. Moreover, scheduling task execution in distributed systems is, for a long time now known as an NP-hard problem [12]. Task allocation is also raising aspects of performance, the "optimality" of the action being one crucial step in the design of modern systems [28].

We identify our problem as both an allocation issue and a scheduling and performance issue. We study the above both considering an *ideal* network - where no packages are lost, as well as a "lossy" network, where a certain number of packages don't reach their intended destination. However, we do not propose here a solution for task allocation - there are several options to follow in the research literature, but, at the moment, we just want to illustrate the effect of this at the levels we discuss.

Our goal is to find a (close to) optimal packing schedule, maximize the packet utilization, thus minimizing the number of packets sent, so that the network occupancy, expressed in Bytes (B), is minimized.

Contribution. We consider to address the following topics:

- Determine a suitable optimization algorithm that minimizes network occupancy given multiple constraints.
- Investigate how different data packing methods affect the network occupancy in a network-centric model.
- Study the impact of network topology on the network occupancy in a network-centric model.
- Study the impact of different task-controller allocation on the network occupancy in a network-centric model.
- Study the impact of package loss on network occupancy and investigate how a suitable optimization algorithm handles package loss.

The considered constraints can be identified as: the size of the package, the assignments of datasets to tasks, the various timing specifications (reading times, packing times, expected arrival times), data updates.

Solution space. The first direction of study brings us to the rich body of research related to *bin packing* [16], [21], [8], [9]. There are many variants and forms of "bin packing" problem and solutions to it. However, each of them tries to fit the finite number of items into the finite number of bins of a fixed volume so that the number of bins is minimal. This problem fits in the class of the 1D packing, where only one dimension varies, and that is the size of the signal. In this problem, packets represent bins of a fixed maximum size, while signals from sensors represent the items whose size varies.

Heuristic and meta-heuristic algorithms and approaches [18] are generally used for finding sub-optimal solutions or solutions that are good enough under given conditions when classical approaches are too slow or fail to find optimal solutions. Alternative approaches are also studied for speeding-up bin packing solutions [10], [13].

Another direction is to investigate the use of various artificial intelligence approaches (*e.g.* genetic algorithms [24]), or more sophisticated approaches. These approaches can be applied to large-scale optimization problems, specifically if problem-specific algorithms do not yield satisfactory results. The main drawback of these algorithms is that they are typically more complex and require adjustment to fit the needs of the specific problem.

For the size of the problem here, we pursue the bin packing approach initially, remaining to look into further solution dimensions if the results are not sufficiently useful.

2. Background

Industrial plants are usually very complex systems, including a large number of devices such as sensors, controllers, and actuators. An important task is to connect all devices into a single infrastructure to make industrial communication work from the field level up to the management level [2]. The switching technology is one among many used in modern industrial networks, built with the support of devices such as hubs and switches. A hub forwards the packet to all its ports. When a packet enters a switch, the switch reads the Medium Access Control (MAC) address stored in the packet header, and, based on the value, it takes decisions on which of its ports to send the packet through.

Controller-Centric Systems. Devices in a *controller-centric architecture* - CCA (Fig. 1) - are directly connected to controllers, thus making controllers to "own" devices. Configuration data from the engineering to devices is deployed over controllers. Controllers not only focus on control logic execution, but also require certain knowledge about device-specific implementations. Routing of device information goes through the controllers.

Network-Centric Systems. The logical topology of the *network-centric architecture* - NCA (Fig. 1) - brings several benefits compared to CCA. As devices and controllers are logically on the same bus, any controller can use signals from any device. Thus, the system is more flexible. However, this also brings some drawbacks, one of which is the possible network congestion, especially in large systems.

The bin packing problem. Items of different volumes must be packed into a finite number of bins or containers each of a fixed given volume in a way that minimizes the number of bins used (Fig. 2).

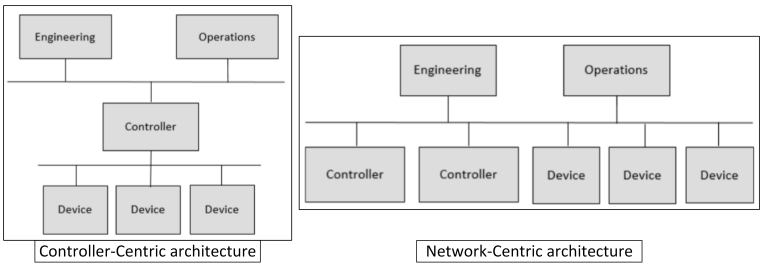


Fig. 1. Controller-Centric vs. Network-Centric Architectures

The total size of the items assigned to one bin should not exceed its capacity. This problem is known as **bin packing**, and it is a very popular challenge amongst the researchers across various domains. It comes in multiple variants, such as three-dimensional packing [25], packing by weight [19], and packing by cost [22].

The bin packing problem has many applications, such as filling up containers [27], loading trucks with weight capacity constraints [17], and job scheduling [6].

Heuristic algorithms. Bin packing is a NP-hard problem, and it is proven that no algorithm can achieve a performance ratio better than $\frac{3}{2}$ (unless $P = NP$) [5]. Heuristic methods are most commonly used to find the optimal solution to this problem.

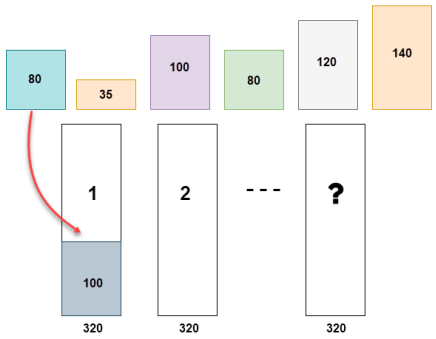


Fig. 2. The Bin packing problem.

One of the simplest heuristic algorithms for bin packing is probably *Next-Fit* (NF) [5]. NF places an item in the currently ”opened” bin, if the item fits inside. Otherwise, the current bin is closed, and the item is placed inside a new bin. A NF modification, *Next-k-Fit* (NkF) [20], holds k containers open instead of only one. *First-Fit* (FF) [8] is one of the most commonly used heuristic algorithms for solving the bin packing problem. Here, each item is placed into the first bin in which it will fit. If there is no such bin, it puts the item inside a new bin. Another well-known algorithm that provides a fast but often non-optimal solution is *Best-Fit* (BF) [9]. The idea is to put the item in the fullest bin in which it fits. *Worst-Fit* (WF) [5] is very similar to BF. Instead of putting an item in the bin

where it leaves the smallest space, it is placed in the bin with minimum load. All these algorithms belong to a group of so-called *online* algorithms, and they consider items in an order defined by a list $\mathbf{L}$.

To measure the performance of online algorithms, two terms are defined:

- $A(\mathbf{L})$. The number of bins used when algorithm A is applied to list $\mathbf{L}$.
- $OPT(\mathbf{L})$. The optimum number of bins for list $\mathbf{L}$.

Table 1 shows the upper bounds for each of the mentioned algorithms. Since the upper bounds on FF and BF algorithms are lower than those on NF and WF, the two most commonly used algorithms are Best-Fit Decreasing (BFD) and First-Fit Decreasing (FFD). Both algorithms initially sort the items in decreasing order of size and assign the larger items first. In this way, a lower upper bound is achieved [7]:

$$FFD(\mathbf{L}) \leq \frac{11}{9}OPT(\mathbf{L}) + \frac{6}{9}.$$

Note that the complexity of the Best-Fit Decreasing (BFD) algorithm is $\mathcal{O}(n \log n)$ for n objects, which is greater than the $\mathcal{O}(n)$ running time of the NF algorithm.

Table 1. Upper bounds of several algorithms.

Algorithm	**Upper bound**
Next-Fit	$NF(\mathbf{L}) \leq 2OPT(\mathbf{L}) - 1$ [5]
First-Fit	$FF(\mathbf{L}) \leq \lfloor 1.7OPT(\mathbf{L}) \rfloor$ [8]
Best-Fit	$BF(\mathbf{L}) \leq \lfloor 1.7OPT(\mathbf{L}) \rfloor$ [9]
Worst-Fit	$WF(\mathbf{L}) \leq 2OPT(\mathbf{L}) - 1$ [5]

3. Related Work

We divide this section on two of the main topics that we address in our study, *traffic optimization* and *package loss*. As observable in the next paragraphs, these are strongly correlated, but also independent solutions can be identified, in specific contexts.

Traffic optimization. Leinberger et al. [16] proposes a new multi-capacity aware bin packing algorithm for job management systems (JMS). Multi-capacity refers to different resource requirements, such as the number of CPUs and amount of memory. The "bin" represents the parallel system, while the job wait queue is represented by an item list. The specific heuristics are though too heavy for the topic we address in this work.

In [27], the container loading problem with expiring orders is addressed. The authors classified this problem as a three-dimensional optimization problem with constraints such as orientation, stability, and loading priority. The items of an order must be entirely placed in the container or entirely be left behind. A heuristic algorithm handles first the expiring and then the non-expiring orders. The timing constraints make this approach not suitable in our context.

In [21], the bin packing problem is formulated as a multi-objective optimization problem: minimizing the bins used and minimizing the heterogeneousness of the elements in each bin. These two conflicting goals are formulated as a vector optimization problem. The authors emphasized the importance of trade-offs in this kind of optimization problem, which is the same we do here (by allowing some signals to not be sent over the network).

The bin packing problem under linear constraints is presented in [29], where the size of items to be packed is not given in advance. A modified Next-Fit algorithm is proposed. Linear programming computes the optimal value for size of items, and then the Next-Fit algorithm is deployed to solve the bin packing problem. The algorithm runs in polynomial time and has the same approximation ratio as CNF. The specifics of our application area - especially timing and package targeting - make the CNF not suitably.

A meta-heuristic approach for real-time task scheduling problems is employed in [24], guaranteeing end-to-end tasks' deadlines in distributed environments. Two different exploration scenarios are analyzed, single (looking for the minimal number of processing units for all the tasks) and multi-objective exploration (considering the total number of processing units and the end-to-end finishing time for all the jobs). Our problem is placed at a different level of granularity (tasks vs. packages), hence not benefiting enough from this approach.

Task allocation. There is a rich body of research focusing on the optimality of task allocation in networked environments. For on-chip networks, for instance, greedy network layer-based algorithms are found to be one solution [14]. Multiple criteria (distance to other nodes, energy levels, energy consumption, position) are analyzed when operating robotic networks [3]. A graph based framework is defined here for dynamically assigning tasks to set(s) of robots. For mesh networks, an interesting research [23] stresses the importance of architectures and algorithms, when considering the introduction of two new allocation algorithms.

We presented the above as just a (very) small part of what researchers focus on when discussing task allocation. We will not try to identify here a (new) allocation policy, but are illustrating the impact of allocation decisions on network occupancy.

Package loss. Several reasons are identified as leading to packages losses in modern day networked systems. As described in popular literature (e.g. [11]), some of the most important ones are *network congestion*, *defective hardware*, *outdated hardware and / or software*. Immediate solutions for the last two causes come from replacements and / or updates, respectively.

Packet loss is addressed at physical network levels [32], when some information is available on the forecasted data, or additional controls are necessary when such data is not available. Differently with the cited work, the transfers we intend to schedule are time-triggered and not event-triggered. Hence, we believe we do not need such a complex approach.

A new end-to-end congestion control algorithm is proposed in [30], based on a Naive Bayesian model. The approach analyzes both wired and wireless systems. The approach does apply priority features to the packets, and provides an identification of "where" (wired/wireless) the losses appear. The priority schemes are used to improve traffic conditions when the congestion is high. In our approach, we try to build the system such that congestion is controlled from the beginning, and thus packet losses are only very rare.

4. Building the approach

The *Network-Centric Control Model* (NCCM) allows us to model the acquisition and delivery of the input from sensors to controllers, their processing and the delivery of the controller signals to actuators on the *Smart Control Platform*. Fig. 3 depicts the NCCM, with the virtual paths that some packages travel.

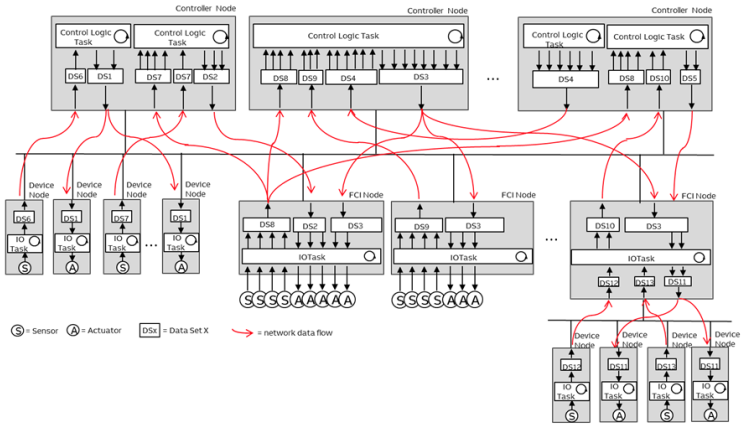


Fig. 3. Network-Centric Control Model

We focus here on the part of the model between sensors and controllers. Sensors are directly connected to the network bus, or via the FCIs - the approach we consider. Signals from assigned sensors are aggregated by the FCI in larger datasets to reduce the network load. Datasets in one FCI are disjunctive to datasets in any other FCI in the network. The time period in which the signal sends the data is negligible, and it is assumed that the FCI's registers contain the latest values of every assigned signal.

The network is modeled as an ideal system without delays. The network occupancy is defined as a number of bytes that flows through the network over a specified time interval. There is a number of switches that forward packets to the targeted controllers, and each switch is directly connected to at least one controller, and an FCI or another switch. Depending on the network topology, the number of network elements, their connections, and their disposition varies. At the other side of the network are controllers running tasks. Each task is defined by its period and execution time and is assigned a dataset of signals to process. These datasets overlap with datasets of signals in the FCIs.

The network package. The term *message* can have multiple meanings in networking. In this paper, we consider payload as a message. So, the data read from the sensor is a payload that is sent on the network together with the rest of the package. The network package also includes protocol, security, and encoding-specific data. Within the FCI, the entire package is created and sent to the appropriate controller. The *DataSetMessage* is actually data from a single sensor, while the *NetworkMessage* is the overall payload that

contains all signals. The Unified Access Data Plane (UADP) dataset payload and other parts of the network package are shown in Fig. 4-A.

The thus package consists, in addition to the payload, of different types of headers, security footer and signature. Everything except payload can be considered a fixed size. A simple representation of the resulting *abstract* package is shown in Fig. 4-B. The messages contained in the payload are sensor signals. The data size of the signal is marked as DS, and it depends on the type of signal.

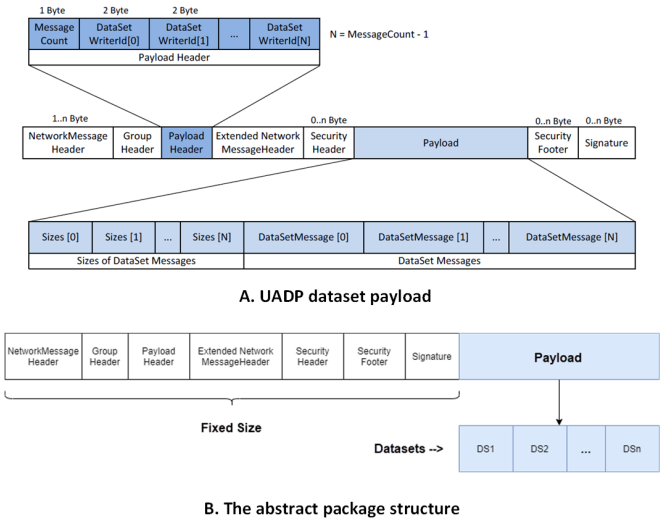


Fig. 4. A: The UADP dataset payload [1]; B: The abstract package structure.

Working example. In Fig. 5, we introduce a simple network topology, which we will use to present the algorithm. Each controller-controller, switch-switch, and the switch-controller connection is marked with a number. These numbers represent network lines / segments. Any information sent on the network travels some of these segments to its destination (one of the controllers). For example, let us assume that FCI 1 sends data to controller 1 and controller 4. When FCI 1 sends data to controller 4, packages pass through segments 3, 4, and 5. When a package is sent to controller 1, it only passes through segment 1. If any of the FCIs attached to switch 2 (SW 2) or switch 3 (SW 3) sends data to controller 1 or controller 2, this would occupy segments 1 and 3. It is expected that the segments connecting the three switches will be the most loaded with data. What further affects the occupancy of the segments is the frequency of packages being sent. The higher the frequency, the higher the network occupancy.

The mathematical model. The package size is defined by

$$p_i = H + \sum_{k=1}^{s} x_{ik} DS_k$$

where

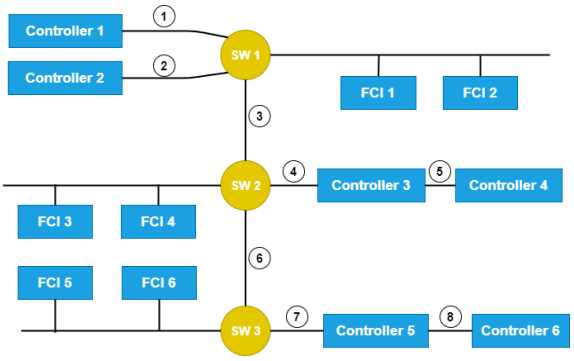


Fig. 5. Simple network

- $\sum_{i=1}^{n} x_{ik} = 1$ (each signal is only included in one package) with $x_{ik} = 1$ if signal k is put into package i and $x_{ik} = 0$ otherwise.
- p_i - the size of package i (bytes)
- H - the size of header and other constant (non-data) parts of the package
- DS_k - the data size of the signal k
- s - the number of signals inside the package
- n - the number of packages

Note that $p_i \leq 1500$ ensures that the total size of the loaded signals is not greater than the size of the package. Next, we try to optimize the network occupancy at a specific time interval (based on the repetitiveness of the signal activities). The relations below define segment occupancy as well as total network occupancy.

$$O_i = \sum_{j=1}^{n} \left\lceil \frac{T}{T_j} \right\rceil p_j, \; O_{tot} = \sum_{i=1}^{s} O_i$$

where

- O_i - Occupancy of the segment i
- O_{tot} - Total network occupancy
- s - The number of segments
- T - The time over which the network occupancy is observed
- T_j - Reading (packing) period of the package j
- $\left\lceil \frac{T}{T_j} \right\rceil$ - The maximum number of times package j is sent within a specified time interval
- p_j - Size of the package j
- n - The number of different packages that are sent through segment i

Packaging considerations. Consider next the same system of Fig. 5, where three controllers run each one task, with the characteristics illustrated in Table 2. The signals providing these datasets are attached to FCI1, and their ideal reading (and packing) periods are provided considering the Nyquist sampling theorem.

Table 2. Example of dataset assignment in FCI1

Task	Period (ms)	Controller	Dataset from sensors	Ideal reading period (ms)	FCI
T1	100	CTRL1	S1-S20	50	1
...	...	...	...	...	...
T3	100	CTRL3	S1-S20	50	1
T4	200	CTRL4	S15-S40	100	1
T5	300	CTRL5	S41-S70	150	1
...	...	...	...	...	...

Observe that, in the case of Controler 1 (CTRL1), all data does not fit into one package (as often is the case). Therefore, multiple packets are created, with the risk that some are not used to the maximum capacity. The packing mechanism advances through the dataset and packs the signals one by one. If the signal does not fit into the package, a new one is opened, and the packing procedure is continued. This packaging procedure applies the NF algorithm - and we use it as a *reference case*. Fig. 6 shows the result of the NF algorithm applied to the data sent from FCI1 to CTRL1.

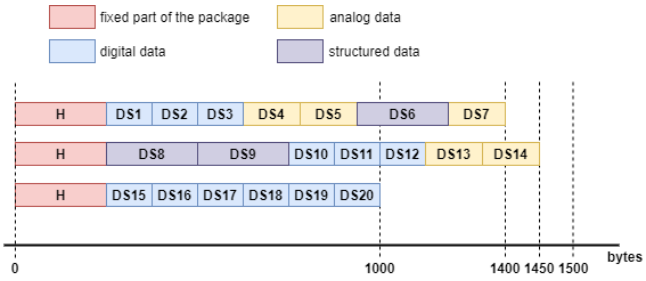


Fig. 6. Reference case packing procedure

Notice next the overlap in the data sampling for sensors S15 to S20, required basically every 100ms, by T3 and every 200ms, by T4 - hence read and ideally packed every 50ms.

If the respective data is packed in different packages (for T3 and T4, respectively), we can see that these datasets will travel through segments 3 and 4 at the same time, even though they contain the same information. This will result in an increase of the occupancy of these segments. However, data may be sent together in the same package to reduce network occupancy, with some timing, and / or package size penalties (in general, when periods don't match into some mathematical integer relation).

The drawback of such an approach is that controller 4 would, in addition to the requested data, also receive data intended for controller 3. Therefore, the occupancy of segment 5 will increase.

This example demonstrates the importance of trade-offs in this type of optimization problem. Using our algorithm, we will try to find a solution that minimizes the number of packages and thus the network occupancy.

The proposed approach. As seen above, often datasets from the same FCI are sent to different controllers at the same time. As a result, some network lines may become congested. Given that there are dozens of such FCIs in the network, the load on the respective segments becomes enormous. One way to influence network occupancy is to consider merging data from different datasets into the same package. If data intended for different controllers share most of the way through the network, putting them together instead of in separate packets can reduce network occupancy, with some potential minor drawbacks - also the occupation of leaf branches of the network would increase. This is a trade-off that must be taken into account when solving this issue.

The packing algorithm in the reference case leaves the packages unfilled. This can lead to unnecessary splitting of the dataset into multiple packages.

One interesting feature of task execution is that it does not require to run on the newest values of sensor data every time. For each signal (and corresponding task), it is specified how many times in a row data can be lost (or not sent in the network). We call this parameter the number of *allowed misses*, referred as such henceforth. Since we are considering an ideal network, in which data cannot be lost, we will use this parameter to intentionally not package some signals. In this way, network occupancy can be significantly reduced.

FCIs represent nodes in this network-centric system. They calculate and create tables according to which they send data to the network. Therefore, the output of the algorithm should provide a schedule for sending packets as well as their content.

The algorithm. The goal of the algorithm is to minimize the number of packets in the network. We describe here the implementation of the algorithm.

Depending on the need of the individual tasks, the signals within the FCI are collected in datasets. The FCI has information on the ideal reading period of these datasets. Based on this information, we can calculate a hyperperiod of reading (packing):

$$hyperperiod = LCM(T1, \ldots, Tn),$$

where $T1, \ldots, Tn$ are the ideal reading periods of respective datasets.

The hyperperiod represents the interval at which the entire schedule of packing and sending in the FCI is repeated. During this time, each dataset will be packed at least once and sent to the controller. It also covers all cases of sending different datasets at the same time. Therefore, the algorithm will calculate and use this interval in each FCI to determine the respective package-transmission schedule. Using the data in table 2, the hyperperiod of FCI1 is:

$$hyperperiod = LCM(50, 100, 150) = 300$$

Then we can determine time moments within hyperperiod in which each of data sets will be packed.

$$\begin{aligned} t_{S1-S20} &= [50, 100, 150, 200, 250, 300] \\ t_{S15-S40} &= [100, 200, 300] \\ t_{S41-S70} &= [150, 300] \end{aligned}$$

At any given time, the algorithm will attempt to pack the data so as to reduce network occupancy. The algorithm consists of two stages:

1. Determine all possible combinations of packing data for different controllers into the same package.

2. Perform bin packing algorithm to put as much data in as few packages as possible, taking into account the parameter of allowed misses.

The first part of the algorithm is illustrated in Fig. 7. For each time in which the packing and sending of data in the network take place, the algorithm performs all possible combinations of packing. The number of combinations is determined by the number of different datasets being sent at that time point. Since we consider a medium-sized plant model and FCIs do not supply a large number of different controllers, it is possible to test all combinations and thus find optimal solution. In large-sized plant models and wider networks, this exhaustive search can lead to the algorithm running for too long. In that case, one possible approach is to test a limited number of randomly selected combinations.

The second part of the algorithm addresses packing. A modified BFD algorithm has been implemented for this purpose. Depending on the size, the signals ready to be packed at a certain point in time are arranged in descending order. Instead of packing all the signals, the algorithm first checks if it is necessary to send a signal and then puts it in a packet. Depending on the requirements from tasks, some signals may be omitted from the packet for several consecutive times.

After packing, the we compute the cost function:

$$f = \sum_{j=1}^{n} \sum_{i \in P(j)} L_j p_i$$

where n is the number of different paths, and L_j is the number of line segments that belong to the path $P(j)$ from FCI to the controller.

The cost function basically calculates the network occupancy caused by a particular packet type. The type of package with the lowest value of the cost function is chosen as a best solution. This procedure is repeated for each FCI.

Package loss handling. The presented algorithm calculates the packing schedule for each FCI before commissioning. The resulting schedule implies that each packet will be sent at a specific time of sending and thus allow the tasks to operate with new data when needed. Losing a packet can result in tasks operating on old data or even task failure. In time and safety-critical systems, this temporary disruption of the proper operation is a potentially serious hazard. In addition, the network occupancy increases due to the data to be re-transmitted to the controllers. The classic approach to addressing this issue is to resend the lost packets in the next sending period. Our strategy for package loss handling is based on the packaging algorithm itself. Once a lost packet is identified, the FCI will associate it with packages that are sent at the next sending time according to a schedule. All datasets contained in the packages are re-grouped into new packages in the manner described in the algorithm. The FCI packing schedule is then updated for the next sending time. There are many advantages to this approach over classic re-sending.

Consider the case where the same packet is sent in two consecutive time moments as shown in Fig. 8. The packet P1 that is lost at time kT is marked in red, where T is the packet sending period. At the moment $(k+1)T$, the result of both approaches is shown, where the classical approach is marked in yellow and our strategy in blue.

On can notice that re-sending the packet will result in two of the same packets being sent to the same location (the re-sent packet is highlighted in green). The proposed algorithm identifies the same data in two packets and decides not to send them again. In this way, network congestion can be reduced.

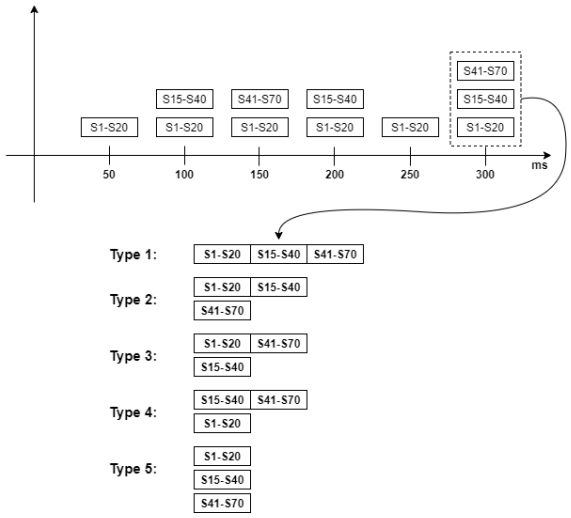


Fig. 7. Representation of the first phase of the algorithm

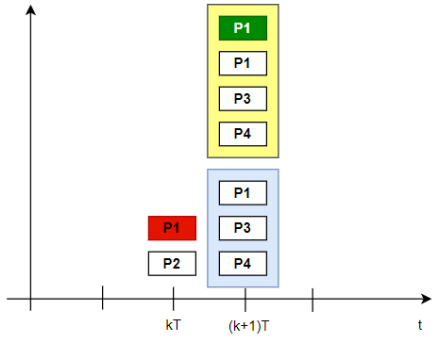


Fig. 8. Re-sending the same package.

Some systems are designed in such a way that devices that collect and package data send new data at each time of packaging. This avoids re-sending the packages and prevents the issue presented above. The disadvantage of such a design is that data is sent to controllers/tasks even when they do not need fresh data.

Apart of not sending the same data, the algorithm tries to merge datasets into the same packets. This is illustrated in Fig. 9, where Fig. 9.a shows part of the schedule of one FCI for two consecutive time moments, while Fig. 9.b and 9.c compare different approaches after losing a package with dataset DS1.

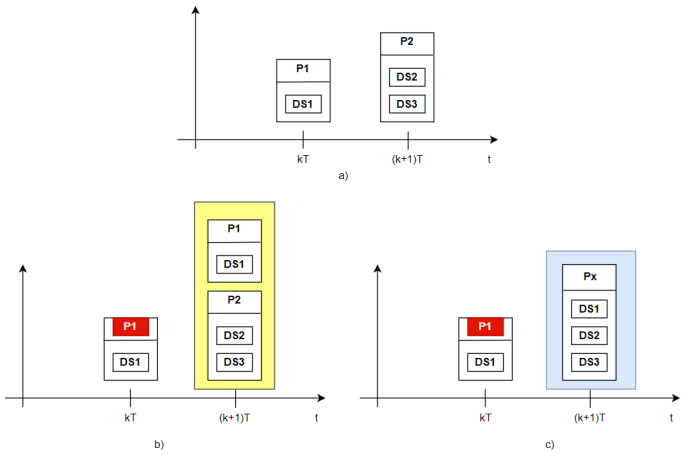


Fig. 9. a) No package loss case. b) "Re-send" approach after package loss. c) Proposed solution with merging datasets into a new package.

If calculated as a better option the algorithm will create a new package and not just re-send the lost one. The newly created package contains the data of the lost package and the packages scheduled at the time of sending. The schedule is being updated and the goal is to reduce network occupancy.

There is one drawback to this approach. If the new packet is lost, it will result in the loss of more data than in the case of just re-sending the packet. Also more controllers will be affected and left without fresh data. However, taking into account that the expected packet loss in the network is less than 10% the observed impact of this drawback is negligible.

In real-time systems, it is very important that tasks are executed using fresh data. Missing a deadline can lead to task failure and, in some cases, catastrophic consequences. Consider the case of 2 packages, P1 and P2 sent with different periods $T1$ and $T2$. Suppose a task that receives package P1 executes also with period $T1$. For the task to be executed in real-time it is necessary that a new data contained in package P1 is delivered before task starts to execute. In an ideal network that would always be fulfilled. However, in case package P1 gets lost it must be sent again before time period $T1$ expires. Using

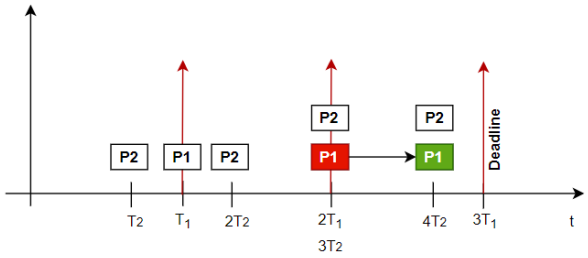


Fig. 10. Package delivery in time

our approach, packet P1 will be sent the next time any packet is sent from the FCI. In the case shown in Fig. 10 packet P1 is sent together with packet P2 before the deadline.

In general, it is guaranteed that package Pi with period Ti will arrive on time in case of loss if there is a package or set of packages that have a sending period less than Ti.

Network and devices. The algorithm is implemented in the Python 3.8 with OOP paradigm. We used Python classes to model the system.

For network and device modeling, we created classes: Segment, Switch, Controller, FCI. Each of these elements has an ID represented as an integer and which serves to distinguish elements in the network. Segments are also specified with two nodes, which can be either switches or controllers. Each controller has a list of tasks assigned to it and the ID of the switch to which it is directly or indirectly connected via other controllers. Switches contain a list of all devices (FCIs and controllers) connected to them.

Each instance of the FCI class, in addition to the ID, also contains datasets it creates after reading the sensors. The list of packages and their contents, as well as the schedule for sending packages, are also filled in after the execution of the algorithm. FCIs later use this information for network routing purposes.

Package, Signal, Task, Dataset. For the modeling of other important elements, we created the following classes.

A *Package* instance is defined with its size, a list of signals it contains, its packing period, and a list of paths to intended controllers through a network. Methods are defined for adding a signal into the package and assigning a packing period.

Classes *Signal* and *Task* have their IDs represented as integers, which serve to distinguish class instances. Signals are further specified with a number of allowed misses, and its size.Each instance of the *Task* class, in addition to the ID, contains its period, its execution time, and a dataset of signals the task processes.

A *Dataset* instance contains the period, a list of signals in the dataset, and FCI and controller to which that dataset is allocated. There is also a list of time moments in which the dataset should be packed during the FCI hyperperiod and a path through which the dataset propagates in its packet from FCI to the controller that requires that dataset.

The above contribute, together with algorithm specific procedures, to the realisation of a small tool, with the interface shown in Fig. 11. At this moment the files used for input are hard-coded. The application expects the introduction of the execution time, and can run both the reference case (NF algorithm for the ideal network case without data

losses) and also the proposed solution for both the ideal network case and the case with data losses, calculating the network occupancy and allowing saves into Microsoft Excel.

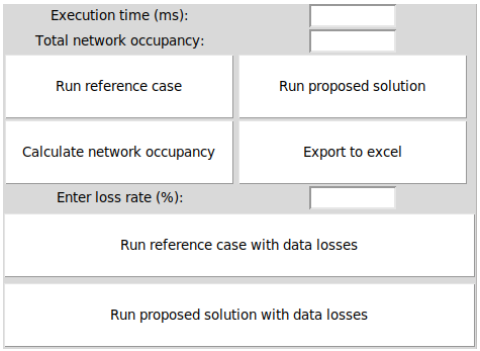


Fig. 11. The application GUI.

5. Simulation Results

To validate our working example, we apply the algorithm on a medium size plant use case, containing 2400 sensors (and associated signals), 80 control tasks, 24 FCIs and 10 controllers.

The input data is fed to the algorithm in the form of Excel datasheets. Three sheets correspond to 3 types of network devices (FCIs, controllers, and switches), and two sheets correspond to signals and tasks. The program reads data from tables and initialize class instances with needed data. Once all needed class instances are created, the optimization procedures can run. We organize the following based on 2 perspectives of the network operation.

1. *ideal*: No packages are lost. This is meant to help us create a picture of what benefits we can extract from our approach without a more complex context;
2. *real*: A "normal" loss of packages is considered and implemented by the simulation tool.

I both the above situation, the input data must be entered correctly in tables in the required format for the algorithm to work as expected. The 2^{nd} perspective above requiring registration of the additional information concerning the percentage of data loss across the network, which we modeled as an input to our graphical tool.

As shown in Fig. 11, a user can input the desired loss percentage as an integer. Below that are the buttons for running reference and our approach considering package losses.

Ideal network operation. The algorithm is tested on three different topologies (T_1. T_2, T_3), illustrated in Fig. 13. In all topologies the network is composed of 13 segments. The values of input data used for evaluation are:

– Allowed misses - random integer in range $[1, 5]$ for every signal

- Task period – period of the task in ms. Random integer between 100 and 500 with the step of 50.
- Task execution time – expressed in ms. Random number between 10 and 50 with the step of 10.
- Signals reading period – an ideal reading period of the set of signals allocated to the task expressed in ms. Defined as one-half of the task period signals are allocated to.

We chose an execution time of 9s, because it is the largest hyperperiod of all used FCIs and represents the shortest time unit after which the packing of all packets is repeated. The output of the algorithm consists of:

- The time moments in which packages should be packed and sent for each FCI
- The content of the packets which are packed and sent in computed time moments for each FCI
- The network segment occupancy
- The total network occupancy

Both the reference case and the proposed solution are executed, and results are exported to Excel datasheets, each file consisting of $R + 1$ sheets. First R sheets represent time moments, sizes of the packets, and packets' content that are sent from the R FCIs in the network (one sheet per FCI). The last sheet contains the network occupancy for every FCI and total network occupancy as a sum of occupancy of each network segment. An example of the packets in the datasheet is given in Fig. 12.

Time moments	Number of packets	Packet 1 size (B)	Packet 1 signals
75	1	585	[3, 4, 14, 2, 6, 7, 8, 9, 10, 11, 12, 1
100	1	582	[94, 96, 75, 77, 78, 81, 83, 84, 85,
150	2	585	[3, 4, 14, 2, 6, 7, 8, 9, 10, 11, 12, 1
175	1	843	[14, 21, 22, 28, 30, 10, 11, 12, 17,

Fig. 12. Sending schedule of packages and their content

Results for each topology are shown in Table 3, considering the packet header size 33B. The occupancy in the proposed solution is lower than the occupancy in the respective reference. The allowance and the selection of "no-send packages" is application and system specific, and is defined by system owners or designers - see also Table 5, further down.

Considering package losses. The algorithm is tested on topology 1 (Fig. 13) of the medium-sized plant without the allowance and selection of "no-send packages", described by the parameter "Allowed misses", as we focus our attention here on strategies to deal with package losses leaving aside the impact and discussion of different network topologies and allowance of "no-send packages" for the case of the ideal network.

As shown in Fig. 11, a user can enter loss percentage, after which one of the algorithms ("reference case" or "proposed approach" - described in detail in the previous section) is called. The lost packets are chosen randomly from the sending schedule created by the proposed solution without data losses until the limit of lost packages is reached, defined by

Table 3. Network occupancy relative reduction, on each topology (T_1, T_2, T_3).

	No misses allowed (%)			**Misses allowed** (%)		
Segments	T_1	T_2	T_3	T_1	T_2	T_3
Segment 1	0.283	0.283	0.283	68.145	68.145	68.145
Segment 2	0	0	0	66.383	66.336	66.383
Segment 3	0	0	0	70.732	70.650	60.572
Segment 4	0.132	0	0	67.798	68.592	68.592
Segment 5	0.010	0.019	0.025	68.178	67.707	66.267
Segment 6	0.124	0.124	0.124	72.184	71.752	71.752
Segment 7	0.064	0.064	0.064	69.564	69.149	69.149
Segment 8	0	0	0	66.035	66.058	66.059
Segment 9	0	0	0	63.129	63.123	70.321
Segment 10	0.020	0.029	0.142	68.852	67.143	66.498
Segment 11	0.273	0.273	0.273	67.484	67.680	67.680
Segment 12	0	0	0	71.220	71.227	71.227
Segment 13	0	0	0	62.292	62.292	62.292
Total network	0.055	0.039	0.052	67.723	67.465	67.282

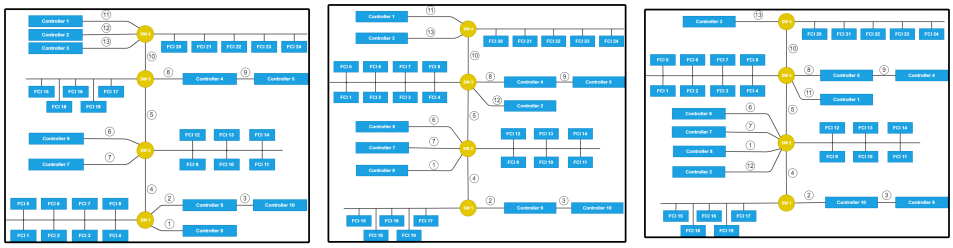


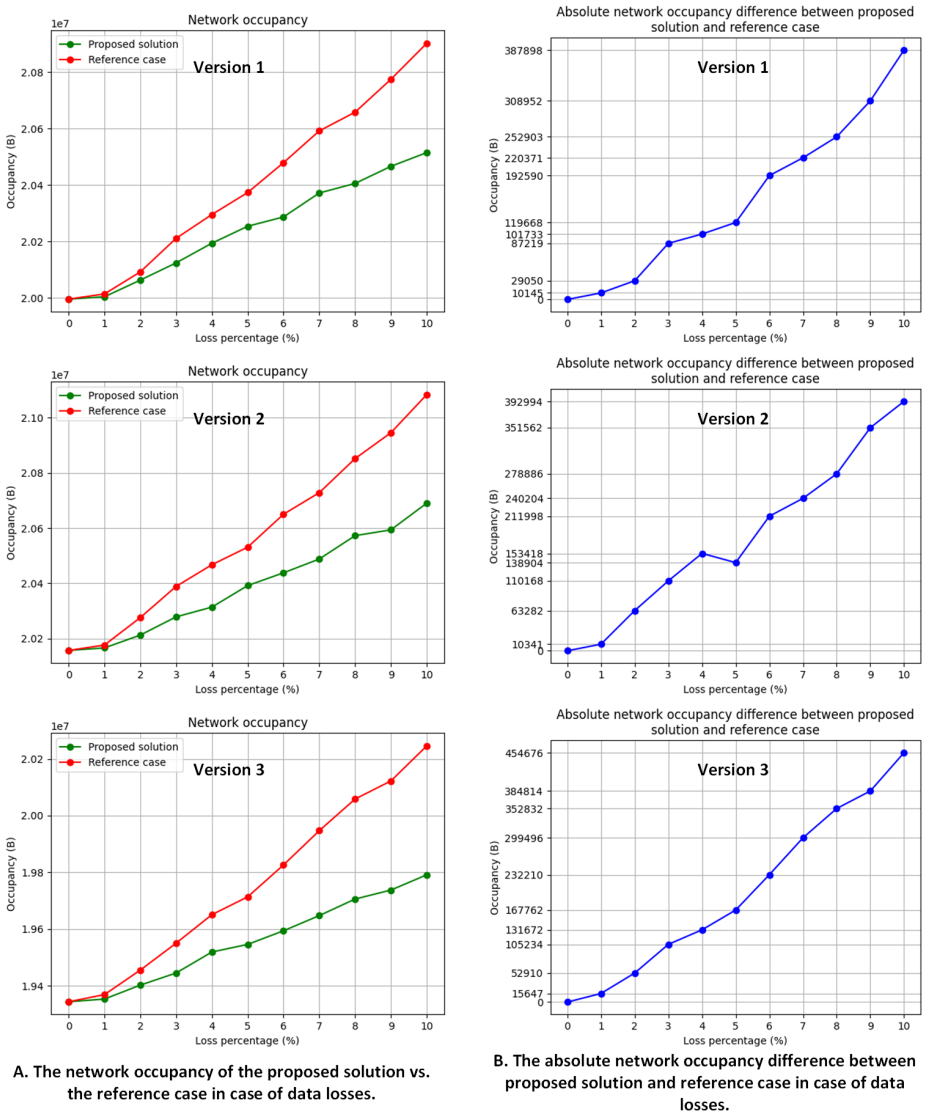
Fig. 13. Topologies

the entered loss percentage. Consequently, the calculation occupancy of network segments and total network is computed, and results are exported together with the packing schedule for every FCI as an Excel file.

An acceptable (depending on the application) loss rate in networks should be below 10%. With respect to that, we run our experiments with integer loss percentage rates from 1% to 10% including both limits. For every loss percentage rate from the closed interval $[1\%, 10\%]$ we run reference case and proposed solution 30 times each so that we have enough samples for treating results as a Gaussian distribution model. For every particular sample, we calculate occupancies of segments and the total network. Here we present averaged results. We test results on three different versions of the topology 1 (Fig. 13). The first version is one also used in the experiments without data losses, while the other two are created by different allocations of tasks to controllers, as discussed further down.

We plot in Fig. 14 the network occupancy change with respect to loss percentage for both the reference case and all the discussed versions. The right side column in the same Fig. shows the absolute network occupancy difference change with respect to loss per-

centage between the proposed solution and the reference case for three different versions of topology 1 (Fig. 13).



A. The network occupancy of the proposed solution vs. the reference case in case of data losses.

B. The absolute network occupancy difference between proposed solution and reference case in case of data losses.

Fig. 14. Simulation results considering packet losses. Column A: Network occupancy of the proposed versions and the reference case. Column B: Absolute network occupancy difference between the proposed versions and the reference case.

Impact of task allocation. In the following, we observe the task allocation impact on the network occupancy, in the context of topology 1. We consider the impact on the already

optimized solution, which we use as a reference. Differently from the already analyzed system, we allocate now the tasks such that on any given controller, there are either tasks which process signals with at least two different reading periods (if these are among the low values), or the tasks process relatively many signals with high reading periods, as the system *version 2*. The *version 3* of the system has the tasks allocated based on their respective signal reading period.

We run the simulation of the system versions, both in the ideal situation as well as considering losses, and the results are presented in Fig. 15.

Segments	Network occupancy per segments (B)		
	Version 1	Version 2	Version 3
Segment 1	988965	537603	891954
Segment 2	1264839	2343096	371304
Segment 3	723399	1689150	218160
Segment 4	3130377	3009135	2527425
Segment 5	4465401	3761547	4193820
Segment 6	320247	421734	555987
Segment 7	725145	539094	644262
Segment 8	2266524	613284	1453329
Segment 9	1342896	197775	776817
Segment 10	2364696	3528786	3656628
Segment 11	857331	2325630	916380
Segment 12	1054581	981789	1750530
Segment 13	490680	208656	1386711
Total network occupancy	19995081	20157279	19343307

Fig. 15. Network occupancy in the 3 analyzed system variants, no package losses.

6. Discussion

For studying how the different packing of input data affect the network occupancy, let us consider the reference case and the proposed solution without allowed misses, running on topology 1. Here, one dataset from FCI1 is assigned an ideal period of 75ms, while another one is assigned an ideal period of 175ms. FCI will pack and send signals only from these datasets at 525ms, because in that time moment there are no other signals from other datasets scheduled for packing. The difference between the reference case and the proposed solution is that:

- Reference case: two packets will be created at 525ms
- The proposed solution: one packet will be created at 525ms.

The algorithm tested both variants: with one packet and with two packets. In each variant, the algorithm calculated the network occupancy and decided in favor of the variant with less network occupancy. As the variant with one packet produced less network occupancy than the variant with two packets, as shown in table 4, the algorithm decided to proceed with the creation of only one packet.

The sum of packet sizes in the reference case is 1428 B, which is higher than the packet size in the proposed solution by 33 B, corresponding to one less header size.

Table 4. Reference vs. proposed solution.

Case	Time moment (ms)	Number of packets	Packets' size (B)
Nr. of packets			
Reference	525	2	843, 585
Proposed solution	525	1	1395
Packet utilization			
Reference	1125	6	1500, 1494, 552, 1332, 1431, 1221
Proposed solution	1125	6	1500, 1500, 546, 1332, 1500, 1152

Influence of the BFD algorithm on package utilization. The BFD algorithm is chosen as a heuristic approach in the proposed solution. It achieves better results than the NF algorithm used in the reference case. A modified version of the BFD algorithm is implemented: instead of packing all the signals, the algorithm first checks if it is necessary to send a signal and then places it in a packet.

Let us consider the reference case and the proposed solution of the described use case, without allowed misses, running on topology 1. At time 1125ms, at FCI18, we have the packet utilization as shown in Table 4. In the proposed solution, three packages are fully utilized, while only one package is fully utilized in the reference case. Although the number of packages in this example is not reduced, the BFD algorithm has been shown to maximize package utilization. In larger networks, where a larger number of packages are sent at the same time, by applying BFD, we can also expect a reduction in the number of packages. This also shows us that using "plain" bin packing algorithms is not sufficient. They require modification and combination with other algorithms to achieve a significant reduction in packets in the network.

Impact of the *Allowed Misses* parameter. We observe here the packing of the S1-S20 dataset at three consecutive time points. The size and number of allowed misses are specified for each signal and shown in Table 5. Since the sending period of this dataset is 75ms, we are interested in time moments of 75ms, 150ms, and 225ms. Table 6 shows both the content and timing of packets within FCI1 when data freshness at controllers is not considered (always required to have new data), and when the allowed misses are taken into account.

Table 5. Allowed misses and signal (S1-S20) data-size

Signal	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Allowed misses	5	3	4	3	1	3	4	3	2	5	4	4	1	3	4	5	4	2	4	1
Size	12	15	105	105	12	15	15	15	15	15	15	15	12	105	12	12	15	12	15	15

When the allowed misses are taken into account, at 150ms, the new values of some signals from the dataset are not sent, as planned by the algorithm. At such moments, the affected tasks will process the signal values previously received and saved in specific registers of the controllers. At 225ms, only three fresh signal values from the dataset (S5, S13, S20) are sent. The reason is that these signals are allowed to skip transmission only

once, as the receiving tasks can use an old value of the signals for one cycle only (see Table 5).

Table 6. Part of FCI1 packing schedule.

Time moment	Nr. of packets	Packet 1 size	Packet 1 signals
No misses allowed			
75	1	585	3,4,14,2,6,7,8,9,10,11,12,17,19,20,1,5,13,15,16,18
150	2	585	3,4,14,2,6,7,8,9,10,11,12,17,19,20,1,5,13,15,16,18
225	1	585	3,4,14,2,6,7,8,9,10,11,12,17,19,20,1,5,13,15,16,18
Misses allowed			
75	1	585	3,4,14,2,6,7,8,9,10,11,12,17,19,20,1,5,13,15,16,18
150	1	405	71,72,73,68,69,74,70
225	1	72	20,5,13

Even though we observed a very short time interval, we can conclude that this feature significantly reduces the network occupancy.

Influence of topology on network occupancy. The relative placement of elements in the network affects the occupancy of some segments in the network. Analyzing the obtained results, we can see that the segments that connect the switches are the most loaded. In our network model, these are segments 4, 5, and 10. This is expected because switches together with segments create a fundamental network tree. All other segments with controllers represent smaller branches of this tree. The idea is that just by moving certain elements to other switches tries to reduce the amount of data on these segments. It should be emphasized that there are also specific physical limitations among the elements in the network. The sensors are usually located in predetermined locations and cannot be deployed. Therefore, clusters of FCIs are formed that collect data from a specific group of sensors. FCIs belonging to one cluster cannot be moved to others. Therefore, moving any FCI to another location in the network in other topologies involves moving the entire cluster.

In topology 2, the occupancy of segments 4 and 10 is significantly reduced (see Table 7). This reduction is due to the shift of controllers 2 and 8 to switches 2 and 3, and the replacement of FCIs clusters on switches 1 and 3. In this way, the number of elements on external switches is reduced, thus reducing the number of devices sending or receiving data through segments 4 and 10. Although a significantly lower occupancy of segments 4 and 10 was achieved, the occupancy of segments 5 increased. This trade-off may be unacceptable because segment 5 is the busiest segment in the network.

Table 7. Occupancy of segments 4, 5, and 10: T_1 vs. T_2

Segment	T_1	T_2
Segment 4	1003725 B	868707 B
Segment 5	1421127 B	1463913 B
Segment 10	736698 B	517059 B

Subsequent movements of the controller in topology 3 resulted in a reduced network occupancy of 14% compared to topology 1. With an additional reduction in the payload in segment 4, a significant difference is also noticeable in segment 5 (see Table 8).

Table 8. Difference between T_2 and T_3 in occupancy of segments 4,5, and 10

Segment	T_2	T_3
Segment 4	868707 B	868707 B
Segment 5	1463913 B	1162998 B
Segment 10	517059 B	442233 B

Adding new segments or switches can further reduce the payload in the network, but this would significantly change the network structure, and thus the results would not be comparable. The cost of new elements in the network should also be taken into account, and we leave that as future work.

Considering package losses. Analyzing the plots in Fig. 14 A and B, it results that the proposed solution achieves lower network occupancy than the reference case with data losses. Additionally, it is visible that the absolute difference between the proposed solution and the reference case increases with the loss percentage - as expected. Table 9 presents percentage-wise the relative differences between the proposed solution and the reference case. The relative difference increases monotonically with the loss percentage increase. Even if these relative differences are not high percentage-wise, given the huge amount of data in real industrial networks, there will still be achieved a high absolute reduction in network occupancy.

Impact of task allocation. We refer in the following to the numbers presented in Fig. 15 and in Table 9. We notice, thus, that *version 3* of the task allocation offers a 3+% decrease of the overall traffic, while *version* 2 increases the traffic by a 0.8%. At the same time, traffic on some segments vary in much larger amounts.

It is important to remind here that the allocation versions are selected on a semi-random basis. The fact stands to only show that a good algorithm for allocation of tasks, would be a good complement to the work described in this report.

Table 9. Network occupation: *Version 1* (reference) vs. *Version 2* and *Version 3*.

	Differences in network occupation (%)													
Segment	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
Version 2	-45,6	85,2	133,5	-3,9	15,8	31,7	-25,7	-72,9	-85,3	49,2	171,3	-6,9	-57,5	0,8
Version 3	9.8	-70,6	-69,8	-19,3	-6,1	73,6	-11,2	-35,9	-42,2	54,6	6,9	66,0	182,6	-3,3

7. Conclusions

We addressed here a multi-constrained network occupancy optimization problem, where a set of signals from different sensors have to be packed into a set of network packages.

We proposed a heuristic approach over two phases. The algorithm first determines all possible combinations of packing data for different destinations into the same package. The second phase is based on a modification of the BFD algorithm. Based on the tasks' requirements on data freshness, the algorithm verifies the necessity of sending a signal, and only if so, it assigns the signal to a package. A reference case was also implemented, according to current industrial practices. The packaging procedure is based on the NF algorithm. The use of the BFD algorithm in the proposed solution shows the increase in package utilization while merging datasets into same packets has reduced network occupancy. The obtained results showed that network occupancy could be significantly reduced by bringing the end nodes closer to the sources. This is not always possible, as controllers may require data from variously placed FCIs. We showed that our approach achieves lower network occupancy in case of data losses and that different task-controller allocations can additionally increase or decrease network occupancy. A good algorithm for allocation of tasks would a good addition to our algorithm.

Future work. Our approach currently did not include several aspects of networked communication, such as propagation time and communication from controllers to actuators. Future research will be conducted to include these aspects. Note also that we only considered one-way communication, but a bi-directional perspective is necessary. However, the same approach can be extended to cover controllers as sources and FCIs as destinations - though with a reduced set of options for packing, as the number of signals at controllers (as sources) is considerably lower than what we see in the other direction. When two-way communication is considered, links between FCIs and switches should be modeled as segments.

References

1. OPC Unified Architecture, Specification, Part 14: PubSub. `https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-14-pubsub` (2018), [Online; accessed 13-May-2020]
2. Industrial Communication (2020), `https://www.pepperl-fuchs.com/global/en/classid_6416.htm`
3. Alirezazadeh, S., Alexandre, L.A.: Dynamic task allocation for robotic network cloud systems. The Intl. Conf. on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking pp. 1221–1228 (2020)
4. Aziz, H., Chan, H., Cseh, Á., Li, B., Ramezani, F., Wang, C.: Multi-robot task allocation – complexity and approximation. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. pp. 133—-141 (2021)
5. Bernhard, K., Vygen, J.: Combinatorial optimization: Theory and algorithms. Springer, Third Edition, 2005. (2008)
6. Coffman, Jr, E.G., Garey, M.R., Johnson, D.S.: An application of bin-packing to multiprocessor scheduling. SIAM Journal on Computing 7(1), 1–17 (1978)
7. Dósa, G.: The tight bound of first fit decreasing bin-packing algorithm is $FFD(I) \leq 11/9OPT(I) + 6/9$. In: International Symposium on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies. pp. 1–11. Springer (2007)
8. Dósa, G., Sgall, J.: First fit bin packing: A tight analysis. In: 30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2013)

9. Dósa, G., Sgall, J.: Optimal analysis of best fit bin packing. In: International Colloquium on Automata, Languages, and Programming. pp. 429–441. Springer (2014)
10. Fleszar, K., Hindi, K.S.: New heuristics for one-dimensional bin-packing. Computers & operations research 29(7), 821–839 (2002)
11. FMS, P.: Packet loss: problems, causes and solutions in 2020 (2020), https://pandorafms.com/blog/packet-loss.
12. GAREY, M.R., JOHNSON, D.S.: Complexity results for multiprocessor scheduling under resource constraints. In: Proceedings of the 8th Annual Princeton Conference on Information Science and Systems (1974)
13. Hopper, E., Turton, B.C.: An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. European Journal of Operational Research 128(1), 34–57 (2001)
14. Jiang, Y., Zhou, Y., Li, Y.: Network layer-oriented task allocation for multiagent systems in undependable multiplex networks. In: The 25th International Conference on Tools with Artificial Intelligence. pp. 640–647. IEEE (2013)
15. Kim, S.I., Kim, J.K., Ha, H.U., Kim, T.H., Choi, K.H.: Efficient task scheduling for hard real-time tasks in asymmetric multicore processors. Lecture Notes in Computer Science 7440, 187–196 (2012)
16. Leinberger, W., Karypis, G., Kumar, V.: Multi-capacity bin packing algorithms with applications to job scheduling under multiple constraints. In: Proceedings of the 1999 International Conference on Parallel Processing. pp. 404–412. IEEE (1999)
17. Liu, D., Tan, K.C., Huang, S., Goh, C.K., Ho, W.K.: On solving multiobjective bin packing problems using evolutionary particle swarm optimization. European Journal of Operational Research 190(2), 357–382 (2008)
18. Lo, V.M.: Heuristic algorithms for task assignment in distributed systems. IEEE Transactions on computers 37(11), 1384–1397 (1988)
19. Loh, K.H., Golden, B., Wasil, E.: Solving the one-dimensional bin packing problem with a weight annealing heuristic. Computers & Operations Research 35(7), 2283–2291 (2008)
20. Mao, W.: Tight worst-case performance bounds for next-k-fit bin packing. SIAM Journal on Computing 22(1), 46–56 (1993)
21. Moneer, O.: Bin packing problem under multiple-criterions, https://www.cse.huji.ac.il/~ai/projects/old/binPacking2.pdf, [Accessed: 9-July-2020]
22. Pisinger, D., Sigurd, M.: The two-dimensional bin packing problem with variable bin sizes and costs. Discrete Optimization 2(2), 154–167 (2005)
23. Postawka, A., Koszałka, I.: Task allocation within mesh networks: Influence of architecture and algorithms. In: Selvaraj, H., Zydek, D., Chmaj, G. (eds.) Advances in Intelligent Systems and Computing. vol. 366, pp. 869–875. Springer, Cham (2015)
24. Salimi, M., Majd, A., Loni, M., Seceleanu, T., Seceleanu, C., Sirjani, M., Daneshtalab, M., Troubitsyna, E.: Multi-objective optimization of real-time task scheduling problem for distributed environments. In: Proceedings of the 6th Conference on the Engineering of Computer Based Systems. pp. 1–9 (2019)
25. Saraiva, R.D., Nepomuceno, N., Pinheiro, P.R.: A layer-building algorithm for the three-dimensional multiple bin packing problem: a case study in an automotive company. IFAC-PapersOnLine 48(3), 490–495 (2015)
26. Schoneveld, A., De Ronde, J., Sloot, P.: On the complexity of task allocation. Complexity 3(2), 52–60 (1997)
27. Sheng, L., Xiuqin, S., Changjian, C., Hongxia, Z., Dayong, S., Feiyue, W.: Heuristic algorithm for the container loading problem with multiple constraints. Computers & Industrial Engineering 108, 149–164 (2017)
28. Wang, S., Dang, Y., Wu, J.: How task allocation strategy affects team performance: A computational experiment. Journal of Systems Science and Systems Engineering 27, 665–676 (2018)
29. Wang, Z., Nip, K.: Bin packing under linear constraints. Journal of Combinatorial Optimization 34(4), 1198–1209 (2017)

30. Y., C., Lu, L., Yu, X., Li, X.: Adaptive method for packet loss types in iot: An naive bayes distinguisher. Electronics 8(2), 134 (2019)
31. Yu, T., Sekar, V., Seshan, S., Agarwal, Y., Xu, C.: Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In: Proceedings of the 14th ACM Workshop on Hot Topics in Networks. pp. 1–7 (2015)
32. Zhang, B., Dou, C., Yue, D., Zhang, Z., Zhang, T.: A packet loss-dependent event-triggered cyber-physical cooperative control strategy for islanded microgrid. Trans Cybern 51(1), 267–282 (2021 Jan)

Amar Halilovic is currently working as a research assistant and Ph.D. student in Explainable AI and Robotics at the Institute of Artificial Intelligence, Ulm University, Ulm, Germany. He graduated from the Faculty of Electrical Engineering, Department of Automatic Control and Electronics, University of Sarajevo, Bosnia and Herzegovina in 2018 with a BS Degree and in 2020 with an MS Degree. Additionally, he holds an MS degree in Computer Science with specialization in Intelligent Embedded Systems from the Mälardalen University, Västerås, Sweden in 2020. The areas he is most interested in are AI, Robotics, Computer Vision, and Embedded Systems.

Nedim Zaimovic is currently working as an embedded systems engineer in the rolling stock industry. He graduated from the Faculty of Electrical Engineering, Department of Automation Control and Electronics, University of Sarajevo, Bosnia and Hercegovina in 2019 with a B.S degree. In addition to his B.S degree in Electrical Engineering, he holds a M.S. degree in Computer Science with specialization in Intelligent Embedded Systems from the Mälardalen University, Västerås, Sweden in 2020. The areas he is most interested in are control systems, real-time embedded systems, and machine learning.

Tiberiu Seceleanu is professor of Distributed Automation Systems at the Mälardalen University (MDU), Västerås, Sweden, since January 2020. He received his M.Sc. (1994) and Lic.Sc. (1995) degrees from the Polytechnical University in Bucharest, Romania, and the Dr.Tech degree (2001) from Åbo Akademi in Turku, Finland. He became a Docent at Univeristy of Turku, in 2007 in Computer Engineering Systems. A principal Scientist at ABB Corporate, Research (2007-2019) he became a docent at MDU in 2009. Current interests relate to dynamic reconfigurable systems (including networking), high levels of system design, hardware and software, machine learning and autonomous systems.

Hamid Reza Feyzmahdavian received the B.Sc. and M.Sc. degrees in electrical engineering with specialization in automatic control from Sharif University of Technology, Tehran, Iran, in 2005 and 2008, respectively, and the Ph.D. degree in automatic control from the KTH Royal Institute of Technology, Stockholm, Sweden, in January 2016. He is a Principal Scientist with the Control and Optimization Group, ABB Corporate Research Center, Västerås, Sweden. From 2016 to 2017, he was a Postdoctoral Researcher with the Department of Automatic Control, KTH Royal Institute of Technology. His current research interests include distributed optimization and machine learning.

Received: October 01, 2021; Accepted: September 01, 2022.

Computer Science and Information Systems 20(1):277–306 https://doi.org/10.2298/CSIS210707057X

Formalization and Verification of Kafka Messaging Mechanism Using CSP

Junya Xu[1], Jiaqi Yin[2,⋆], Huibiao Zhu[1,⋆] and Lili Xiao[1]

[1] Shanghai Key Laboratory of Trustworthy Computing,
East China Normal University, Shanghai, China
hbzhu@sei.ecnu.edu.cn
[2] Northwestern Polytechnical University, Xi'an, China
jqyin@nwpu.edu.cn

Abstract. Apache Kafka is an open source distributed messaging system based on the publish-subscribe model, which achieves low latency, high throughput and good load balancing. As a popular messaging system, the transmission of messages between applications is one of the core functions of Kafka. Therefore, the reliability and security of data in the process of message transmission in Kafka have become the focus of attention. The formal methods can analyze whether a model is highly credible. Therefore, it is significant to analyze Kafka messaging mechanism which describes the communication process and rules between each module entity in Kafka from the perspective of formal methods.

In this paper, we apply the process algebra CSP (Communicating Sequential Processes) and the model checking tool PAT (Process Analysis Toolkit) to analyze Kafka messaging mechanism. The results of verification show that the model caters for its specification and guarantees the reliability of messages in the normal communication process. Moreover, in order to further analyze the security of Kafka messaging mechanism, we add the intruder model and the authentication protocol Kerberos model and compare the verification results of Kafka messaging mechanism with or without the secure protocol Kerberos. The results show that the Kerberos protocol has improved the security of Kafka messaging mechanism in some aspects, but there are still some security loopholes.

Keywords: Distributed Messaging System, Kafka Messaging Mechanism, CSP, Formalization, Verification

1. Introduction

Since distributed messaging system provides an effcient and stable transmission channel for the realization of streaming calculation, data transmission and other functions, it has been widely used to capture and analyze large amounts of data in real time. A messaging system is responsible for the transmission of data among applications, and these applications only focus on the data rather than the datials of transmission. Data transmission in the communication process of distributed message system is based on a reliable message queue, which mainly has the following two modes: point-to-point and publish-subscribe [6,25]. In a point-to-point mode, a producer sends data to a queue and one or more consumers consume data from this queue in sequence. Each data can only be used once, i.e.,

⋆ Corresponding authors

when a consumer consumes a piece of data in this queue, this data is removed from the messaging queue. ActiveMQ, ZeroMQ and RabbitMQ [1,16] are well-known message queuing platforms. Unlike point-to-point, in a publish-subscribe mode, producers publish messages grouped into topics, while consumers can subscribe to one or more topics and consume all the data in these topic. In addition, the same piece of data can be consumed by multiple different groups of consumers, and the data will not be deleted immediately after consumption. Apache Kafka is a high-performance cross-language distributed messaging system based on publish-subscribe mode [8,24,26], which has been widely used by Internet companies such as Yahoo, Twitter, etc.

As a popular open source distributed messaging system, Kafka has the following advantages [8,26]. First, Kafka provides high throughput for both publishers and subscribers. It can produce about 250,000 messages (50 MB) per second and process 550,000 messages (110 MB) per second. Second, Kafka can be persisted. Messages are persisted to disks, so it can be used for bulk consumption, such as Extract-Transform-Load (ETL) and for real-time applications. At the same time, it prevents data loss by persisting data to disks and using replica mechanism. Third, it is easier for Kafka to expand outward. There are multiple producers, brokers and consumers, all of which are distributed. As a result, it can expand the machine without downtime. Fourth, the state in which messages are processed is maintained on the consumer side, not on the server side, so that it can be automatically balanced when message processing fails. Finally, Kafka supports both online and offline scenarios.

The formal methods are the research methods based on mathematical logic, which can verify and evaluate the reliability of the model through the specification, modeling and analysis of the model. Therefore, we consider to analyze whether the data in Kafka messaging mechanism is reliable from the perspective of formal methods. In this paper, we use the classical process algebra CSP [3,9,19] to give a formal model of Kafka messaging mechanism, and utilize the model checker PAT [10,13,17,20] to verify some important properties, including *Deadlock Freedom, Acknowledgement Mechanism, Parallelism, Sequentiality* and *Fault Tolerance*. Moreover, we introduce the intruder to simulate the attack behavior in the real network and introduce the authentication protocol Kerberos to improve the security of the Kafka messaging mechanism. We also model the intruder and Kerberos based on the original model to further analyze the security of Kafka.

The remainder of this paper is organized as follows. Section II gives a brief introduction to Kafka messaging system, the process algebra CSP and the model checking tool PAT. In section III, we model the core components in Kafka messaging system using CSP. At the same time, we adopt the model checking tool PAT to implement the constructed model and verify five properties. Moreover, we analyze the security of Kafka messaging mechanism by modeling the intruder and Kerberos and comparing the verification results of the constructed model with or without the Kerberos protocol in Section IV. Finally, we conclude this paper and make a discussion on the future work in Section V.

2. Background

In this section, we start with an overview of Kafka messaging system. At the same time, we also give a brief introduction to process algebra CSP and model checking tool PAT.

2.1. Kafka messaging system

In this paper, we focus on Kafka messaging mechanism which describes the process of communication among producers, consumers and other components as shown in Fig.1.

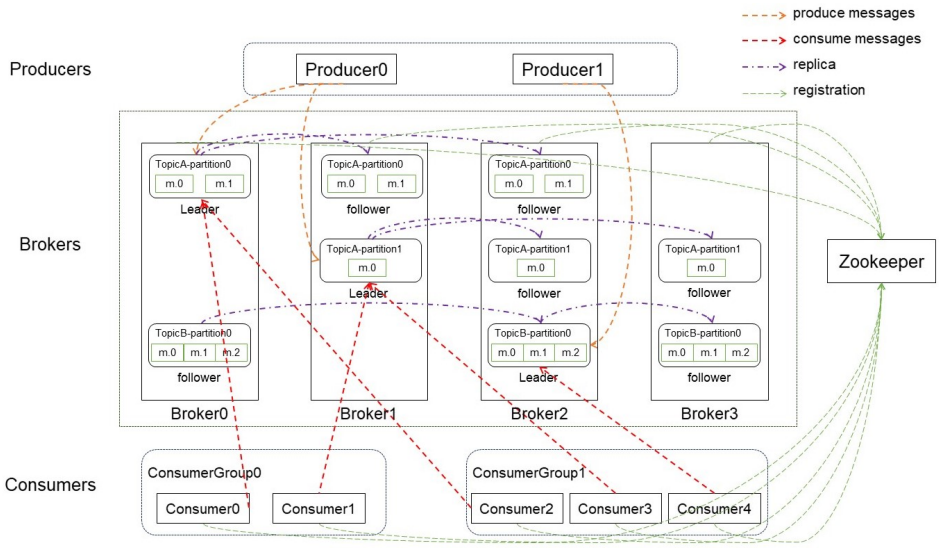


Fig. 1. The Kafka Messaging System

Before we introduce the Kafka messaging system, we first need to know a few common terms [16] in Kafka as follows:

- **ZooKeeper**: ZooKeeper helps Kafka store and manage the information of Kafka cluster.
- **Producer**: Producers consist of applications and are publishers of data, primarily sending messages to Brokers in Kafka.
- **Broker**: The Kafka messaging system contains one or more brokers that save data from producers and provide these data to consumers.
- **Topic**: Each message published to Kafka has a category called Topic. Producers and consumers only need to specify the topic of messages to publish and consume the data, regardless of which brokers data is stored on.
- **Partition**: A topic can be divided into multiple partitions, each of which is an ordered queue, and each message in a partition is assigned an ordered id. Kafka only guarantees that messages are sent to the consumer in the order of one partition, not the order of the whole of a topic.
- **Replica**: Replica is for backup to ensure that data is not lost when a broker in Kafka fails and Kafka continues to work. Each partition of a topic has several replicas including one leader and several followers.

- **Leader**: The leader is a 'master' replica of multiple replicas of each partition, and is the object on which the producer sends the data and the object on which the consumer consumes the data.
- **Follower**: The follower is the 'slave' replica of multiple replicas of each partition, and synchronizes data from the leader in real time to keep the data synchronized with the leader.
- **Consumer Group**: Each consumer group consists of multiple consumers subscribing to the same topic. In Kafka, the data of the same partition can only be consumed by one consumer within the group, but consumers in other groups still use this data.
- **Group Coordinator**: There is only one group coordinator for a consumer group. It needs to manage the load balancing among the consumers in this group, and sends partitioning strategy to all the consumers in that group.
- **Consumer**: It is the consumer of messages and pulls messages from the subscribed topic.

Based on the above concepts, we introduce Kafka messaging mechanism from the following three aspects: production of messages, rebalance of a consumer group and consumption of messages.

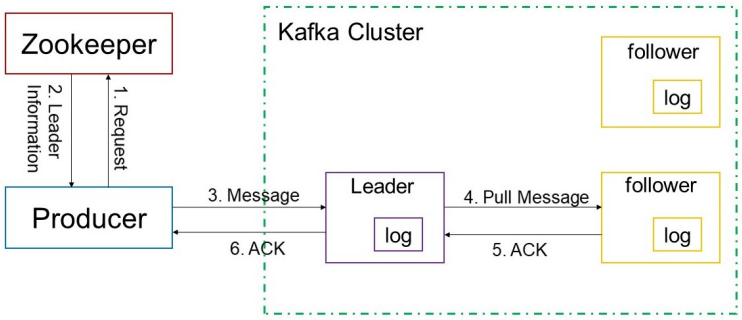


Fig. 2. The Production of Messages

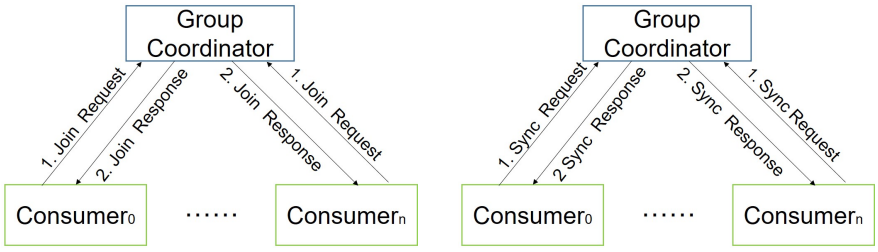
As shown in Fig.2, before publishing this message, a producer firstly needs to find the leader of a partition in this topic from ZooKeeper[18] and then sends the message to that leader. When a leader receives the message, it needs to write the message to a local log. Afterwards, followers of this partition need to pull this message from the leader and send 'ACK' to the leader after writing it to the local log. Finally, after receiving 'ACK' from all followers of this partition, the leader needs to send 'ACK' to the producer to state that the message has been delivered successfully.

In Kafka messaging mechanism, there are three types of values of 'ACK' sent by the leader in a partition to a producer: 'ACK=0', 'ACK=1' and 'ACK=all' [5]. In this paper, our model takes the third approach to ensure data security and reliability.

- 'ACK = 0' indicates that a producer does not care about the processing result of the message on the brokers. As long as it sends the message, it considers the message delivered successfully.

- 'ACK = 1' means that the message only needs to be written to the local log of the broker by the leader in this partition to return a successful commit.
- 'ACK = all' represents that before it is considered as a successful commit, the message not only needs to be stored by the leader, but also requires to be stored by all the followers of this leader .

The group coordinator of a consumer group need to manage all members in this group, and the process is called rebalance, which consists of two main steps: join and synchronization. In the process of consumers joining the group as shown in Fig.3 (a), all consumers in this group send 'JoinGroup Request' to the group coordinator and request to join. After receving all requests, the group coordinator selects one from members to play the role of leader and sends a reply message to each member of this group along with the member information of the cousumer group and the subscribed topic information to the leader. The leader needs to complete the partitioning strategy, which means that each consumer in this group should consume data from the corresponding partition of a topic. Fig.3 (b) shows the process of group synchronization. When the leader of consumers in this group completes the partitioning strategy, it encapsulates the strategy in the message named 'SyncGroup Request' to the group coordinator. Other consumers also send messages called 'SyncGroup Request'. After receiving all requests of synchronization, the group coordinator sends the partition strategy to all members of this group.



(a) Process of JoinGroup (b) Process of SyncGroup

Fig. 3. The Rebalance of a Consumer Group

Finally, we introduce the process by which consumers pull messages from the subscribed topic. In Kafka messaging system, consumers use the 'pull' mode to consume messages in sequence from the leader of the partition. Because the data transmission rate in the 'push' mode is determined by a server, but it is easy to cause problems like network congestion due to the lack of time for consumers to process the data.

It is worth noting that the message in the same partition of a topic can only be consumed by one consumer within the consumer group, but consumers in other consumer groups can still pull this messages. Moreover, when the number of partitions is greater than the number of consumers in a group, some of the consumers can pull messages from multiple partitions. In addition, we know that a topic can be divided into multiple partitions, and 'ordered' means that messages within each partition are sent to consumers in order, but there is no guarantee that messages within a topic are ordered.

2.2. CSP

Communicating Sequential Processes (CSP) [3,9] is the algebraic theory proposed by C.A.R. Hoare. The language is mainly designed to describe and analyze the behavior of concurrent systems and processes, which has been successfully applied in modeling and verifying various concurrent systems and protocols [7,11,27]. We give the following syntax of the CSP language used to describe the process in this paper, where P and Q are processes, a denotes the event and c represents the name of channel.

$$P, Q \ = \ Skip \mid Stop \mid a \rightarrow P \mid c?x \rightarrow P \mid c!x \rightarrow P \mid P \,\Box\, Q \mid$$
$$P \parallel Q \mid P \mid\mid\mid Q \mid P \lhd b \rhd Q \mid P; Q \mid P[\mid X \mid]Q$$

- $Skip$ represents that the process which does nothing but terminates successfully.
- $Stop$ denotes that the process does nothing and it is in the state of deadlock.
- $a \rightarrow P$ describes an object which first performs the event a and then behaves like P.
- $c?x \rightarrow P$ receives a message through channel c and stores the value in the variable x and then the behavior is like process P.
- $c!x \rightarrow P$ sends message x through channel c and then behaves like process P.
- $P \,\Box\, Q$ stands for the choice between process P and process Q, and this selection is decided by the environment.
- $P \parallel Q$ denotes that processes P and Q execute concurrently and are synchronized with the same communication events.
- $P \mid\mid\mid Q$ describes that processes P and Q run concurrently without barrier synchronization.
- $P \lhd b \rhd Q$ indicates if the Boolean condition b is true, the process behaves like P, otherwise like Q.
- $P; Q$ describes that processes P and Q execute in sequence.
- $P[\mid X \mid]Q$ denotes that the parallel composition of P and Q performs the concurrent events on the set X of channels.

2.3. PAT

Process Analysis Toolkit (PAT) [13,17], a toolset based on the process algebra CSP, is designed for applying model checking techniques for analysis of various systems and protocols. It supports to check for more properties [4,21], including deadlock freedom, reachability, complete LTL model checking, etc. Here we give some syntax of PAT used in this paper as follows.

- # $define\ V\ 0$
  It defines a global constant V with the initial value 0, and a global constant must be assigned an initial value in PAT.
- $var\ x\ =\ 1$
  It means that a variable x is defined with an initial value of 1. If the variable is not assigned an initial value, it defaults to 0.
- $channel\ c\ 0$
  This statement declares that c is the channel name and 0 is the buffer size. Notice that channel buffer size must be greater than or equal to 0. When the buffer size of a channel is equal to 0, it sends and receives messages synchronously.

- # $assert\ P\ deadlockfree$;
  This statement defines an assertion and it checks whether process P will enter a deadlock state or not.
- # $define\ goal\ x = false$;
  # $assert\ P\ reaches\ goal$;
  This first statement defines an assertion and the second statement checks whether process P will reach a state, where the property goal is satisfied or not.
- # $define\ goal\ x = false$;
  # $assert\ P \mid =\ goal$;
  This statement declares an assertion that checks whether process P always satisfies a state, where the property goal is satisfied or not.
- $|||i : \{0..N\}\ @P(i)$;
  This statement means that multiple processes run interspersed, specifically expressed as $P(0)$, $P(1)$, $P(2)$ ... $P(N)$.

3. Modeling

In this section, we use process algebra CSP to model the Kafka messaging mechanism which descirbes the process of communication between components in Kafka messaging system shown in Fig.1.

3.1. Sets, Messages and Channels

In order to model the process of message transmission and the behavior of components, such as producers and consumers, etc. In Kafka, we give the definitions of sets, messages and channels used in our model.

Table 1. The relationship between involved constants and pre-defined sets

Set	Constants
Module	Z(zookeeper), P(producer), PA(partition), F(follower), GC(groupcoordinator), C(consumer)
ID	TID(topic id), PID(producer id), LPAID(leader-partition id), FPAID(follower-partition id), CLeadID(leader-consumer id), CID(consumer id), GCID(GroupCoordinator ID)
Data	Data
Req	ReqData(request for data), reqTID(Request for the topic's id), reqLPAID(Request for the leader-partition's id), Join(request to join a group), Sync(request for group synchronization)
Ack	true/1(positive feedback), flase/0(negative feedback)

First, we give the definitions of some sets that are used in the model. **Module** set is composed of all modules in Kafka messaging system, including zookeeper, producers, consumers, groupcoordinators, leader-partitions and follower-partitions. **ID** set consists of unique identifier for each of the above module. **Req** set defines request information.

Table 2. The relationship between involved variables and pre-defined sets

Set	Constants
Module	z(zookeeper), p(producer), pa(partition), f(follower), gc(groupcoordinator), c(consumer)
ID	tid(topic id), lpaid(leader-partition id), fpaid(follower-partition id), cleadid(leader-consumer id), cid(consumer id), gcid(GroupCoordinator ID)
Data	data
Req	reqdata(request for data), reqtid(Request for the topic's id), reqlpaid(Request for the leader-partition's id), join(request to join a group), sync(request for group synchronization)
Ack	p_ack, c_ack, f_ack, sync_ack (positive feedback/negative feedback)

Data set includes the data transmitted between modules and **Ack** set contains feedback information.

In addition, we also give some constants based on the defined sets in Table I and some important variables we use in Table II respectively.

Based on the above sets, we describe the definition of the messages transmitted among components. In this paper, messages during communication are defined into the following three types:

$$\begin{aligned} MSG_{req} &= \{msg_{req}.A.B.content \mid A \in Module, B \in Module,\ content \in Req\} \\ MSG_{rep} &= \{msg_{rep}.A.B.content \mid A \in Module, B \in Module,\ content \in Ack\} \\ MSG_{data} &= \{msg_{data}.A.B.content \mid A \in Module, \\ &\qquad\qquad B \in Module,\ content \in \{Data,\ ID\}\} \end{aligned}$$

where, MSG_{req} is composed of the request messages transmitted between compents, MSG_{rep} represents the response messages and MSG_{data} means the data messages. A and B is sender and the receiver respectively, and $content$ represent content contained in each messagee.

Then, we define that MSG consists of the above three types of messages.

$$MSG = MSG_{req} \cup MSG_{rep} \cup MSG_{data}$$

Next, we define the channels used to simulate the communication among various modules. These channels of all modules use **COM_PATH** to represent in this paper:

- *ComZP*: the channels between zookeeper and producers. In a system, zookeeper may interact with multiple producers, and corresponding channels will also be generated. We use subscript p to distinguish each channel expressed as $ComZP_p$.
- *ComPL*: the channels between producers and leaders of the partitions. A produder can publish data on different topics to different leaders of topics by corresponding channels. We use the subscript i to distinguish each channel, which is described as $ComPL_i$.
- *ComLF*: the channels between followers and leaders of the partitions. A partition usually has one leader and multiple followers, so we use the subscript i and $ComLF_j$ to describe the multiple channels between leader and followers in a partition.

- *ComGC*: the channels between group coordinators and consumers. A consumer sends a request to the group coordinator, and the group coordinator publishes synchronization messages over this channel. Since a group coordinator manages multiple consumers within a consumper group, and there may be multiple consumper groups, we use the subscript m to distinguish them and denote them as $ComGC_m$.
- *ComLC*: the channels between consumers and leaders of the partitions. Each consumer can pull the data from mutiple leaders of partitions in different topics, and we use $ComLC_n$ to distinguish each channel.
- *ComFC*: the channels between consumers and followers of the partitions. When a broker on which the leader of a partition goes down, the consumer pulls data over the channel when one of followers becomes the leader of this partition. We use subscript l and $ComFC_j$ to describe the multiple channels.

3.2. Overall Modeling

According to the above description, the model of Kafka messaging mechanism includes six subprocesses, including $ZooKeeper$, $Producer$, $Consumer$, $GroupCoordinator$, $LPartition$ and $FPartition$. In order to facilitate the overall modeling and simulate the communication process of the model entity, we abstract the data transmission in Kafka. The overall model is shown in Fig.4.

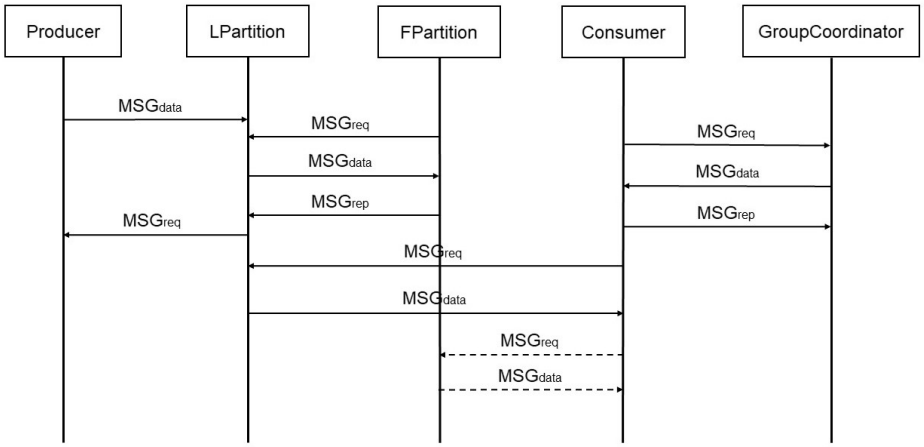


Fig. 4. The Communication Flow of Kafka Messaging Mechanism

Then, we formalize the whole model $System()$ as below:

$$
\begin{aligned}
System() =_{df}\ & |||_{\,pid\in PID,\ lpaid\in LPAID,\ fpaid\in FPAID,\ gcid\in GCID,\ cid\in CID} \\
& (\ ZooKeeper\ [|COM\_PATH|]\ Producer_{pid} \\
& \ [|COM\_PATH|]\ LPartition_{lpaid}\ [|COM\_PATH|]\ FPartitionr_{fpaid} \\
& \ [|COM\_PATH|]\ GroupCoordinator_{gcid}\ [|COM\_PATH|]\ Consumer_{cid}\ )
\end{aligned}
$$

where, the $System$ process is composed of the following processes: $ZooKeeper$, $Producer$, $LPartition$, $FPartition$, $GroupCoordinator$ and $Consumer$ concurrently using a set of channels COM_PATH. In addition, we define identifiers and a range of values for identifiers to distinguish each process. pid represents a producer's number and PID indicates the range of pid. Other characters, such as $lpaid$ and $fpaid$, have similar meanings. $[|COM\_PATH|]$ is the communication channel.

3.3. ZooKeeper

In Kafka, messages from the same Topic are divided into partitions and distributed over multiple brokers, and zookeeper needs to maintain a relationship between the partitions and the brokers. After receiving the request message from $Producer_{pid}$, $ZooKeeper$ process sends the information of these partitions to $Producer_{pid}$ by the channel $ComZP_p$. Each channel has its own id to prevent multiple processes of the same type from competing for resources on the same channel.

$$\begin{aligned} ZooKeeper() =_{df}\ & ComZP_p?msg_{req}.pid.reqtopid.reqlpaid \rightarrow \\ & ComZP_p!msg_{data}.PID.TID.LPAID \rightarrow \quad ZooKeeper() \end{aligned}$$

3.4. Producer

Producers, responsible for publishing data to partitions in a specified topic, are the important parts of Kafka messaging system. $Producer_{pid}$ process needs to find the leader of each partition from $ZooKeeper$ on the channel $ComZP_p$. At the same time, $Producer_{pid}$ provides a core parameter 'ack' to define the conditions for the message to be 'submitted'. In our model, it requires that the message has been stored not only by the leader-partition, but also by all follower-partitions of this leader.

$$\begin{aligned} Producer_{pid}() =_{df}\ & ComZP_p!msg_{req}.PID.reqTID.reqLPAID \rightarrow \\ & ComZP_p?msg_{data}.pid.tid.lpaid \rightarrow \\ & \begin{pmatrix} ComPL_i!msg_{data}.PID.LPAID.Data \\ \Box\ ComPL_i?msg_{rep}.p\_ack \end{pmatrix};\ Producer_{pid}() \end{aligned}$$

where, p_ack is a variable to describe a response message recevied from $LPartition_{lpaid}$. When its content is $P\_ack[lpaid] = 1$, it means that the data was successfully stored to the broker by all the replicas in this partition numbered $lpaid$.

3.5. LPartition

Each topic are divided into multiple partitions, but each partition has only one leader. As an important component of data storage, it needs to communicate with producers, consumers and followers of this partition. There are three types of channels that need to be used in this process: $ComPL_i$, $ComLC_n$, and $ComLF_j$.

First, it needs to accept the message from $Producer_{pid}$ process and then stores the message. Second, it needs to send the message which $Consumer_{cid}$ process needs. Finally, it also needs to send the message to $FPartition_{fpaid}$ of this partition to complete the copy of the message.

$$
\begin{aligned}
&LPartition_{lpaid}() =_{df} \\
&\left(\begin{array}{l}
\left(\begin{array}{l}
ComPL_i?msg_{data}.pid.lpaid.data \rightarrow \\
ComLF_j?msg_{req}.fpaid.tid.lpaid \rightarrow \\
ComLF_j!msg_{data}.Data \rightarrow \\
ComLF_j?msg_{rep}.fpaid.f\_ack \rightarrow \\
GetStateF(tid, lpaid, fpaid); \\
\left(\begin{array}{l}
(ComPL_i!msg_{rep}.P\_ack[lpaid]\ \{P\_ack[lpaid] = 1\}) \\
\lhd F\_ack == true\ \&\&\ lpaid \in LPAID \rhd\ SKIP
\end{array}\right)
\end{array}\right) \\
\Box \left(\begin{array}{l}
ComLC_n?msg_{req}.reqdata\ \{DataS[lpaid][cid][seq] = 1; \\
\qquad seq = Seq[lpaid];\ Seq[lpaid]++\} \rightarrow \\
ComLC_n!msg_{data}.Data
\end{array}\right)
\end{array}\right); \\
&LPartition_{lpaid}()
\end{aligned}
$$

In the above formula, $GetStateF(tid, lpaid, fpaid)$ is used to get the status of all followers to check whether they have all stored data; $P\_ack[lpaid] = 1$ indicates that the data is stored not only by the leader, but also by all followers of this partition numbered $lpaid$; $DataS[lpaid][cid][seq]$ records the state of the data with a sequence number of seq; and $Seq[lpaid]$ is the order of the data in $LPartition_{lpaid}$.

3.6. FPartition

The followers of each partition plays an important role in data security and data reliability which supports a copy mechanism. In detail, when process $LPartition_{lpaid}$ receives a message from $Producer_{pid}$, all processes $FPartition$ of this partition need to pull the message and store it. This is to ensure that when the broker of on which $LPartition_{lpaid}$ is located fails, $FPartition_{fpaid}$ of this partition can communicate with $Consumer_{cid}$ instead of it.

$$
\begin{aligned}
&FPartition_{fpaid}() =_{df} \\
&\left(\begin{array}{l}
\left(\begin{array}{l}
ComLF_j!msg_{req}.FPAID.TID.LPAID \rightarrow \\
ComLF_j?msg_{data}.data\{stateF[tid][lpaid][fpaid] = 1\} \rightarrow \\
ComLF_j!msg_{rep}.FPAID.F\_ack
\end{array}\right) \\
\Box \left(\begin{array}{l}
ComFC_l?msg_{req}.reqdata \rightarrow \\
ComFC_l!msg_{data}.Data
\end{array}\right)
\end{array}\right); \\
&FPartition_{fpaid}()
\end{aligned}
$$

where, the array $stateF[tid][lpaid][fpaid]$ indicates the state of $FPartition_{fpaid}$ of the $Lpartition_{lpaid}$ in $topic_{tip}$, i.e., whether data is received. After receiving the data successfully, the value of $stateF[tid][lpaid][fpaid]$ will change to 1.

3.7. GroupCoordinator

The group coordinator is responsible for managing all members of this group. After all consumers in this group making requests, $GroupCoordinator_{gcid}$ process selects a con-

sumer to take a leadership role and sends it group membership information and subscription information. In addition, process $GroupCoordinator_{gcid}$ notifies each $consumer_{cid}$ in the group of the partitioning strategy developed by the leader of consumers.

$$
\begin{aligned}
GroupCoordinator_{gcid}() =_{df}\ & ComGC_m?msg_{req}.join \rightarrow GetStateC(cid); \\
& \big(ConsumerL[cid] = 1 \lhd C\_ack == true \rhd SKIP\big); \\
& \left(\begin{array}{l}
\left(\begin{array}{l} ComGC_m!msg_{data}.Join.CLeadID.CID.TID.LPAID \rightarrow \\ ComGC_m?msg_{req}.sync.cid.tid.lpaid \rightarrow \\ ComGC_m!msg_{rep}.Sync\_Ack \end{array}\right) \\
\quad \lhd ConsumerL[cid] == 1 \rhd \\
\left(\begin{array}{l} ComGC_m!msg_{data}.Join.CLeadID.CID \rightarrow \\ ComGC_m?msg_{req}.sync \rightarrow ComGC_i!msg_{rep}.Sync\_Ack \end{array}\right)
\end{array}\right) \\
& ; GroupCoordinate_{gcid}()
\end{aligned}
$$

where, $ConsumerL[cid] = 1$ indicates that the consumer whose number is cid takes a leadership role in this consumer group and is responsible for the assignment of consumer and partition.

3.8. Consumer

Consumers is the core part in data consumption. First, it needs to join a consumer group by sending a request to $GroupCoordinator_{gcid}$. After joining and synchronizing the consumer group, it can pull messages from the assigned $LPartition$ according to the partitioning strategy. In addition, the $Consumer_{cid}$ can also pull messages from the follower-partitions of $LPartition$ in order to ensure that after the leader-partition crashes, consumers will still have access to the information they need.

$$
\begin{aligned}
& Consumer_{cid}() =_{df} ComGC_i!msg_{req}.Join \rightarrow \\
& \left(\begin{array}{l}
\left(\begin{array}{l} ComGC_m?msg_{data}.join.cleadid.cid.tid.lpaid\{consumer[lpaid][cid] = 1\} \\ \rightarrow ComGC_m!msg_{req}.Sync.CID.TID.LPAID \\ \rightarrow ComGC_m?msg_{rep}.sync\_ack\{consumerS[cid] = 1\} \end{array}\right) \\
\Box \left(\begin{array}{l} ComGC_m?msg_{data}.join.cleadid.cid \rightarrow ComGC_m!msg_{req}.Sync \\ \rightarrow ComGC_m?msg_{rep}.sync\_ack\{consumerS[cid] = 1\} \end{array}\right)
\end{array}\right) \\
& ; GetSync(cid); \\
& \left(\begin{array}{l}
\left(\begin{array}{l}
\left(\begin{array}{l} ComLC_n!msg_{req}.ReqData \rightarrow \\ ComLC_n?msg_{data}.data\{Data[lpaid][cid] = 1\} \end{array}\right) \\
\Box \left(\begin{array}{l} ComFC_l!msg_{req}.ReqData \rightarrow \\ ComFC_l?msg_{data}.data\{DataF[lpaid][fpaid][cid] = 1\} \end{array}\right)
\end{array}\right) \\
\lhd S\_Ack == true \ \&\& \ consumer[lpaid][cid] == 1 \rhd \\
SKIP
\end{array}\right) \\
& ; Consumer_{cid}()
\end{aligned}
$$

In the above formula, $consumer[lpaid][cid] = 1$ represents that $consumer_{cid}$ establishes a connection to $LPartition_{lpaid}$, that is, this consumer can pull messages from

the leader of partition numbered $lpaid$. $consumerS[cid]$ indicates whether $consumer_{cid}$ completes the group synchronization. $GetSync(cid)$ is a function to get the state of process $consumer_{cid}$ synchronization. $Data[lpaid][cid]$ indicates the transmission of data between $Consumer_{cid}$ and $LPartition_{lpaid}$, where $Data[lpaid][cid] = 1$ indicates success. In addition, we use $DataF[lpaid][fpaid][cid]$ to define whether $consumer_{cid}$ can pull data from $FPartition_{fpaid}$ of $LPartition_{lpaid}$.

4. Architecture Verification

In order to evaluate the correctness and reliability of Kafka in the normal communication process, we use the model checking tool PAT to verify the properties of the constructed formal model, including *Deadlock Freedom, Acknowledgement Mechanism, Parallelism, Sequentiality* and *Fault Tolerance*. Here, we give the detailed verification procedure:

4.1. Deadlock Freedom

We need to ensure that each process in the system we build can communicate and interact with each other smoothly, and that the whole system does not stop due to one process get into a deadlock state. PAT provides a primitive assertion to describe this situation:

$$\#assert\ System()\ deadlockfree;$$

The *Deadlock Freedom* property is used to verify whether our system is in the deadlock state.

4.2. Acknowledgement Mechanism

In order to avoid losing data, there is an ack mechanism designed to describe a scenario that followers copy data from the corresponding leader-partition in Kafka. Data reliability is important for data storage, thus we give the definition of the property and the assertion:

$$\#define\ Acknowledgement\_Mechanism\ \ (\ P\_ack[1] == 1\ \&\&\ P\_ack[2] == 1\ );$$
$$\#assert\ System()\ reaches\ \ Acknowledgement\_Mechanism;$$

If all the final values of the variable $P\_ack[lpaid]$ are changed from 0 to 1, we will say that the property *Acknowledgement Mechanism* is satisfied.

4.3. Parallelism

According to the partitioning strategy in our model, a partition of the same topic can only send data to one consumer of the same consumer group. In addition, we should ensure that when a consumer pulls data from the corresponding partitions, it will not affect the data of other consumers. Then we define the assertion as follows:

$$\begin{aligned}&\#define\ Parallelism\ (\ Data[0][0] == 1\ \&\&Data[0][1] == 0\\ &\qquad\qquad\qquad\qquad \&\&\ Data[1][0] == 0\ \&\&\ Data[1][1] == 1\ );\\ &\#assert\ System()\ reaches\ Parallelism;\end{aligned}$$

If our system satisfies the property *Parallelism*, then according to the partitioning strategy, $consumer_{cid}$ can connect to the corresponding channels of $partition_{lpaid}$ in a topic respectively.

4.4. Sequentiality

In Kafka, the data pulled by consumers and sent by corresponding partitions are all in order. In the above implementation section, we adopts $DataS[LPA][C][Seq]$ to represent whether the data is sent, where if its value is equal to 1, it means the data sent successfully, otherwise, not. The definitions and assertion are as follows:

$$
\begin{aligned}
&\#define\ channel1\_Seq \\
&\quad ((DataS[0][0][0] == 1 \&\& DataS[0][0][1] == 0) \\
&\quad || (DataS[0][0][0] == 1 \&\& DataS[0][0][1] == 1) \\
&\quad || (DataS[0][0][0] == 0 \&\& DataS[0][0][1] == 0)); \\
&\#define\ channel2\_Seq \\
&\quad ((DataS[1][1][0] == 1 \&\& DataS[1][1][1] == 0) \\
&\quad || (DataS[1][1][0] == 1 \&\& DataS[1][1][1] == 1) \\
&\quad || (DataS[1][1][0] == 0 \&\& DataS[1][1][1] == 0)); \\
&\#define\ Sequentiality\ (channel1\_Seq\ \&\&\ channel2\_Seq); \\
&\#assert\ System() \mid =\ Sequentiality;
\end{aligned}
$$

Since we randomly set the number of data to 2, there are three cases where the system satisfies this property. The first shows none of the messages are sent, the second indicates that the previous message is sent and the subsequent message is not sent, and the third expresses that all the messages are sent successfully.

4.5. Fault Tolerance

The *Fault Tolerance* property describes that a consumer still gets the needed message when a partition taking a leadership role breaks down. The replica mechanism enables Kafka messaging system to own this property, and the assertion is defined as follow:

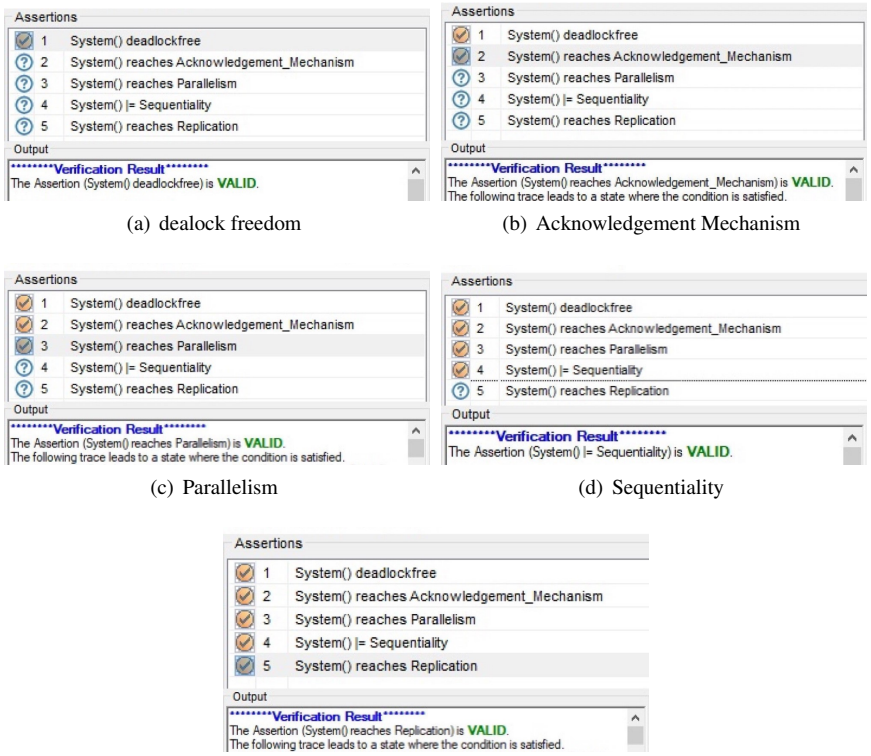
$$
\begin{aligned}
&\#define\ Replication \\
&\quad (DataF[0][0][0] == 1\ \&\&\ DataF[0][1][0] == 1 \\
&\quad \&\&\ DataF[1][0][1] == 1\ \&\&\ DataF[1][1][1] == 1); \\
&\#assert\ System()\ reaches\ Replication;
\end{aligned}
$$

Based on the partitioning strategy assumed in this paper, if processes $consumer_0$ and $consumer_1$ can respectively pull data from the followers of $partition_0$ and $partition_1$, it means the property *Fault Tolerance* is satisfied.

4.6. Verification and Results

Based on the above definitions and assertions, we implement the code in PAT and it searches the state space of the system until it finds a counter example or runs out of state space. At the end, we get the results of verification shown in Fig.5.

From Fig.5, we can see that the five properties are all valid, which means the pattern of the distributed messaging system can guarantee the correctness and reliability of communications.



(a) dealock freedom (b) Acknowledgement Mechanism

(c) Parallelism (d) Sequentiality

(e) Fault Tolerance

Fig. 5. Verification Results in $System()$

- The propertry *Deadlock Freedom* means that our model does not run into a deadlock state.
- The propertry *Acknowledgement Mechanism* represents that each replica stores the message published by the producer to ensure the reliability of the message delivery.
- The propertry *Parallelism* ensures that there is no interference between consumers within the same consumer group.
- The propertry *Sequentiality* indicates that the data consumption of each partition in Kafka is orderly.
- The propertry *Fault Tolerance* describes that Kafka messaging system is robust and does not crash even if the leader of a partition fails.

5. Security Verification

In this section, we introduce the intruder model to judge whether the messaging mechanism can guarantee its reliability and security. Meanwhile, we introduce the Kerberos protocol to improve the security of data transmission and further analyze Kafka messaging mechanism by comparing the verification results.

In an insecure network environment, some intruders may attack the process of data transmission, resulting in data leakage and other problems, which are described:

- **Camouflage**: An intruder can either send bogus messages to a broker by pretending to be a producer, or send request for consuming data to a broker pretend to be a consumer. The broker is unable to determine whether the entity sending messages has a legitimate identity, and will store the data recevied from the intruder disguised as the producer as the normal data, or will transmit the data stored on the broker to the intruder disguised as the consumer, resulting in data inauthenticity or data leakage and other problems.
- **Interception**: When a producer transmits a message containing real data to the broker, an intruder can intercept the message and discard it or tamper with it so that the broker does not receive the real message. In other case, when the broker sends data to a consumer, an intruder can also intercept the message so that the real consumer cannot receive the message and is in a waiting state, and the real data is stolen by the intruder, resulting in data leakage.

5.1. Intruder

We also deem an intruder as a process that can pretend to be a producer or a consumer to intercept messages on channels *ComPL* and *ComCL*, as well as to use fake channels to send bogus or tampered messages to the broker. Here, we introduce these channels that an intruder might use:

$$INTER\_PATH =_{df} FakePL \cup InterceptPL \cup FakeCL \cup InterceptCL \cup InterceptCF$$

Then, we define the set $Fact$, which represents the fact that an intruder might acquire:

$$Fact =_{df} Producer \cup Consumer \cup MSG$$

Next, we define the rule to express how the intruder can deduce new facts from what it has known, shown as follows:

$$F \mapsto f \wedge F \subseteq F' \Rightarrow F' \mapsto f$$

where, set F denotes the facts the intruder has known, and f is the fact deduced from set F. $F \mapsto f$ represents that fact f can be deduced from the set F.

Also, we define the funtion $Info$, which indicates how an intruder obtains a new fact from an already obtained message:

$$Info(msg.A.B.content) =_{df} \{ A.B.content \}$$

In addition, we declare a channel $Deduce$ used for deducing new facts:

$$Channel\ Deduce :\ Fact.P(Fact)$$

Based on the above description, an intruder can eavesdrop and intercept message transmitted between processes on the normal channels to obtain new facts, and can also

interfere with the communication by sending false messages. We first present a model of an intruder masquerading as a producer process:

$$
\begin{aligned}
&FakePro(F) =_{df} \ \Box_{m\in MSG} InterceptPL!m \rightarrow FakePro(F \cup Info(m)) \\
&\quad \Box\Box_{m\in MSG\cap Info(m)\subset F} FakePL!m \rightarrow FakePro(F) \\
&\quad \Box\Box_{f\in Fact, f\notin F, F\mapsto f} Init\{datac\_leakage = flase\} \rightarrow Deduce.f.F \rightarrow \\
&\left(\begin{array}{l}(DataP\_Leaking\_Success\{datap\_leakage = true\} \rightarrow FakePro(F \cup \{f\})) \\ \lhd\ f == Data \ \rhd \\ (DataP\_Leaking\_Success\{datap\_leakage = false\} \rightarrow FakePro(F \cup \{f\}))\end{array}\right)
\end{aligned}
$$

Similarly, we also present an intruder model masquerading as a consumer:

$$
\begin{aligned}
&FakeCon(F) =_{df} \ \Box_{m\in MSG} InterceptCL!m \rightarrow FakeCon(F \cup Info(m)) \\
&\quad \Box\Box_{m\in MSG\cap Info(m)\subset F} FakeCL!m \rightarrow FakeCon(F) \\
&\quad \Box\Box_{f\in Fact, f\notin F, F\mapsto f} Init\{datac\_leakage = flase\} \rightarrow Deduce.f.F \rightarrow \\
&\left(\begin{array}{l}(DataC\_Leaking\_Success\{datac\_leakage = true\} \rightarrow FakeCon(F \cup \{f\})) \\ \lhd\ f == Data \ \rhd \\ (DataC\_Leaking\_Success\{datac\_leakage = false\} \rightarrow FakeCon(F \cup \{f\}))\end{array}\right)
\end{aligned}
$$

5.2. Updated Model

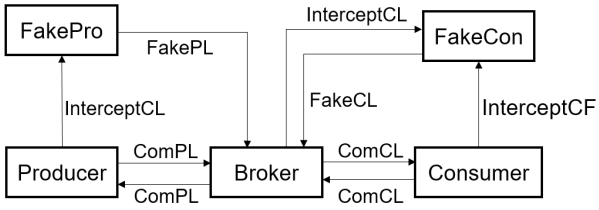


Fig. 6. Channels of Kafka Messaging Mechanism with Intruders

After modeling the intruder, we consider adding intruders to the existing system model as shown in Fig.6. In this system model, we only extract one producer process, one broker process and one consumer process. Therefore, we need to add the intruder process and the channels used on the basis of the original process interaction, so that the intruder can complete the communication interaction with the normal process.

$$
\begin{aligned}
&System\_I =_{df} \ System\_FakingP \ || \ System\_FakingC \\
&System\_FakingP =_{df} \ (\ Producer^{'} \ [|COM\_PATH|] \ Broker^{'} \\
&\qquad [|COM\_PATH|] \ Consumer^{'} \ [|INTR\_PATH|] \ FakePro\ ) \\
&System\_FakingC =_{df} \ (\ Producer^{'} \ [|COM\_PATH|] \ Broker^{'} \\
&\qquad [|COM\_PATH|] \ Consumer^{'} \ [|INTR\_PATH|] \ FakeCon\ )
\end{aligned}
$$

Updated Producer Next, we need to update the $Producer$ process so that an intruder can send fake data to a broker, and intercept data sent by a producer to a broker. Therefore, we need to add channels $FakePL$ and $InterceptPL$ to replace the original normal communication channel. We use the rename operation in CSP to update the communication channels, where $\{|c|\}$ describes the set of all events that occur on channel c:

$$\begin{aligned} Producer'() =_{df}\ & Producer()[[\\ & ComPL?\{|ComPL|\} \leftarrow ComPL?\{|ComPL|\}, \\ & ComPL!\{|ComPL|\} \leftarrow ComPL!\{|ComPL|\}, \\ & ComPL!\{|ComPL|\} \leftarrow InterceptPL!\{|ComPL|\}]] \end{aligned}$$

Updated Broker Then, we need to update the $Broker$ process so that an intruder can transmit messages to a broker by disguising a producer, or it can also transmit the message for requesting data to a broker by disguising a consumer and steal the data transmitted by a broker. We added the renamed channel to update the communication channel to complete the communication behavior of the intruder.

$$\begin{aligned} Broker'() =_{df}\ & Broker()[[\\ & ComPL?\{|ComPL|\} \leftarrow ComPL?\{|ComPL|\}, \\ & ComPL?\{|ComPL|\} \leftarrow FakePL?\{|ComPL|\}, \\ & ComPL!\{|ComPL|\} \leftarrow ComPL!\{|ComPL|\}, \\ & ComCL?\{|ComCL|\} \leftarrow ComCL?\{|ComCL|\}, \\ & ComCL?\{|ComCL|\} \leftarrow InterceptCL?\{|ComCL|\}, \\ & ComCL!\{|ComCL|\} \leftarrow ComCL!\{|ComCL|\}, \\ & ComCL!\{|ComCL|\} \leftarrow FakeCL!\{|ComCL|\}]] \end{aligned}$$

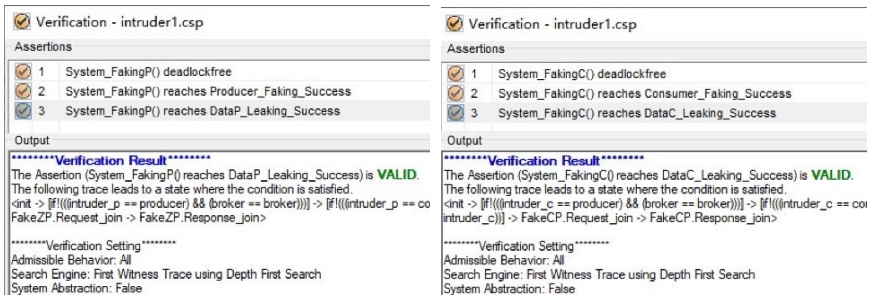
Updated Consumer Similarly, an intruder can disguise as a real consumer to intercept and tamper with a request message from a consumer over the normal channel to a broker. The updated $Consumer$ process describes as follow:

$$\begin{aligned} Consumer'() =_{df}\ & Consumer()[[\\ & ComCL?\{|ComCL|\} \leftarrow ComCL?\{|ComCL|\}, \\ & ComCL!\{|ComCL|\} \leftarrow ComCL!\{|ComCL|\}, \\ & ComCL!\{|ComCL|\} \leftarrow InterceptCF!\{|ComCL|\}]] \end{aligned}$$

5.3. Verification Results of Model with Intruder

We implement the updated model and the intruder model in PAT and get the results of verification shown in Fig.7.

From Fig.7 (a), we can see the three properties are all valid, which means that an intruder can disguise a producer and successfully intercept the data message transmitted by



(a) Camouflage Producer (b) Camouflage Consumer

Fig. 7. Verification Results in $System\_I()$

a real producer in the normal transmission process, thus causing the data leakage problem. Similarly, we see the results shown in Fig.7 (b) that three properties are all valid, which means that an intruder can disguise a consumer to intercept the data transmitted by a broker to a consumer, thus leading to the problem of data leakage.

5.4. Kerberos

After the 0.9.0.0 version of Kafka messaging system, security mechanism was introduced. This paper analyzes the security of Kafka messaging mechanism by adding the modeling of Kerberos authentication protocol and comparing the validation results of the messaging mechanism model with no security mechanism.

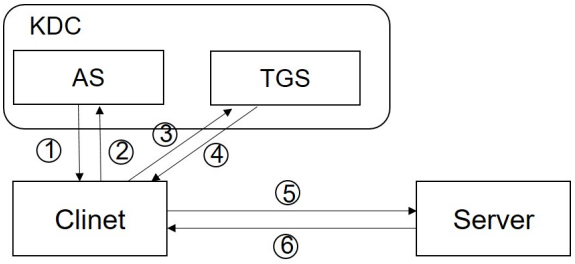


Fig. 8. Channels of Kafka Messaging Mechanism with Intruders

Kerberos [2,22] is a protocol based on key encryption technology by MIT, which provides authentication of identity for applications of client and server. There are three main modules: client, server and key distribution center(KDC). KDC is composed of authentication server (AS) and ticket granting server (TGS). Fig.8 describes the authentication process of Kerberos protocol in detail.

– In the first step, the client needs to prove his identity to the AS in the KDC.

- In the second step, AS determines whether the user's identity is valid. If it is valid, AS will generate the session key between the client and TGT[3]. Then it encrypts the session key and TGT with the client's public key and transmits it to the client.
- Third, after receving the encrypted message transmitted by AS, the client decrypts the ciphertext message with its private key to obtain TGT and session key. Then, the client transmits TGT to TGS together with Auth[4].
- Step 4, Similarly, TGS uses its private key to decrypt TGT to obtain the information of the client and session key between the client and TGS, and uses the session key to decrypt Auth to obtain the identity information provided by the client. At this time, TGS checks whether the timestamp is expired and verifies the consistency of the client's information by comparison, as well as encrypts the session key between the client and the server. When the information is verified successfully, TGS returns the encrypted session key and ST[5].
- Step 5, the client decrypts the message with the session key between the client and TGS to obtain the session key of the client and the server, and uses this session key to encrypt Auth$'$[6]. Then the ciphertext message ST and Auth$'$ are sent to the server.
- The sixth step, the server obtains the identity information by decrypting ST with its private key and the identity information by decrypting Auth$'$ with the session between the client and the server, respectively. After the verification is passed, the server uses the session key between the client and the server to encrypt the server's identity information and timestamp, and transmits the ciphertext message to the client. Finally, the client determines whether the server identity is consistent. If consistent, the connection is successful, that is, the client can access the server.

Kerberos Model First, we need to add some definitions for other sets: the set **Key** contains the long-term key **MK** and the short-term key **SK**, the set **Time** contains the definition of discrete time.

In addition, we also defined the encryption function **E** and decryption function **D**, which are specifically expressed as follows:

$$E(k, msg);\ D(k, emsg)$$

Where, function **E** uses key **k** to encrypt the message **msg**, which is represented as **emsg**, and function **D** uses key **k** to decrypt the encrypted message **emsg**. Therefore, we can draw the following conclusion:

$$D(k, E(k, msg)) = msg$$

Next, we define some of the channels used by the processes when modeling the Kerberos protocol, as follows:

$$Time,\ ComUA,\ ComTA,\ ComSA$$

[3] TGT: the ticket provided to TGS when applying for the server ticket encrypted the public key of TGS, containing the client's identity, time stamp and the session key between the user and TGS.

[4] Auth: authentication information of the client encrypted by the session key between the client and TGS, including user name, timestamp.

[5] ST: the ticket to access the server encrypted with the server's public key, containing the client's identity, timestamp, and session key between the server and the client.

[6] Auth$'$: authentication information of the client encrypted by the session key between the client and the server

Based on the definitions of sets, functions and channels, we model the Kerberos protocol. First, we need to define a process $Clock$ to synchronize time of the whole system. $Clock$ process represents a discrete increase in time t, and when receiving a request message, it responds with a message containing the current time t.

$$Clock() =_{df} tick -> \{t = t + 1\} -> Clock() \\ \Box\, time\,?\, request -> time\,!\, t -> Clock();$$

To facilitate the modeling of the Kerberos protocol, we will model steps 1-6 in Fig.8 respectively. The Step 1 and Step 2 in Fig.8 are the interactions between the client and AS. Process $USER\_AS$ needs to send a request to process $Clock$ and takes the returned time t as the initial time. Next, process UER_AS sends a message including its name and initial time to AS and expects a reply from AS. If the authentication fails, process UER_AS will stops. If the anthentication is successful, process UER_AS will decrypt the message with its private key MK_{user}^{-1} to get TGT and session key between TGS and client SK_{TGS}.

$$USER\_AS(user) =_{df} time\,!\, Request \rightarrow time\,?\, t\,\{\, starttime = t\,\}\,; \\ ComUA\,!\, name.starttime\{name = user\} \rightarrow \\ \left(\begin{array}{l} ComUA\,?\, name\_fail \rightarrow Stop \\ \Box \left(\begin{array}{l} ComUA\,?\, x \\ \{\, TGT = D(MK_{user}^{-1}, x);\ SK_{TGS} = D(MK_{user}^{-1}, x)\} \rightarrow Skip \end{array} \right) \end{array} \right)$$

In the communication between the client and AS, process AS will judge whether the username is valid after receiving the message sent by process UER_AS. If not, a failed reponse is returned to UER_AS. If successful, AS generates SK_{TGS}, as well as uses the client's public key MK_{user} to encrypt SK_{TGS} and TGT, and sends the encrypted message to UER_AS.

$$AS() =_{df} ComUA?\, x\{name = getname(x);\ name\_faking = \neg(valid(name))\}\,; \\ \left(\begin{array}{l} ComUA\,!\, Name\_Fail \rightarrow Skip \\ \lhd\, name\_invalid == true \,\rhd \\ \left(\begin{array}{l} ComUA\,!\, E(MK_{user},\ (TGT,\ SK_{TGS})) \\ \{TGT = E(MK_{TGS}, (username, starttime, lifetime, SK_{TGS}))\} \rightarrow Skip \end{array} \right) \end{array} \right);$$

In the above formula, the function $getname()$ is used to get the client's name, and the function $valid(name)$ is used to determine whether the name is valid.

Steps 3 and 4 in Fig.8 are the interactions between the client and TGS. $USER\_TGS$ process sends the TGT and $Auth$ to process TGS, and expects to receive a reply from TGS. The first case is when the process stops because the ticket expires due to a timeout. In the second case, when the identity information in TGT does not match the identity information $Auth$, the process terminates. The last one is that the authentication succeeds, and process $USER\_TGS$ receives ST and the session key SK_{Server} between the client and the server.

$$USER\_TGS(user, server) =_{df} \; ComTA\,!\,TGT.Auth$$
$$\{Auth = E(SK_{TGS}, (username, starttime, lifetime))\} \rightarrow$$
$$\left(\begin{array}{l} ComTA\,?\,Timeout \rightarrow Stop \\ \Box \left(\begin{array}{l} ComTA\,?\,invaild \rightarrow Stop \\ \Box(ComTA\,?\,y\,\{\, SK_{Server} = D(SK_{TGS}, y)\} \rightarrow Skip\,) \end{array} \right) \end{array} \right)$$

After receiving the message from process $USER\_TGS$, process TGS first sends a request message to process $Clock$ and obtains the time t of the current system. Then, it uses the private key MK_{TGS}^{-1} to decrypt TGT to get SK_{TGS} as well as the client's identity information, initial time and life time provided by AS. Next, TGS decrypts $Auth$ transmitted by process $USER\_TGS$ with SK_{TGS}, obtains the client's information, initial time and life time given by client, and judges whether the timeout is determined by comparing the life time of the ticket with the current time. If the time runs out, TGS will directly send the message named $Timeout$ to $USER\_TGS$. If the ticket is within the valid time, TGS compares the information provided by AS and the client, and an invalid reply message is sent if it does not match. If it is consistent, TGS generates ST and Sk_{Server} encrypted with SK_{TGS}, and sends them to process $USER\_TGS$.

$$TGS() =_{df} \; dComTA\,?\,y \rightarrow time\,!\,Request \rightarrow time\,?\,t\,\{\, nowtime = t\,\} \rightarrow$$
$$TGT = get(y) \rightarrow msg_{TGT} = D(MK_{TGS}^{-1},\, TGT) \rightarrow$$
$$SK_{TGS} = get(msg_{TGT}) \rightarrow a = getname(msg_{TGT}) \rightarrow$$
$$starttime_t = get(msg_{TGT}) \rightarrow lifetime_t = get(msg_{TGT}) \rightarrow$$
$$Auth = get(y) \rightarrow msg_{Auth} = D(SK_{TGS},\, Auth) \rightarrow b = getname(msg_{Auth}) \rightarrow$$
$$starttime_a = get(msg_{Auth}) \rightarrow lifetime_a = get(msg_{Auth}) \rightarrow$$
$$\left(\begin{array}{l} ComTA\,!\,Timeout \rightarrow Stop \\ \lhd\, nowtime - starttime_t > lifetime_t \,||\, nowtime - starttime_a > lifetime_a \rhd \\ \left(\begin{array}{l} ComTA\,!\,invaild \rightarrow Stop \quad \lhd\; a \neq b \rhd \\ ComTA\,!\,E(SK_{TGS},\, Sk_{Server}).ST \\ \{ST = E(MK_{Server}, (username, starttime, lifetime, SK_{Server}))\} \rightarrow Skip \end{array} \right) \end{array} \right);$$

Finally, the interactions between the client and the server correspond to steps 5 and 6 in Fig.8. Process $USER\_Server$ transmits $Auth^{'}$ and the ticket ST to process $Server$. If process $USER\_Server$ receives an invalid message from $Server$, the process stops. Otherwise, $USER\_Server$ decrypts the message by using Sk_{Server} to get the information of the server, and determines whether it is valid. If it is valid, the authentication is successful, otherwise process $USER\_Server$ terminates.

$$USER\_Server(user, server) =_{df} \; ComSA\,!\,ST.Auth^{'}$$
$$\{Auth^{'} = E(SK_{Server}, (username, starttime, lifetime))\} \rightarrow$$
$$\left(\begin{array}{l} ComSA\,?\,invaild \rightarrow Stop \\ \Box \left(\begin{array}{l} ComSA\,?\,z\,\{\, server\_name = D(Sk_{Server},\, z)\} \rightarrow \\ Server\_faking\_success = \neg(valid(server\_name)) \rightarrow \\ (Stop \lhd\; Server\_faking\_success == true \;\rhd Skip) \end{array} \right) \end{array} \right);$$

Firstly, process $Server$ uses the private key MK^{-1}_{Server} to decrypt ST to get the client's name provided by TGS and the session key SK_{Server}. Then, $Server$ decrypts $Auth^{'}$ using SK_{Server} to obtain the client's name provided by $USER\_Server$. If this name matches the name provided by TGS, process $Server$ sends a message containing the identity information of the server encrypted with SK_{Server} to $USER\_Server$, otherwise it sends an invalid message reply.

$$
\begin{aligned}
&Server(server) =_{df}\ ComSA\,?\,z \ \rightarrow\ ST = get(z) \ \rightarrow \\
&\quad msg_{ST} = D(MK^{-1}_{Server},\ ST) \rightarrow\ \ c = getname(msg_{ST}) \rightarrow \\
&\quad SK_{Server} = D(MK^{-1}_{Server},\ ST) \rightarrow\ Auth^{'} = get(z) \ \rightarrow \\
&\quad msg_{Auth'} = D(SK_{Server},\ Auth^{'}) \rightarrow\ d = getname(msg_{ST}) \ \rightarrow \\
&\begin{pmatrix} ComSA\,!\,invaild \ \rightarrow Stop \quad \lhd\ e \neq f \rhd \\ ComSA\,!\,E(SK_{Server},\ (server\_name, starttime, lifetime)) \ \rightarrow\ Skip \end{pmatrix};
\end{aligned}
$$

5.5. Updated Model Based on Kerberos

Updated Producer based on Kerberos In Kafka messaging mechanism based on Kerberos, process $Producer$, as a client, first authenticates with AS and obtain a ticket to access TGS. Then it needs to authenticate with TGS to obtain tickets to access the server $Broker$ and authenticate with $Broker$ to send messages to $Broker$. The updated $Producer^{'}$ model is as follows:

$$
\begin{aligned}
Producer^{'} =_{df}\ & USER\_AS(producer);\ USER\_TGS(producer, broker); \\
& \begin{pmatrix} ComPL\,!\,Data \ \rightarrow\ ComPL\,?\,ack \ \rightarrow Producer^{'} \\ \lhd\ Connect\_Success == true \ \rhd \quad Stop \end{pmatrix};
\end{aligned}
$$

Updated Broker based on Kerberos AS a server, process $Broker$ needs to verify the identity of the client who wants to access it. After the verification is passed, $Broker$ also needs to provide its own identity to the client. Only after the two-way authentication is successful, the following communication with the client will continue. The updated model is as follows:

$$
\begin{aligned}
&Broker^{'} =_{df}\ \ Server(broker); \\
&\begin{pmatrix} ComPL\,?\,data \ \rightarrow\ ComPL\,!\,Ack \ \rightarrow Broker^{'} \\ \Box ComCL\,?\,request \ \rightarrow\ ComCL\,!\,Data \ \rightarrow Broker^{'} \end{pmatrix};
\end{aligned}
$$

Updated Consumer Based on Kerberos Similarly, process $Consumer$ is a client that need to be authenticated by AS and TGS to obtain a ticket to access $Broker$. It also needs to authenticate with $Broker$ to send messages requesting data. The updated $Consumer^{'}$ model is as follows:

$$
\begin{aligned}
Consumer^{'} =_{df}\ & USER\_AS(consumer);\ USER\_TGS(consumer, broker); \\
& \begin{pmatrix} ComCL\,!\,Request \ \rightarrow\ ComCL\,?\,data \ \rightarrow Consumer^{'} \\ \lhd\ Connect\_Success == true \ \rhd \quad Stop \end{pmatrix};
\end{aligned}
$$

Updated Intruder First, we update the facts the intruder has learned:

$$FACT^{'} =_{df} Fact \cup Time \cup Key \cup MSG \\ \cup \{ E(k, content) \mid k \in Key,\ content \in \{Data, Key, Time\}\}$$

Next, we add the following rules :

$$\{MK^{-1}, E(MK, content)\} \mapsto content, \\ \{SK^{-1}, E(SK, content)\} \mapsto content, \\ \{MK, content)\} \mapsto E(MK, content), \\ \{SK, content)\} \mapsto E(SK, content)$$

The first two rules describe that the intruder can use the corresponding key to decrypt the encrypted messages and get some contents. In the same way, the next two rules represent encryption. The final rule is a structural rule, explaining that the intruder can deduce fact f from a lager set $F^{'}$, if f can be deduced from set F.

Finally, we present an updated model of the intruder process disguised as a producer:

$$FakePro^{'}(F) =_{df} \\ \Box_{m \in MSG} InterceptPL!m \rightarrow FakePro^{'}(F \cup Info(m)) \\ \Box\Box_{m \in MSG \cap Info(m) \subset F} FakePL!m \rightarrow FakePro^{'}(F) \\ \Box\Box_{f \in Fact', f \notin F, F \mapsto f} Init\{datac\_leakage = flase\} \rightarrow Deduce^{'}.f.F \rightarrow \\ \begin{pmatrix} (DataP\_Leaking\_Success\{datap\_leakage = true\} \rightarrow FakePro^{'}(F \cup \{f\})) \\ \lhd\ f == Data\ \rhd \\ (DataP\_Leaking\_Success\{datap\_leakage = false\} \rightarrow FakePro^{'}(F \cup \{f\})) \end{pmatrix}$$

Similarly, we update the intruder model that disguises the consumer:

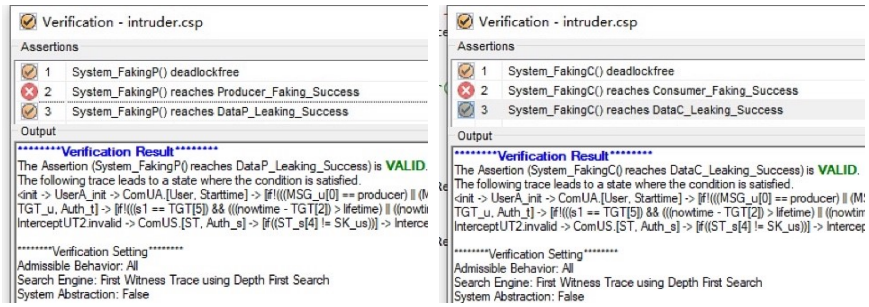
$$FakeCon(F)^{'} =_{df} \\ \Box_{m \in MSG} InterceptCL!m \rightarrow FakeCon(F \cup Info(m)) \\ \Box\Box_{m \in MSG \cap Info(m) \subset F} FakeCL!m \rightarrow FakeCon^{'}(F) \\ \Box\Box_{f \in Fact', f \notin F, F \mapsto f} Init\{datac\_leakage = flase\} \rightarrow Deduce^{'}.f.F \rightarrow \\ \begin{pmatrix} (DataC\_Leaking\_Success\{datac\_leakage = true\} \rightarrow FakeCon^{'}(F \cup \{f\})) \\ \lhd\ f == Data\ \rhd \\ (DataC\_Leaking\_Success\{datac\_leakage = false\} \rightarrow FakeCon^{'}(F \cup \{f\})) \end{pmatrix}$$

Overall Model based on Kerberos In Kafka messaging mechanism based on Kerberos protocol, the overall model is described as follows:

$$
\begin{aligned}
&System\_K =_{df} \ System\_FakingP \ || \ System\_FakingC \\
&System\_FakingP =_{df} \ (\ Producer^{'} \ [|COM\_PATH|] \ Broker^{'} \ [|COM\_PATH|] \\
&\quad Consumer^{'} \ [|COM\_PATH|] \ Kerberos \ [|INTR\_PATH|] \ FakePro^{'}(F)) \\
&System\_FakingC =_{df} \ (\ Producer^{'} \ [|COM\_PATH|] \ Broker^{'} \ [|COM\_PATH|] \\
&\quad Consumer^{'} \ [|COM\_PATH|] \ Kerberos \ [|INTR\_PATH|] \ FakeCon^{'}(F)) \\
&Kerberos =_{df} \ (\ AS \ [|COM\_PATH|] \ TGS \ [|COM\_PATH|] \ Clock)
\end{aligned}
$$

5.6. Verification Results of Model based on Kerberos

We implement the updated model based on Kerberos protocol and the updated intruder model in PAT and get the results of verification shown in Fig.9.



(a) dealock freedom (b) Acknowledgement Mechanism

Fig. 9. Verification Results in $System\_K()$

From Fig.9 (a), we can see *deadlock freedom* is vaild, which means the communication can be completed successfully in the system. $Producer\_Faking\_Success$ is invalid, which means that an intruder cannot disguise a producer and send bogus messages. The property $DataP\_Leaking\_Success$ is valid, which means that an intruder can still intercept the data message transmitted by a real producer successfully, thus causing the data leakage problem.

Similarly, in Fig.9 (b), the property *deadlock freedom* is vaild, which means each process in the system will not run into a deadlock state . The property $Consumer\_Faking\_Success$ is invalid, which means that an intruder cannot send a bogus messages for requesting data by disguising a consumer. The property $DataC\_Leaking\_Success$ is valid, which means that an intruder can still intercept the data transmitted by a broker to a consumer, thus leading to the problem of data leakage.

6. Related Work

In recent years, there have been some researches on the performance of Kafka messaging system in the field of distributed messaging system [14,15,16,29,30]. For instance, in order to set the configuration of the Kafka system correctly under certain hardware conditions to ensure its performance, Han et al. [29] analyzed the structure and workflow of Kafka and proposed a queue-based package flow model to predict the performance of Kafka cloud services. In the paper, they observed the effect of these parameters on the performance by substituting the correlation and fitting results into the fundamental constants of the model and inputting various configuration parameters.

Also, Han et al. [30] introduced a testing tool TRAK to compared the reliability of different messaging transmission semantics in Kafka under the environment of poor network performance by using two indicators namely message loss rate and repetition rate. And in the reliability evaluation of Kafka application scenarios, such as tracking website user information, monitoring server logs, online bank transfer and online booking, etc. Han et al. [28] also tested the effect of various configuration parameters on the reliability of the Kafka system in order to help users weigh the performance and reliability of the application in practical application.

In addition, Sean et al. [15] wanted to find the practical problems that arise when companies use Kafka as a single data store, and to be able to propose solutions to solve these problems. To this end, they proposed some preliminary approaches to ensure the consistency of data from multiple database tables when distributed over Kafka, and how to solve compliance problems by encrypting/decrypting data from Kafka producers and consumers. We can see that these studies mostly focused on the performance analysis of Kafka and how to improve the performance of Kafka applications, but in this paper we focus on the reliability and security of data in the interaction and messaging transmission of various components in Kafka.

At the same time, there are many successful studies on the property analysis and verification of systems and network protocols by formal methods [7,11,12,23,27]. For instance, Lowe et al. [12] analyzed and verified the communication protocol TMN using CSP and FDR, and they have detected the security loopholes of the protocol and put forward the optimization scheme from the theoretical aspect. Thampibal et al. [23] proposed an alternative of formalizing the high-level railway network by using hierarchical timed coloured Petri nets and verified the constructed model with CPN tool to ensure its correctness and security. Wang et al. [27] analyzed the security of the OpenFlow scheduled bundle mechanism and found that it suffered from some kinds of possible attacks by modeling and verifying the mechanism using CSP and PAT. In this paper, we choosed the process algebra CSP and model checking tool PAT to analyze and verify the reliability and security of Kafka messaging mechanism.

7. Conclusion and Future Work

In this paper, we adopted the process algebra CSP to model Kafka messaging mechanism, and utilized the model checker PAT to verify five properties, including *Deadlock Freedom, Acknowledgement Mechanism, Parallelism, Sequentiality* and *Fault Tolerance*. The results of verification show that all properties are valid, which means the pattern of

the distributed messaging system can guarantee the correctness and reliability of communications. In order to further analyze the security of Kafka messaging mechanism, we added the intruder model and the Kerberos model. By comparing the results of Kafka messaging mechanism with or without the secure protocol Kerberos, we can conclude that the protocol Kerberos can effectively prevent the camouflage attack of the intruder, but it can not resist attacks to intercept data, so there are still some security problems.

In the future work, we will put forward a preliminary improvement method from the theoretical aspect to solve these security problems, including digital signature and key encryption, so as to further improve the security of Kafka in the process of messaging transmission.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 62032024, 61872145), the "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (Grant No. 22510750100), and the Dean's Fund of Shanghai Key Laboratory of Trustworthy Computing (East China Normal University).

References

1. Rabbitmq., https://www.rabbitmq.com/tutorials/amqp-concepts.html
2. Adams, C.: Kerberos authentication protocol. In: van Tilborg, H.C.A., Jajodia, S. (eds.) Encyclopedia of Cryptography and Security, 2nd Ed, pp. 674–675. Springer (2011), https://doi.org/10.1007/978-1-4419-5906-5_81
3. Brookes, S.D., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. J. ACM 31(3), 560–599 (1984), https://doi.org/10.1145/828.833
4. Clarke, E.M., Henzinger, T.A., Veith, H.: Introduction to model checking. In: Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.) Handbook of Model Checking, pp. 1–26. Springer (2018), https://doi.org/10.1007/978-3-319-10575-8_1
5. Dobbelaere, P., Esmaili, K.S.: Kafka versus rabbitmq: A comparative study of two industry reference publish/subscribe implementations: Industry paper. In: Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, DEBS 2017, Barcelona, Spain, June 19-23, 2017. pp. 227–238. ACM (2017), https://doi.org/10.1145/3093742.3093908
6. Eugster, P.T., Felber, P., Guerraoui, R., Kermarrec, A.: The many faces of publish/subscribe. ACM Comput. Surv. 35(2), 114–131 (2003), https://doi.org/10.1145/857076.857078
7. Fei, Y., Zhu, H.: Modeling and verifying NDN access control using CSP. In: Sun, J., Sun, M. (eds.) Formal Methods and Software Engineering - 20th International Conference on Formal Engineering Methods, ICFEM 2018, Gold Coast, QLD, Australia, November 12-16, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11232, pp. 143–159. Springer (2018), https://doi.org/10.1007/978-3-030-02450-5_9
8. Hesse, G., Matthies, C., Uflacker, M.: How fast can we insert? an empirical performance evaluation of apache kafka. In: 26th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2020, Hong Kong, December 2-4, 2020. pp. 641–648. IEEE (2020), https://doi.org/10.1109/ICPADS51040.2020.00089
9. Hoare, C.A.R.: Communicating sequential processes. Commun. ACM 21(8), 666–677 (1978), https://doi.org/10.1145/359576.359585
10. Lee, V.Y., Liu, Y., Zhang, X., Phua, C., Sim, K., Zhu, J., Biswas, J., Dong, J.S., Mokhtari, M.: ACARP: auto correct activity recognition rules using process analysis toolkit (PAT). In: Donnelly, M.P., Paggetti, C., Nugent, C.D., Mokhtari, M. (eds.) Impact Analysis of Solutions

for Chronic Disease Prevention and Management - 10th International Conference on Smart Homes and Health Telematics, ICOST 2012, Artiminio, Italy, June 12-15, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7251, pp. 182–189. Springer (2012), https://doi.org/10.1007/978-3-642-30779-9_23
11. Liu, A., Zhu, H., Popovic, M., Xiang, S., Zhang, L.: Formal analysis and verification of the PSTM architecture using CSP. J. Syst. Softw. 165, 110559 (2020), https://doi.org/10.1016/j.jss.2020.110559
12. Lowe, G., Roscoe, A.W.: Using CSP to detect errors in the TMN protocol. IEEE Trans. Software Eng. 23(10), 659–669 (1997), https://doi.org/10.1109/32.637148
13. PAT: Process analysis toolkit., http://pat.comp.nus.edu.sg/
14. Prabhu, C., Gandhi, R.V., Jain, A.K., Lalka, V.S., Thottempudi, S.G., Rao, P.P.: A novel approach to extend KM models with object knowledge model (OKM) and kafka for big data and semantic web with greater semantics. In: Barolli, L., Hussain, F.K., Ikeda, M. (eds.) Complex, Intelligent, and Software Intensive Systems - Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2019, Sydney, NSW, Australia, 3-5 July 2019. Advances in Intelligent Systems and Computing, vol. 993, pp. 544–554. Springer (2019), https://doi.org/10.1007/978-3-030-22354-0_48
15. Rooney, S., Urbanetz, P., Giblin, C., Bauer, D., Froese, F., Garcés-Erice, L., Tomic, S.: Kafka: the database inverted, but not garbled or compromised. In: Baru, C., Huan, J., Khan, L., Hu, X., Ak, R., Tian, Y., Barga, R.S., Zaniolo, C., Lee, K., Ye, Y.F. (eds.) 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019. pp. 3874–3880. IEEE (2019), https://doi.org/10.1109/BigData47090.2019.9005583
16. Sharvari T, S.N.K.: A study on modern messaging systems- kafka, rabbitmq and NATS streaming. CoRR abs/1912.03715 (2019), http://arxiv.org/abs/1912.03715
17. Si, Y., Sun, J., Liu, Y., Dong, J.S., Pang, J., Zhang, S.J., Yang, X.: Model checking with fairness assumptions using PAT. Frontiers Comput. Sci. 8(1), 1–16 (2014), https://doi.org/10.1007/s11704-013-3091-5
18. Skeirik, S., Bobba, R.B., Meseguer, J.: Formal analysis of fault-tolerant group key management using zookeeper. In: 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2013, Delft, Netherlands, May 13-16, 2013. pp. 636–641. IEEE Computer Society (2013), https://doi.org/10.1109/CCGrid.2013.98
19. Sun, D., Zhu, H., Fei, Y., Xiao, L., Lu, G., Yin, J.: Formalization and verification of TESAC using CSP. Int. J. Softw. Eng. Knowl. Eng. 29(11&12), 1741–1760 (2019), https://doi.org/10.1142/S0218194019400199
20. Sun, J., Liu, Y., Dong, J.S.: Model checking CSP revisited: Introducing a process analysis toolkit. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation, Third International Symposium, ISoLA 2008, Porto Sani, Greece, October 13-15, 2008. Proceedings. Communications in Computer and Information Science, vol. 17, pp. 307–322. Springer (2008), https://doi.org/10.1007/978-3-540-88479-8_22
21. Sun, J., Liu, Y., Dong, J.S., Pang, J.: PAT: towards flexible verification under fairness. In: Bouajjani, A., Maler, O. (eds.) Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5643, pp. 709–714. Springer (2009), https://doi.org/10.1007/978-3-642-02658-4_59
22. Tbatou, Z., Asimi, A., Asimi, Y., Sadqi, Y., Guezzaz, A.: A new mutuel kerberos authentication protocol for distributed systems. Int. J. Netw. Secur. 19(6), 889–898 (2017), http://ijns.jalaxy.com.tw/contents/ijns-v19-n6/ijns-2017-v19-n6-p889-898.pdf
23. Thampibal, L., Vatanawood, W.: Formalizing railway network using hierarchical timed coloured petri nets. In: ICIT 2019 - The 7th International Conference on Information Tech-

nology: IoT and Smart City, Shanghai, China, December 20-23, 2019. pp. 338–343. ACM (2019), https://doi.org/10.1145/3377170.3377221
24. Treat, T.: Benchmarking nats streaming and apache kafka., https://dzone.com/articles/benchmarking-nats-streaming-and-apachekafka
25. Vucnik, M., Svigelj, A., Kandus, G., Mohorcic, M.: Secure hybrid publish-subscribe messaging architecture. In: Begusic, D., Rozic, N., Radic, J., Saric, M. (eds.) 2019 International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2019, Split, Croatia, September 19-21, 2019. pp. 1–5. IEEE (2019), https://doi.org/10.23919/SOFTCOM.2019.8903868
26. Wang, G., Koshy, J., Subramanian, S., Paramasivam, K., Zadeh, M., Narkhede, N., Rao, J., Kreps, J., Stein, J.: Building a replicated logging system with apache kafka. Proc. VLDB Endow. 8(12), 1654–1655 (2015), http://www.vldb.org/pvldb/vol8/p1654-wang.pdf
27. Wang, H., Zhu, H., Xiao, L., Fei, Y.: Formalization and verification of the openflow bundle mechanism using CSP. Int. J. Softw. Eng. Knowl. Eng. 28(11-12), 1657–1677 (2018), https://doi.org/10.1142/S0218194018400223
28. Wu, H.: Research proposal: Reliability evaluation of the apache kafka streaming system. In: Wolter, K., Schieferdecker, I., Gallina, B., Cukier, M., Natella, R., Ivaki, N.R., Laranjeiro, N. (eds.) IEEE International Symposium on Software Reliability Engineering Workshops, ISSRE Workshops 2019, Berlin, Germany, October 27-30, 2019. pp. 112–113. IEEE (2019), https://doi.org/10.1109/ISSREW.2019.00055
29. Wu, H., Shang, Z., Wolter, K.: Performance prediction for the apache kafka messaging system. In: Xiao, Z., Yang, L.T., Balaji, P., Li, T., Li, K., Zomaya, A.Y. (eds.) 21st IEEE International Conference on High Performance Computing and Communications; 17th IEEE International Conference on Smart City; 5th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2019, Zhangjiajie, China, August 10-12, 2019. pp. 154–161. IEEE (2019), https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00036
30. Wu, H., Shang, Z., Wolter, K.: TRAK: A testing tool for studying the reliability of data delivery in apache kafka. In: Wolter, K., Schieferdecker, I., Gallina, B., Cukier, M., Natella, R., Ivaki, N.R., Laranjeiro, N. (eds.) IEEE International Symposium on Software Reliability Engineering Workshops, ISSRE Workshops 2019, Berlin, Germany, October 27-30, 2019. pp. 394–397. IEEE (2019), https://doi.org/10.1109/ISSREW.2019.00101

Junya Xu obtained her master degree in formal methods from East China Normal University, Shanghai, in 2021. Her research interests include process algebra and its applications, program analysis and verification.

Jiaqi Yin is currently an assistant professor in Northwestern Polytechnical University, Xi'an, China. He earned his Ph.D. degree in software engineering from East China Normal University, Shanghai, in 2022. His research interests contain formal methods. edge computing, and process algebra.

Huibiao Zhu is currently a professor in East China Normal University, Shanghai. He earned his Ph.D. degree in formal methods from London South Bank University, London, in 2005. During these years, he has studied various semantics and their linking theories for Verilog, SystemC, web services and probability system. He was the Chinese PI of the Sino-Danish Basic Research Center IDEA4CPS.

Lili Xiao is currently a lecturer in Donghua University, Shanghai. She earned her Ph.D. degree in software engineering from East China Normal University, Shanghai, in 2022. Her research interests include process algebra and its applications, program analysis and verification, and weak memory.

Received: July 07, 2021; Accepted: May 11, 2022.

Computer Science and Information Systems 20(1):307–327 https://doi.org/10.2298/CSIS210908058P

Complete Formal Verification of the PSTM Transaction Scheduler*

Miroslav Popovic[1], Marko Popovic[1], Branislav Kordic[1], and Huibiao Zhu[2]

[1] University of Novi Sad, Faculty of Technical Sciences, Trg D. Obradovica 6,
21000 Novi Sad, Serbia
{miroslav.popovic, marko.popovic, branislav.kordic}@rt-rk.uns.ac.rs
[2] East China Normal University, Shanghai Key
Laboratory of Trustworthy Computing,
Shanghai 200062, China
hbzhu@sei.ecnu.edu.cn

Abstract. State of the art formal verification is based on formal methods and its goal is proving given correctness properties. For example, a PSTM scheduler was modeled in CSP in order to prove deadlock-freeness and starvation-freeness. However, as this paper shows, using solely formal methods is not sufficient. Therefore, in this paper we propose a complete formal verification of trustworthy software, which jointly uses formal verification and formal model testing. As an example, we first test the previous CSP model of PSTM transaction scheduler by comparing the model checker PAT results with the manually derived expected results, for the given test workloads. Next, according to the results of this testing, we correct and extend the CSP model. Finally, using PAT results for the new CSP model, we analyze the performance of the PSTM online transaction scheduling algorithms from the perspective of the relative speedup.

Keywords: Formal Verification, Process Algebra, Transaction Scheduling, Python, Software Transactional Memory.

1. Introduction

As contemporary society is becoming increasingly dependent on software, which is ubiquitously used in everyday life, software verification is gaining paramount importance for our society and the environment. State of the art formal verification based on model checking is performed in two steps: (i) constructing a formal model of a given safety critical software, and (ii) proving that this formal model satisfies a given set of correctness properties, which consists of safety and liveness properties. For example, a PSTM (Python Software Transactional Memory) transaction scheduler was recently modeled (see [1]) in process algebra Communicating Sequential Processes (CSP) [2], in order to automatically prove the subject's deadlock-freeness (a safety property) and starvation-freeness (a liveness property) by the model checker PAT (Process Analysis Toolkit) [3].

* A preliminary version of this paper appears in the Proceedings of the 7th Conference on the Engineering of Computer Based Systems (ECBS), article no. 10, pages 1-10, Novi Sad, Serbia, May 2021 [25].

However, as this paper shows, conducting traditional formal verification in two steps (described above) as was, for example, done in [1], [4], [5], and [6] is not sufficient. As will be elaborated in more detail in Section 1.1 (related work), the main problem with the traditional two-steps formal verification is that the formal model constructed in the first step is not directly tested. Therefore, possible formal model shortcomings may not be discovered, and consequently, they may compromise the formal verification results.
As a solution to this problem, in this paper, we propose a method for a complete formal verification of trustworthy software, which jointly uses formal verification and formal model testing. Our method is based on the iterative procedure with the following steps (the procedure inputs comprise the initial formal model and the manually derived expected results):

1. Test the formal model by using the model checker and the expected results.
2. If the results are not as expected, correct the formal model and return to step 1.
3. Make the final report.

In the paper, we demonstrate our method using an example in which we applied the complete formal verification on the PSTM transaction scheduler. Within the example, we: (i) tested the previous CSP model of PSTM transaction scheduler by comparing the model checker PAT results with the manually derived expected results, for the given test workloads, (ii), according to the testing results, we corrected and extended the CSP model in each iteration (see the last two paragraphs in Section 1.1 for the history of all the corrections that were made in more iterations), and (iii) using PAT results for the final CSP model (henceforth called "the new CSP model"), we analyzed the performance of PSTM online transaction scheduling algorithms from the perspective of the relative speedup.

The rest of the paper is organized as follows. Section 1.1 presents closely related work, Section 2 presents the testing of the previous CSP model, Section 3 presents the new CSP model, Section 4 presents the performance analysis of the four PSTM online transaction scheduling algorithms, and Section 5 presents the paper conclusions.

1.1. Related Work

A brief overview of the most closely related research presented in this section covers formal verification and its testing, PSTM, and PSTM transaction scheduler formal verification chronology.

The formal verification process used in this paper is based on model checking. "Model checking is a technique for automatic verification of software and reactive system, and it consists of verifying some properties of the model of the system", see [7]. We selected three recent papers in order to illustrate formal verification state of the art [4], [5], and [6].

The paper [4] was motivated by the importance of the discovery and control service of an IoT system based on the Chord protocol, and the obvious fact that errors in concurrent systems are difficult to reproduce and find using solely program testing. The authors manually proved the correctness of the Chord protocol using the logic of time and knowledge with the respect to the set of possible executions (that maintain ring topology while the nodes can freely join or leave). The given proof was not

automatically verified in one of the formal proof assistants (e.g., Coq, Isabelle/HOL), and the authors only mention this as a possible challenge for their future work.

The paper [5] addresses the issues of safety-critical software verification and testing that are key requirements for achieving DO-178C and DO-331 regulatory compliance for airborne systems. The verification is performed by the symbolic model checker MCMAS+ that uses OBDDs. To validate their model, the engineers need to perform review and tracing activities. Review means fixing syntax errors, whereas tracing means checking model behaviour along all the possible traces within the complete model's state space. Both activities are conducted manually, so they are time-consuming and error-prone. Moreover, it seems that validation is not based on theoretically expected results, so the engineers are left to handle it according to their experience and intuition.

The paper [6] presents an approach for specifying, verifying, and deploying time-constrained business processes (BPs) in a mono-cloud, multi-edge context. At design-time, four stages take place: (i) specification in Business Process Model and Notation (BPMN), (ii) placement of tasks and data on cloud or edge, (iii) transformation from BPMN model to Timed Petri-Nets (TPN) model, and (iv) verification whether TPN model has any time violations (this is done automatically using a model checker like TINA). The main disadvantage of this approach is that the initial BPMN model is not checked. Additionally, the engineers: (i) do not derive some kind of theoretically expected results of the placement, and (ii) they do the placement manually as a series of trial-and-error attempts.

Testing as defined in the Cambridge Dictionary is "the process of using or trying something to see if it works, is suitable, obeys the rules, etc." [8]. In this paper, we test the formal model, or more precisely we test the formal verification process itself, in order to cross-check whether it produces the expected theoretical results. To the best of our knowledge, this is the first paper that advocates and demonstrates such an approach to the complete formal verification. So, the testing performed in this paper should not be confused with software testing, which is considered to be woefully inadequate for detecting errors in highly concurrent designs [7].

Transactional memory (TM) was conceived as an extension of a cache-coherence protocol that supports transactions executed on multicores, which operate on shared variables (called t-variables, or t-vars) [9], [10]. Software TM (STM) appeared as a TM implemented in software [11]. Python STM (PSTM) [12] is a general purpose STM for Python, which is applicable in a wide range of application-specific domains, from computational chemistry simulations [13], to data science, to IoT-based systems such as smart homes, vehicles, and cities.

PSTM was formally verified in three complementary papers. The authors of the first paper [14] constructed a formal model as a network of timed automata [15] representing: linear and cyclic transactions, the queue used by the remote procure call mechanism, and the transactional memory itself. Using the model checker UPPAAL [16], they automatically proved the following three properties: safety (atomicity), liveness (in a set of concurrent transactions, one must get committed), and termination (the cyclic transactions must eventually get committed).

The authors of the second paper [17] constructed a formal model as a group of CSP processes representing: a transaction, PSTM API, PSTM server, and PSTM dictionary. Using the model checker PAT, they automatically proved the following three properties:

deadlock freeness, ACI (atomicity, consistency, and isolation), and optimism (essentially the same as the liveness in [14]).

The authors of the third paper [18] constructed a formal PSTM push/pull semantic model as the mapping of operations within the PSTM's generic transactional algorithm to the four relevant push/pull rules: PULL, APPLY, PUSH, and CMT (the general push/pull semantic model is defined in [19]). They manually proved that the model satisfies correctness criteria for the relevant push/pull rules, and that therefore PSTM satisfies serializability (i.e. sequential consistency).

Generally, STM transactions are easy to program (as a simple sequential code that is executed atomically), they cannot deadlock (since one of the concurrent transactions gets committed), and they provide great performance for low concurrency workloads (since they are executed speculatively, i.e. without the overhead incurred by locks). However, in case of high concurrency workloads, the system performance may be degraded, because many transactions may get aborted and re-executed, or even worse, some of the transactions may be starved. Therefore, various transaction schedulers (or contention managers) were introduced in order to sustain good performance even for high concurrency workloads, e.g. [20], [21], [22].

PSTM transaction scheduler architecture and the four online scheduling algorithms, named Round Robin (RR), Execution Time Load Balancing (ETLB), Avoid Conflicts (AC), and Advanced Avoid Conflicts (AAC), were developed with the main goal to minimize the makespan (the total execution time) and consequently to maximize the throughput (the number of transactions per second) for an arbitrary workload [23], [24]. These algorithms were developed hierarchically, from the simplest RR to the most advanced AAC, and they were compared from the perspectives of time complexity, quality of theoretical initial schedules, and the experimentally measured speedup over RR and the number of aborts.

The theoretical schedules in [24] were manually derived for three test workloads: CFW (Conflict Free Workload), RDW (Read Dominated Workload), and WDW (Write Dominated Workload), which were previously used for the experimental evaluation in [23]. The experimental results and the theoretical results are well aligned, and this fact validates both the theoretical analysis in [24] and the algorithms' implementations in Python and their experimental evaluation in [23].

The authors of the paper [1] constructed the formal model of the PSTM scheduler architecture and the first three online transaction scheduling algorithms (RR, ETLB, and AC) from [23] as a group of CSP processes representing: an application, the scheduler input queue, the scheduler, the worker input queues, the workers, and the processes formalizing RR, ETLB, and AC algorithms. Using the model checker PAT, they automatically proved deadlock and starvation freeness properties and analyzed algorithms performance from the perspective of makespan, relative speedup, number of aborts, and throughput. However, instead of using test workloads from [23] and [24], they introduced three simplified test workloads: CFW, CW-1 (Conflict Workload 1), and CW-2 (Conflict Workload 2), so a direct comparison of their results with the results of [23] and [24] was not possible. Moreover, the formal verification in [1] was made with some shortcomings, which we discovered and remedied in [25].

In [25], we tested the CSP# model of the PSTM transaction scheduler introduced in [1] by comparing the model checker PAT results with the manually derived expected results, for the test workloads defined in [1], and we discovered six shortcomings. Next,

we extended the CSP# model in order to eliminate the discovered shortcomings. Finally, using PAT, we automatically analyzed the performance of the PSTM transaction scheduling algorithms from the perspective of makespan and throughput.

In this paper, we conduct a complete formal verification of the PSTM transaction scheduler for the three test workloads similar to those used in [23] and [24], in order to be able to compare the PAT's results with the results in [23] and [24]. More precisely, we first derived complete theoretical schedules for these workloads and tested the CSP# model from [25] with them. We discovered two additional shortcomings of the CSP# model from [25], and we extended the model accordingly. Next, using PAT, we analyzed the performance of the PSTM transaction scheduling algorithms from the perspective of the relative speedup. Finally, we compared the PATs results with the previous experimental results in [23].

2. Testing

This section presents the testing of the formal models developed in [1] and [25]. The main goal of the testing was to check whether the formal verification results are aligned with the theoretical results. The next three subsections present the testing method, theoretical schedules for the given test workloads, and the testing findings.

2.1. Testing Method

The testing method is based on the analysis of theoretical schedules, which are expected to be produced by the subject online transaction scheduling algorithms for the given test workloads. The method comprises the following steps:

- derive theoretical schedules;
- calculate the makespan and the number of aborts;
- compare the results of the previous step with the model checker PAT's results.

In addition to the test workloads used in [1] and [25], in this paper, we introduce the new test workload, named: CF (Conflict Free), RD (Read Dominated), and WD (Write Dominated) workloads, which are similar to the three test workloads (CFW, RDW, and WDW) used in [24]. The main difference between the former and the latter is that the former have 6 transactions, whereas the latter have 12 transactions. The second difference is that each transaction in the CF workload writes to a single t-variable, whereas each transaction in the CFW in [24] is a money transfer (MT) transaction i.e. it reads and writes two t-variables.

The new test workloads (CF, RD, and WD) are modeled in CSP# as arrays of six transactions [T_0, …, T_5]. There are three kinds of transactions: M, R, and W transactions. M transaction writes to a single t-variable from the set of t-variables {*A*, *B*, *C*, *D*, *E*, *F*}, R transaction sequentially reads all the t-variables, and W transaction sequentially writes to all the t-variables. The duration of the M transaction is 10 time units, whereas the durations of the R and W transactions are 40 time units each (these durations are the same as in [24]). The CF workload is a series of M transactions

operating of different t-variables, the RD workload is the interleaving of R transactions and M transactions, and the WD workload is the interleaving of W and M transactions.

2.2. Theoretical Schedules

As cores in a multicore processor use the same clock, a multicore processor is a synchronous system and therefore transactions assigned to separate workers' cores execute synchronously in parallel. The experiments conducted in [23] confirm this fact, and it was accepted as a postulate when formalizing PSTM transaction scheduler architecture and deriving the theoretical schedules for the given number of workers and given test workloads.

We derived the theoretical schedules for the workloads introduced in [1] in the preliminary version of this paper [25] (see them there). Here we consider the new test workloads (CF, RD, and WD). In order to save space, we derive the theoretical schedules for the PSTM scheduler architecture only with two workers, because these simple schedules are easy to comprehend and interpret even by readers not too familiar with this topic. The theoretical schedules for three test workloads CF, RD, and WD are shown in Fig. 1, Fig. 2, and Fig. 3, respectively.

In each figure, the queue with input transactions is shown at its top, where each transaction is labeled with its index (i.e. its ID in the CSP# model) and its type (M, R, or W). The expected schedules for individual online transaction scheduling algorithms are shown below the input queue, where the two workers' queues are labeled W_0 and W_1, respectively.

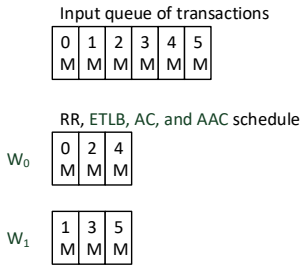


Fig. 1. The theoretical schedules for CF workload

Fig. 1 shows the expected schedule for the CF workload, which is conflict free, because all the transactions operate on different t-variables. Since there are two workers, the RR algorithm works by modulo two, so it assigns even transactions to W_0 and odd ones to W_1 in its first scheduling iteration. This schedule is executed without aborts (because there are no conflicts) and therefore this schedule happens to be the complete schedule with the makespan equal to 30 (3 x 10) and the number of aborts equal to 0.

Because all the transactions have the same duration, the ETLB algorithm behaves as the RR algorithm, and because there are no potential conflicts among transactions (since they operate on different t-variables), the AC and the AAC algorithms behave as the

ETLB algorithms, i.e. the RR algorithm. So, all the algorithms produce the same schedule shown in Fig. 1.

Figure 2 shows the expected schedule for the RD workload, which has 9 potential conflicts – these are the conflicts between each R transaction (T_0, T_2, and T_4) and each M transaction (T_1, T_3, and T_5) because R transactions sequentially read all the t-variables and M transactions (T_1, T_3, and T_5) write to a single t-variable (B, D, and F, resp.). Since there are two workers, the RR algorithm works by modulo two, so it assigns even transactions to W_0 and odd to W_1 in its first scheduling iteration. In this initial schedule, T_0 (assigned to W_0) is in conflict with all the M transactions (assigned to W_1). Because all the M transactions end before T_0, they all get committed, whereas T_0 gets aborted, rescheduled and successfully re-executed in the second iteration. So the complete schedule has the makespan equal to 160 (4 x 40), and the number of aborts is equal to 1.

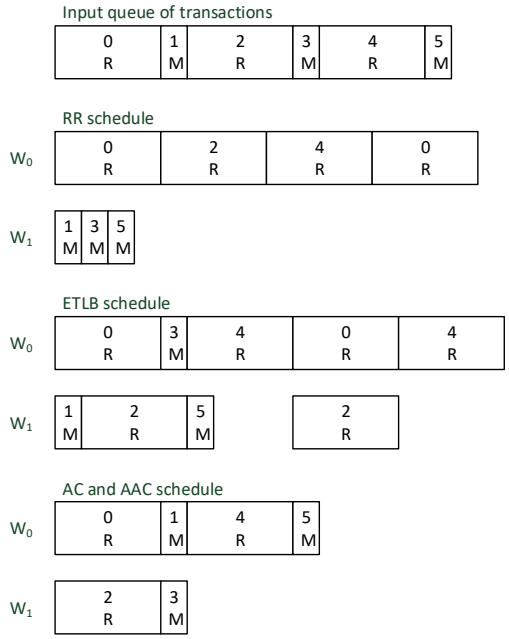


Fig. 2. The theoretical schedules for RD workload

The ETLB algorithm performs load balancing by making the following sequence of assignments in its first iteration (see Fig. 2, ETLB schedule): T_0 to W_0, T_1 to W_1, T_2 to W_1, T_3 to W_0, T_4 to W_0, and T_5 to W_1. However, this initial schedule is worse than the one produced by the RR algorithm, because it is executed with the following 3 conflicts causing 3 aborts: (i) T_0 and T_1 are in conflict, and T_1 is faster, so T_0 gets aborted, (ii) T_2 and T_3 and in conflict, and they end simultaneously, so in the worst case T_2 gets aborted (this is the worst case because T_2's duration is greater than T_3's) and (iii) T_4 and T_5 are in conflict, and T_5 is faster, so T_4 gets aborted. In the second iteration, the ETLB makes the following series of assignments (see Fig. 2): T_0 to W_0, T_2 to W_1, and T_4 to W_0. This schedule is executed without aborts because it is conflict free. So the complete schedule has the makespan equal to 170 (4 x 40 + 10), and the number of aborts is equal to 3.

The AC and the AAC algorithms do better than the ETLB algorithm because they are aware of the potential conflicts. Actually, for this particular workload, the AC algorithm produces the same result as the optimal AAC algorithm using a simple heuristic (schedule the next transaction to the least loaded worker if that does not cause a conflict). Both algorithms make the following sequence of assignments (see Fig. 2, AC and AAC schedule): T_0 to W_0, T_1 to W_0 (because of the potential conflict between T_0 and T_1), T_2 to W_1, T_3 to W_1 (because there is no conflict between T_2 and T_3), T_4 to W_0, and T_5 to W_0 (because of the potential conflict between T_4 and T_5). This initial schedule is conflict free, and therefore constitutes the complete schedule with the makespan equal to 100 (2 x 40 + 2 x 10), and the number of aborts equal to 0.

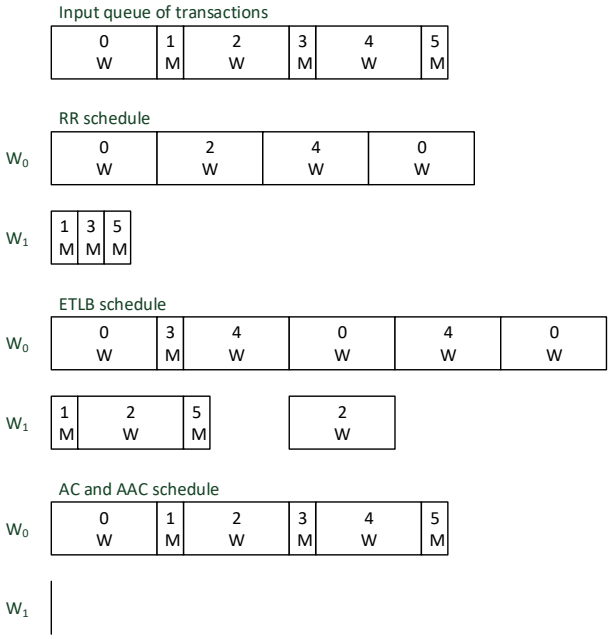


Fig. 3. The theoretical schedules for WD workload

Figure 3 shows the expected schedule for the WD workload, which has potential conflicts among all the transactions. The RR algorithm works by modulo two, so it assigns even transactions to W_0 and odd to W_1 in its first scheduling iteration. In this initial schedule, T_0 (assigned to W_0) is in conflict with all the M transactions (assigned to W_1). Because all the M transactions end before T_0, they all get committed, whereas T_0 gets aborted, rescheduled and successfully re-executed in the second iteration. So the complete schedule has the makespan equal to 160 (4 x 40), and the number of aborts is equal to 1. This result is practically the same as that for the RD workload (see Fig. 2).

The ETLB algorithm performs load balancing by making the following sequence of assignments in its first iteration (see Fig. 3, ETLB schedule): T_0 to W_0, T_1 to W_1, T_2 to W_1, T_3 to W_0, T_4 to W_0, and T_5 to W_1. This initial schedule is practically the same as for the RD workload (see Fig. 2), and for the same reasons T_0, T_2, and T_4 get aborted (in the worst case) and rescheduled in the second iteration. However, unlike the case for the RD

workload, T_0 and T_2 are now in conflict, so one of them (say T_0) gets aborted and re-executed in the third iteration. So, the complete schedule has the makespan equal to 210 (5 x 40 + 1 x 10) and the number of aborts equal to 4.

The AC and AAC algorithms serialize all the transactions (because of the conflicts among them) by assigning them all to W_0, whereas W_1 stays idle (see Fig. 3, AC and AAC schedule). So, this initial conflict-free schedule constitutes the complete schedule with the makespan equal to 150 (3 x 40 + 3 x 10) and the number of aborts equal to 0.

Table 1 summarizes the expected theoretical results for the makespan (*ms*) and the number of aborts (*na*) for the number of workers (*Index*) equal to 2 (derived above), as well as for the *Index* equal to 3, and 4 (which are derived analogously).

Table 1. The expected theoretical results for CF, RD, and WD workloads

Load & Alg.		*Index* = 2		*Index* = 3		*Index* = 4	
Load	Alg.	*ms*	*na*	*ms*	*na*	*ms*	*na*
CF	RR	30	0	20	0	20	0
	ETLB	30	0	20	0	20	0
	AC	30	0	20	0	20	0
	AAC	30	0	20	0	20	0
RD	RR	160	1	90	3	120	2
	ETLB	170	3	100	3	90	3
	AC	100	0	60	0	70	0
	AAC	100	0	50	0	50	0
WD	RR	160	1	180	7	160	3
	ETLB	210	4	180	6	170	6
	AC	150	0	150	0	150	0
	AAC	150	0	150	0	150	0

2.3. Testing Findings

Formal model testing was organized in two phases. The object of the first phase was the initial CSP# model from [1], whereas the object of the second phase was the extended CSP# model from [25]. The first phase was conducted in [25] using the three test workloads defined in [1], (CFW, CW-1, and CW-2), whereas the second phase was conducted in this paper using the three test workloads very similar to those used in [23] and [24], namely CF, RD, and WD workloads.

Table 2 summarizes the testing findings. In the first testing phase, we discovered items 1, 5, and 6 in Table 2, as well as the initial shortcomings in items 2-4. Next, in the second testing phase, we discovered items 7 and 8, as well as additional shortcomings in items 2-4.

By comparing the expected values of makespan and the values in [1], we found that, by an oversight, the time used (TU) for the model checking by the model checker PAT was interpreted as equal to the makespan. Next, the fact that the expected values for the number of aborts and the values in [1] were different, indicated that the real schedules from [1] are different from the expected schedules derived in [25]. A rather tedious reconstruction of the real schedules from the model checker PAT's log files confirmed this indication.

Table 2. The summary of testing findings

No.	Finding
1	PAT's time used (TU) reported as the makespan
2	The macro isConflict shortcomings
3	Pessimistic concurrency control
4	Asynchronous transactions' execution
5	The number of workers fixed to *Index* = 2
6	AAC algorithm not supported
7	Read operations and the star convention not supported
8	The macros findmin and findmax shortcomings

The three main root causes of the discrepancies between the real and the expected schedules (the findings 2-4 in Table 2) are the following: (i) the macro isConflict, which checks whether there are conflicts between the next transaction to be scheduled x and any already scheduled transaction i, included the case $i = x$, (ii) the CSP# model from [1] uses the pessimistic concurrency control, whereas the real PSTM uses the optimistic concurrency control (see Sec. 2-1 in [10]), and (iii) the workers in the CSP# model from [1] execute transactions asynchronously, and therefore this model violates the postulate of the synchronous transaction execution introduced in Section 2.2.

By a quick inspection of the model from [1], we saw that the number of workers is fixed to *Index* = 2 and that the AAC algorithm is not supported (items 5-6 in Table 2). In the second testing phase, we needed to model RD and WD workloads, and at that time we discovered that the model in [25] does not support read operations and the star convention by which * means all the t-variables (item 7 in Table 2). While fixing item 7, we found additional shortcomings related to items 2-4 and we discovered that in the case when the array *load* has equal elements, both macro findmin and macro findmax return the index of the last of them, whereas in the theoretical schedules we took the index of the first such element (item 8).

3. The New Model

This section presents the new CSP# model, which evolved from the previous model presented in [25]. The next subsections present the three most important parts of the model related to (i) conflict detection, (ii) optimistic concurrency control, and (iii) synchronous transactions' execution. This organization was made according to the findings 2-4 and 7 in Section 2.3 (other extensions are skipped because of the space limits).

3.1. Conflict Detection

The macro isConflict sets the variable *IsConflict* to 0 if the new transaction x starting at *t1* and ending at *t2* is not in conflict with the already scheduled transactions; otherwise it sets *IsConflict* to 1, see Algorithm 1. Initially, it sets the variables i and *IsConflict* to 0 (lines 4-5; i is the index of the i-th transaction). Next, it repeats the loop while i is less than TNum (the number of transactions) and there is no conflict (lines 6-18).

Within the loop, it performs a series of nested if statements, which is equivalent to a single if statement with a conjunction of all the conditions, in order to check whether the transaction *i* is scheduled (line 7), and the transactions *i* and *x* overlap in time (lines 8-9), and *i* is not *x*, and the transactions *i* and *x* operate on the same t-variable or one of them operates all the t-variables as indicated by the constant Star (line 10), and the transactions *i* or *x* write to the t-variable(s) (line 11), and if yes it sets *isConflict* to 1.

Algorithm 1. The macro isConflict

```
01: var IsConflict = 0;
02: #define isConflict(x, t1, t2)
03: {
04:   var i = 0;
05:   IsConflict = 0;
06:   while(i<TNum && IsConflict!=1) {
07:     if(T_isScheduled[i]==1) {
08:       if((T_StartTime[i]<t2 && t2<=T_EndTime[i])||
09:       (t1<T_EndTime[i] && T_EndTime[i]<=t2)) {
10:         if((i!=x)&&(T_Var[i]==T_Var[x]||T_Var[i]==Star||T_Var[x]==Star)) {
11:           if(T_VarOp[i]==Wtvar || T_VarOp[x]==Wtvar) {
12:             IsConflict = 1;
13:           }
14:         }
15:       }
16:     }
17:     i++;
18:   }
19: };
```

3.2. Optimistic Concurrency Control

The PSTM's optimistic concurrency control model is shown in Algorithm 2. This model is much simpler than the complete PSTM models developed in [14] and [17], and it was made as such, in order to keep the model state space exploration fast and feasible.

The array *T_VarVer* (line 1) models the PSTM dictionary, and its elements store the versions of the respective t-variables (TvarNum is the number of t-variables). Since the number of transactions in a workload is equal to TNum, we bounded the number of versions for each t-variable by TNum, in order to reduce the state space to be explored (if TNum transactions update a single t-variable, its final version would be TNum-1). The number of a t-variable versions is effectively bounded by counting them with modulo TNum (line 13 for single t-variable updates and line 18 for the updates using the star convention).

In order to support updates using the star convention (i.e. updates of all the t-variables), we define the *star version* (i.e. the version of all the t-variables) as the sum of the versions of all the t-variables. The macro calciver (called in line 6) stores its result in the variable *iver* (line 2), and the result is either the star version if the argument *key* is equal to Star, or the version *T_VarVer*[*key*] if *key* is not equal to Star.

The process *Pstm* models the PSTM itself (lines 3-29). The worker processes send their messages to *Pstm* over the channel *worker2pstm*, whereas *Pstm* sends its replies to the worker process W_i via the channel *pstm2worker*[*i*].

Algorithm 2. The PSTM's optimistic concurrency control model

```
01: var T_VarVer[TvarNum]:{0..TNum-1} = [0(TvarNum)];
02: var iver:{0..TvarNum*TNum} = 0;

03: Pstm() =
04:   worker2pstm?i.req.op.key.ver->
05:   atomic {
06:     {call(calciver, key)}->
07:     if(req == GetVars) {
08:       pstm2worker[i]!iver->Pstm()
09:     } else { // commitVars
10:       if(ver == iver) { // T_VarVer[key] replaced with iver
11:         if(op == Wtvar) {
12:           if(key != Star) {
13:             {T_VarVer[key] = (T_VarVer[key]+1)%TNum}->
14:             pstm2worker[i]!1->Pstm()
15:           } else {
16:             {
17:               var ii = 0;
18:               while(ii<TvarNum){T_VarVer[ii]=(T_VarVer[ii]+1)%TNum;ii++}
19:             }->
20:             pstm2worker[i]!1->Pstm()
21:           }
22:         } else {
23:           pstm2worker[i]!1->Pstm()
24:         }
25:       } else {
26:         pstm2worker[i]!0->Pstm()
27:       }
28:     }
29:   };
```

The compound messages sent by W_i to *Pstm*, over the channel *worker2pstm*, have the format *i.req.op.key.ver*, where *i* is the worker's index, *req* is the type of request (the existing types of requests are: GetVars and CommitVars, which correspond to the PSTM API functions getVars and commitVars, respectively), *op* is the type of operation (the existing types of operations are: Wtvar and Rtvar, which correspond to the write and read operations, respectively), *key* is the index of the t-variable or Star, and *ver* is the version of the t-variable or the star version.

The replies sent from *Pstm* to W_i, over the channel *pstm2worker*[*i*], have a single element whose semantics depend on the type of the request: for GetVars request, the reply is the t-variable' version, whereas for CommitVars request, the reply is either 0 or 1 whether the transaction gets aborted or successfully committed, respectively.

After receiving the message *i.req.op.key.ver* (line 4) the process *Pstm* atomically (line 5) serves the request as follows. It calls the macro calciver (line 6) to set the *iver* for the given *key* (which may be an index of a t-variable or Star). Then it checks the type of the request (line 7). In the case of the GetVars request (line 8), it returns *iver* (which may be the version of a t-variable or the star version). Alternatively, in the case of the CommitVars request, *Pstm* checks whether the version *ver* from the input message is equal to *iver* (line 10). If they are equal, *Pstm* checks the type of *op* (line 11).

If *op* is equal to Wtvar (line 11), *Pstm* compares *key* with Star (line 12). If *key* is not equal to Star, *Pstm* increments the current version of the t-variable key (line 13), whereas if *key* is equal to Star, *Pstm* increments current versions of all the t-variables (line 18), and in both cases *Pstm* sends the reply 1 signaling successful commit (line 20). If *op* is not equal to Wtvar (i.e. *op* is the read operation), *Pstm* sends the reply 1 signaling successful commit (line 23). If *ver* is not equal to *iver*, *Pstm* sends the reply 0 signaling abort (line 26).

As defined above, *Pstm* provides optimistic concurrency control by servicing two types of requests made by synchronous concurrent workers executing transactions. Each transaction starts with the GetVars request (to get local copies of specified t-variables), proceeds with some data processing (on local copies), and ends with the CommitVars request (to update the specified shared t-variables).

3.3. Synchronous Transactions' Execution

The worker's behavior model is shown in Algorithm 3 (excluding unimportant parts). The worker W_i initially behaves as the process *Worker*(*i*) (line 1). After receiving the signal READY from the scheduler, it behaves as the process *Worker_1*(*i*) (line 2).

The process *Worker_1*(*i*) is an iterative process (line 3). In each iteration, it checks the input queue *Queue*[*i*] (line 4). If there are no transactions in the input queue, *Worker_1*(*i*) executes the process *Worker_2*(*i*), sends the signal done to the scheduler, and continues behaving as the process *Worker*(*i*) (line 5). Alternatively, if there is a transaction in the input queue (line 6), *Worker_1*(*i*) dequeues the transaction, estimates the duration of the transaction, sets the variables related to the transaction (these steps are not shown in Algorithm 3), and executes the process *Working*(*i, currentT*[*i*]), where *currentT*[*i*] is the index of the current transaction executed by W_i (line 7).

By the definition of the parallel composition operator ||, all the parallel processes must simultaneously engage in their *common events* (i.e. the events in the intersection of their alphabets) [2]. In Algorithm 3, all the workers synchronize using the so-called *lock-step synchronization*, i.e. they engage in their common event *tick* simultaneously. Therefore, all the workers must engage in the same number of ticks, *nt*, in each scheduling iteration. In order to calculate *nt*, let nt_i be the total load plus the number of transactions allocated to W_i (the number of transactions is added because starting each transaction requires one *tick*). Then, *nt* is the maximal nt_i, nt_m (nt_i for the worker $i = m$), for $i = 0, \ldots,$ Index – 1:

$$nt_i = load_i + \text{count}(Queue_i) \,. \tag{1}$$

$$nt = nt_m = max_i \, nt_i \,. \tag{2}$$

Next, let it_i be the number of idle ticks that must be executed by W_i after it processed all the transactions from its input queue, $Queue_i$:

$$it_i = nt - nt_i \,. \tag{3}$$

The processes *Working*(*i, txn*) and *Working_1*(*i, txn*) model the behavior of W_i processing its current transaction *txn*. The former models the start of the transaction, whereas the latter models the rest of the transaction.

Algorithm 3. The worker's behavior

```
01: Worker(i) =
02:   comSW[i]?READY->Worker_1(i);

03: Worker_1(i) =
04:   if(Queue[i].Count() == 0) {
05:     Worker_2(i); output!done -> Worker(i)
06:   } else {
        …
07:     Working(i, currentT[i]);
        …

08: Worker_2(i) =
09:   if(idleTicks[i] > 0) {
10:     tick -> tau{idleTicks[i]--} -> Worker_2(i)
11:   } else {Skip};

12: Working(i, txn) =
13:   tick ->
14:   worker2pstm!i.GetVars.0.currentT_Var[i].0 ->
15:   pstm2worker[i]?tvarver ->
16:   {currentT_VarVer[i] = tvarver; workertime[i]++} ->
17:   Working_1(i, txn);

18: Working_1(i, txn) =
19:   tick ->
20:   if(workertime[i] < currentT_Time[i]) {
21:     working{workertime[i]++} ->
22:     Working_1(i, txn)
23:   } else {
24:     worker2pstm!i.CommitVars.T_VarOp[txn].currentT_Var[i].currentT_VarVer[i] ->
25:     pstm2worker[i]?resp ->
26:     {currentT_Cmt[i] = resp; workertime[i] = 0} ->
27:     Skip
28: };
```

The process *Working*(*i, txn*) (lines 12-17): (i) sends the GetVars request to the process *Pstm* (line 14) and receives the value of *tvarver* (which is either the version of the t-variable *key* if *key* is not Star or the star version if *key* is equal to Star) in the reply from *Pstm* (line 15), and (ii) stores *tvarver* into *currentT_VarVer*[*i*] and increments its working time by updating *workertime*[*i*] (line 16).

The process *Working_1*(*i, txn*) (lines 18-28) checks whether it has to do more processing (line 20), and if yes, it increments its working time (line 21). At the end of the transaction (line 23), it: (i) sends the CommitVars request to the process *Pstm* (line 24) and receives the *Pstm*'s reply *resp*, which is 1 if the transaction got successfully committed, otherwise it is 0 (line 25), and (ii) stores *resp* into *currentT_Cmt*[*i*] and resets the working time (for the next scheduling iteration) by clearing *workertime*[*i*] (line 26).

After the process *Working_1*(*i, txn*) terminates, the process *Worker_1*(*i*) resumes at line 5, where it executes the process *Worker_2*(*i*) (lines 8-11). The latter checks whether

the number of its idle ticks *idleTicks*[*i*] is greater than zero (line 9), and if yes, decrements *idleTicks*[*i*] (line 10); otherwise it terminates (line 11).

4. Formal Verification

This section presents the formal verification for the new CSP# model. The next two sections present the verification results and the performance analysis.

4.1. Verification Results

First, we define the following three system conditions, which are used in the assertions:
- the condition *Done* requires that *snum* is equal to TNum (where *snum* is the number of the successfully executed transactions), which means that all the transactions have been successfully executed;
- the condition *MaxNA* requires that *na* is nonnegative and that Done is satisfied;
- the condition *MaxMS* requires that *ms* is nonnegative and that *Done* is satisfied.

Next, we introduce the five assertions that were checked for each workload defined in the previous paper [25] (CFW, CW-1, and CW-2) and in this paper (CF, RD, and WD workloads) and each version of the system, where the version of the system is defined by the given number of workers (Index) and the given online transaction scheduling algorithm (RR, ETLB, AC, and AAC). The first assertion corresponds to a safety property, and the other assertions correspond to liveness properties.

These five assertions are defined as follows:
- the system is deadlock-free;
- the system reaches a state satisfying the condition *Done*;
- all the system's evolution paths satisfy the CSP# LTL formula []<>*comSA*.complete, which means that always eventually ([]<>) the signal complete is sent from the scheduler to the application, over the channel *comSA*;
- the system reaches a state satisfying the condition *MaxNA* over a path that maximizes *na* (this is achieved by using the clause "with max(*na*)", which instructs PAT to search and report the maximal value of *na*);
- the system reaches a state satisfying the condition *MaxMS* over a path that maximizes the expression (*ms*+*na*) (this is achieved by using the clause "with max(*ms*+*na*)").

Using PAT, we checked these five assertions for each of the six workloads (CFW, CW-1, CW-2, CF, RD, and WD workloads) and for each of the system versions. There are system versions with 2, 3, and 4 workers (i.e. Index = 2, 3, 4), and for each of the 4 online transaction scheduling algorithms (RR, ETLB, AC, and AAC), i.e. there are 12 system versions (3 x 4 = 12) in total. All the 360 assertions (6 x 12 x 5 = 360) that we checked were found to be valid (i.e. satisfied).

The third assertion for the case with 4 workers, WD workload, and ETLB algorithm was the most time-consuming to validate. The verification statistics reported by PAT for this case are the following: 14386432 visited states, 46722717 passed transitions, 526 s

of used time, and 6950446 KB of used memory (on a PC with 16GB DDR4 memory and CPU i7-8750H 2.2 GHz with turbo bust to 4.1 GHz).

We also manually checked the values for *na* and *ms* reported by the last two assertions, and they matched the expected results. The expected results for the workloads defined in this paper are given in Table 1, whereas the expected results for the workloads defined in [25] are given in Table 1 and Table 3 in [25].

4.2. Performance Analysis

In this section, we: (i) introduce the necessary definitions, (ii) analyze the performance of the PSTM online transaction scheduling algorithms from the perspective of the relative speedup using the theoretical results (confirmed by PAT) from [25] and this paper, and (iii) we validate the theoretical results from this paper with the experimental results from [23].

First, we define the *number of independent transactions* within a workload, *L*, which we use to quantitatively characterize the level of parallelism for a given *L*.

Definition 1. The number of independent transactions, *nit*, is the max number of transactions in *L* that could be scheduled online (i.e. without changing the transactions arrival order), in parallel, on an infinite number of processors, without a conflict.

For example, the values of *nit* for WD, RD, and CF workloads, are 1, 3, and 6, respectively. So, WD has the lowest level of parallelism, and CF has the highest.

Next, we define the *relative speedup* of an algorithm A over the algorithm RR.

Definition 2. The relative speedup, *s*, of an algorithm A over the algorithm RR is defined as the ratio ms_{RR}/ms_{A}, where ms_{RR} and ms_{A} are the makespans for the algorithms RR and A, respectively.

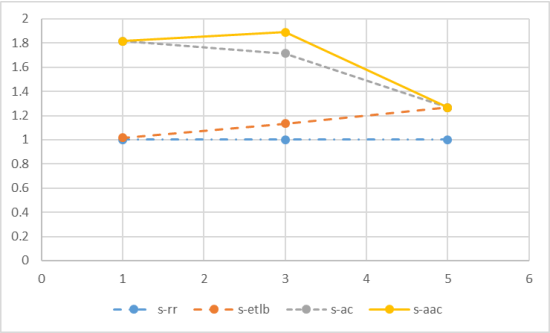


Fig. 4. The average relative speedup for the theoretical results from [25]

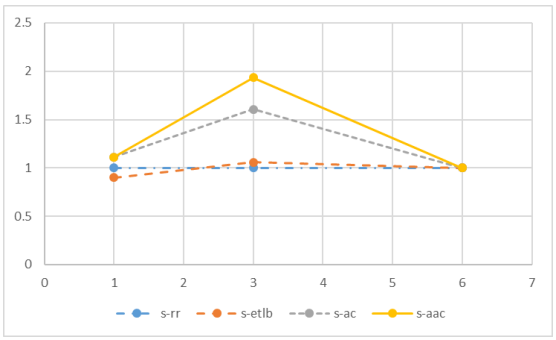


Fig. 5. The average relative speedup for the theoretical results in this paper

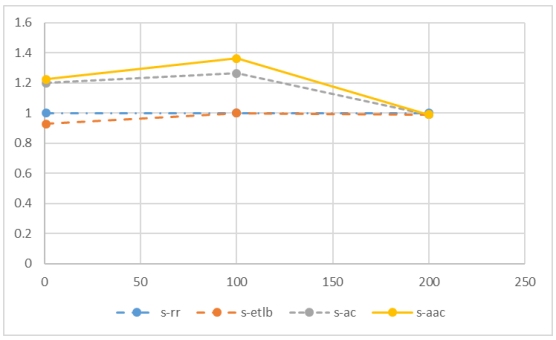


Fig. 6. The average relative speedup for the experimental results from [23]

Using data from [25], we calculated the average relative speedup for each algorithm (RR, ETLB, AC, and AAC) and each workload (CW-2, CW-1, and CFW), which are characterized by their *nit* values (1, 3, and 5); the average is calculated over the values of relative speedup for Index = 2, 3, and 4 (i.e. 2, 3, and 4 workers).

The values of the average *s* for RR, ETLB, AC, and AAC algorithms are illustrated in Fig. 4 with the curves denoted as *s-rr*, *s-etlb*, *s-ac*, and *s-aac*, respectively.

Similarly, using data from this paper, we calculated the average relative speedup for each algorithm (RR, ETLB, AC, and AAC) and each workload (WD, RD, and CF workloads), which are characterized by their *nit* values (1, 3, and 6); the average is again calculated over the values of relative speedup for Index = 2, 3, and 4. The values of the average *s* for RR, ETLB, AC, and AAC algorithms are illustrated in Fig. 5 with the curves again denoted as *s-rr*, *s-etlb*, *s-ac*, and *s-aac*, respectively.

After analyzing the data and the shape of the curves in Fig. 4 and Fig. 5, we may conclude that according to their performance, in terms of the average relative speedup, the PSTM online transaction scheduling algorithms can generally be ranked as follows: (i) AAC is the best, (ii) AC is worse than AAC, (iii) ETLB is worse than AC, and (iv) RR is the worst. There is a single exception to the general conclusion: in the case of the WD workload, ETLB is worse than RR, see Fig. 5. This exception is not unexpected,

because, for some workloads (like the WD workload), the simple RR algorithm may outperform the ETLB algorithm that sometimes may be too greedy.

Next, we compare the theoretical results in this paper (illustrated in Fig. 5) with the experimental results from [23]. It is important to realize that these results are not quantitatively comparable, because of the following reasons: (i) the workloads from [23] have 200 transactions each (whereas the workloads in this paper have 6 transactions each), (ii) the duration of transactions in the workloads from [23] are: M transaction takes 0.65 ms whereas R and W transactions take 45 ms each (whereas the duration of transactions in this paper are: M takes 10 time units whereas R and W take 40 time units), (iii) the experiments in [23] were made only for the two cases: with 2 and 3 workers, and (iv) the experiments in [23] were made on a PC with 4 cores, consequently some schedules for the case with 3 workers were compromised by the local OS, since 4 cores were not sufficient to host 3 workers and the OS processes.

On the other hand, we had an intuition that these theoretical and experimental results should be qualitatively comparable, because of the following reasons: (i) the workloads from [23] and from this paper have the same rather simple patterns, which when scheduled on a small number of workers (like in [23]) should yield schedules having rather small periods, and (ii) if we take the time unit to be one ms, at least the durations of R and W transactions would be the same.

Therefore, using the data from [23], we calculated the average relative speedup for each algorithm and each workload (namely WDW, RDW, and CFW, which are characterized by their *nit* values: 1, 100, and 200, respectively); the average is calculated over the values of relative speedup for Index = 2 and 3 (2 and 3 workers). The values of the average *s* for RR, ETLB, AC, and AAC algorithms are illustrated in Fig. 6 with the curves again denoted as *s-rr*, *s-etlb*, *s-ac*, and *s-aac*, respectively.

After analyzing the shapes of the curves in Fig. 5 and in Fig. 6, we may conclude that the theoretical results in this paper are qualitatively well aligned with the experimental results from [23].

5. Conclusions

Modern society is becoming strongly dependent on the pervasive use of software in everyday life, and therefore software verification is becoming extremely important. Traditionally, formal methods have been seen as a key for the successful design of safety critical systems. However, in this research, we learned that using solely formal methods, like CSP, is not sufficient. As a solution, we proposed the method for the complete formal verification of trustworthy software, which jointly uses formal verification and formal model testing.

In the paper, we applied this method and conducted the complete verification of the PSTM online transaction scheduler and the accompanying scheduling algorithms, through the iterative procedure of testing and correcting/extending the initial CSP model. The final result of this iterative procedure is called the new CSP model. Using this new CSP model, we analyzed the performance of the PSTM online transaction scheduling algorithms from the perspective of the relative speedup, and got the results that were as expected and well aligned with the previous research [23], [24], and [25].

The main difficulty that we faced in this research was to make a realistic model that is simple enough to be checkable on an off-the-shelf PC that was at our disposal. This difficulty caused the main limitations of the presented example: (i) the limited number of workers (up to 4), (ii) the limited number of test workloads (6 altogether), (iii) the limited number of transactions in a workload (up to 6), (iv) fixed transactions' durations, and (v) the transactions either operate on a single t-variable or on all the t-variables (using the star convention). In our future work we plan: (i) to address these limitations, and (ii) to apply the complete formal verification on other software architectures.

Acknowledgement. This work was partially supported by: (i) the Ministry of Education, Science and Technology Development of Republic of Serbia under Grant 451-03-68/2020-14/200156, and (ii) the "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (Grant No. 22510750100).

References

1. Xu, C., Wu, X., Zhu, H., Popovic, M.: Modeling and Verifying Transaction Scheduling for Software Transactional Memory using CSP. In Proceedings of the 13th Theoretical Aspects of Software Engineering Symposium. IEEE, Guilin, China, 240-247. (2019)
2. Hoare, C.A.R.: Communicating Sequential Processes. Prentice/Hall International, New Jersey, USA. (1985)
3. Si, Y., Sun, J., Liu, Y., Dong J. S., Pang, J., Zhang, S. J., Yang, X.: Model checking with fairness assumptions using PAT. Frontiers of Computer Science, Vol. 8, No. 1, 1-16. (2014)
4. Marinković, B., Ognjanović, Z., Glavan, P., Kos, A., Umek, A.: Correctness of the Chord Protocol. Computer Science and Information Systems, Vol. 17, No. 1, 141-160. (2020)
5. Elqortobi, M., El-Khouly, W., Rahj, Amine R., Bentahar, J., Dssouli, R.: Verification and Testing of Safety-Critical Airborne Systems: a Model-based Methodology. Computer Science and Information Systems, Vol. 17, No. 1, 271-292. (2020)
6. Cheikhrouhou, S., Kallel, S., Guidara, I., Maamar, Z.: Business Process Specification, Verification, and Deployment in a Mono-Cloud, Multi-Edge Context. Computer Science and Information Systems, Vol. 17, No. 1, 293-313. (2020)
7. Berard, B., Bidoit, J. M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, Ph., McKenzie, P.: Systems and Software Verification. Springer, Berlin, Germany. (1999)
8. Testing. Cambridge online dictionary. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/testing (current September 2021)
9. Herlihy, M., Moss, J. E. B.: Transactional memory: Architectural support for lock-free data structures. In Proceedings of the 20th Annual International Symposium on Computer Architecture. ACM, San Diego, CA, USA, 289-300. (1993)
10. Harris, T., Larus, J. R., Rajwar, R.: Transactional Memory, 2nd edition. Morgan and Claypool, San Rafael, CA, USA. (2010)
11. Shavit, N., Touitou, D.: Software transactional memory. In Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing. ACM, Ottawa, Ontario, Canada, 204-213. (1995)
12. Popovic, M., Kordic, B.: PSTM: Python software transactional memory. In Proceedings of the 22nd Telecommunications Forum (TELFOR). IEEE, Belgrade, Serbia, 1106-1109. (2014)
13. Kordic, B., Popovic, M., Popovic, M., Goldstein, M., Amitay, M., Dayan, D.: An Evolutionary Computational System Architecture Based on a Software Transactional

Memory. Revue Roumaine des Sciences Techniques. Ser. Electrotechnique et Energetique, Vol. 61, No. 1, 47-52. (2021)
14. Kordic, B., Popovic, Ghilezan, M., S.: Formal Verification of Python Software Transactional Memory Based on Timed Automata. Acta Polytechnica Hungarica, Journal of Applied Sciences, Vol. 16, No. 7, 197-216. (2019)
15. Alur, R., Dill, D. L.: A theory of timed automata. Theoretical Computer Science, Vol. 126, No. 2, 183-235. (1994)
16. Behrmann, G., David, A., Larsen, K. G.: A Tutorial on Uppaal. In: Bernardo, M., Corradini, F. (eds.): Formal Methods for the Design of Real-Time Systems. Lecture Notes in Computer Science, Vol. 3185. Springer-Verlag, Berlin Heidelberg New York, 200-236. (2004)
17. Liu, A., Zhu, H., Popovic, M., Xiang, S., Zhang, L.: Formal Analysis and Verification of the PSTM Architecture Using CSP. Journal of Systems and Software, Vol. 165, 1–14. (2020)
18. Popovic, M., Popovic, M., Ghilezan, S., Kordic, B.: Formal Verification of Local and Distributed Python Software Transactional Memories. Revue Roumaine des Sciences Techniques. Ser. Electrotechnique et Energetique, Vol. 64, No. 4, pp. 423–428. (2019)
19. Koskinen, E., Parkinson, M.: The Push/Pull Model of Transactions. In Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation. Morgan Kaufmann, Portland, Oregon, USA, 186-195. (2015)
20. Yoo R.M., Lee, H.-H.S.: Adaptive transaction scheduling for transactional memory systems. In Proceedings of the 20th annual symposium on Parallelism in algorithms and architectures. ACM, Munich, Germany, 169–178. (2008)
21. Ansari, M., Luján, M., Kotselidis, C., Jarvis, K., Kirkham, C., Watson, I.: Steal-on-Abort: Improving Transactional Memory Performance through Dynamic Transaction Reordering. In: Seznec, A., Emer, J., O'Boyle, M., Martonosi, M., Ungerer, T. (eds.): High Performance Embedded Architectures and Compilers. HiPEAC 2009. Lecture Notes in Computer Science, Vol. 5409. Springer, Berlin, Heidelberg, 4-18. (2009)
22. Dolev, S., Hendler, D., Suissa, A.: CAR-STM: scheduling based collision avoidance and resolution for software transactional memory. In Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing. ACM, Toronto, Ontario, Canada, 125-134. (2008)
23. Popovic, M., Kordic, B., Popovic, M., Basicevic, I.: Advanced Algorithm for Scheduling TM Transactions with Conflict Avoidance. In Proceedings of the 25th Telecommunications Forum (TELFOR). IEEE, Belgrade, Serbia, 844-847. (2017)
24. Popovic, M., Kordic, B., Popovic, M., Basicevic, I.: Online Algorithms for Scheduling Transactions on Python Software Transactional Memory. Serbian Journal of Electrical Engineering, Vol. 16, No. 1, 85-104. (2019)
25. Popovic, M., Popovic, M., Kordic, B., Zhu, H.: PSTM Transaction Scheduler Verification Based on CSP and Testing. In Proceedings of the 7th Conference on the Engineering of Computer Based Systems. ACM, Novi Sad, Serbia, Article No. 10, 1-10. (2021)

Miroslav Popovic received his Dipl. Eng., M.Sc., and Ph.D. degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 1984, 1988 and 1990, respectively. He is a Full Professor at the University of Novi Sad from 2002. Currently he is giving courses on parallel programming, real-time systems programming, and inter-computer communications and computer networks. In the past, he has supervised many real-world projects for the industry, mostly in real-time and embedded systems. His research interests are engineering of computer-based systems, intelligent distributed systems, and security. He has authored or co-authored about 30 peer-reviewed journal papers, more than 120 conference papers, and the book Communication protocol engineering, Second Edition (CRC Press, Taylor & Francis Group, 2018).

Marko Popovic received his B.Sc., M.Sc., and PhD degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 2015, 2017, and 2020, respectively. Currently he is Scientific Researcher affiliated with the RT-RK Institute of Computer Based Systems, Novi Sad, Serbia. His research interests are in the areas of engineering of computer-based systems and intelligent distributed systems. He has authored or co-authored more than 15 scientific papers.

Branislav Kordic received B.Sc, M.Sc, and PhD degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 2012, 2013, and 2020, respectively. He was a teaching assistant at the Faculty of Technical Sciences, University of Novi Sad. Currently he works as a professional software engineer. His domains of interest are real-time systems and systems for parallel and distributed computing.

Huibiao Zhu is currently a professor in East China Normal University, Shanghai. He earned his Ph.D. degree in formal methods from London South Bank University, London, in 2005. During these years, he has studied various semantics and their linking theories for Verilog, SystemC, web services and probability system. He was the Chinese PI of the Sino-Danish Basic Research Center IDEA4CPS.

Received: September 08, 2021; Accepted: May 11, 2022.

Computer Science and Information Systems 20(1):329–357 https://doi.org/10.2298/CSIS211001059B

Supporting 5G Service Orchestration with Formal Verification*

Peter Backeman[1], Ashalatha Kunnappilly[2], and Cristina Seceleanu[1]

[1] Mälardalen University,
Västerås, Sweden
{peter.backeman,cristina.seceleanu}@mdu.se
[2] Alstom,
Västerås, Sweden
ashalatha.kunnappilly@alstomgroup.com

Abstract. The 5G communication technology has the ability to create logical networks, called network slices, which are specifically carved to serve particular application domains. Due to the mix of different application criticality, it becomes crucial to verify if the applications' service level agreements are met. In this paper, we propose a novel framework for modeling and verifying 5G orchestration, considering simultaneous access and admission of new requests to slices as well as virtual network function scheduling and routing. By combining modeling in user-friendly UML, with UPPAAL model checking and satisfiability-modulo-theories-based model finding, our framework supports both modeling and formal verification of service orchestration. We demonstrate our approach on a e-health case study showing how a user, with no knowledge of formal methods, can model a system in UML and verify that the application meets its requirements.

Keywords: 5G, model checking, SMT, UML

1. Introduction

The fifth generation of wireless technology, 5G, has the potential to support a variety of applications with different requirements, be it low latency, high bandwidth or increased number of connections. This is ensured via its ability to create end-to-end *network slices*, tailored to support respective application requirements [15]. A 5G network slice is composed of several *virtual network functions* (VNFs) that are chained in order to meet the application's functionality. Most often, VNFs have constraints on CPU, RAM, storage, which need to be met by the servers that host them. In addition, servers are connected via links, hence chaining VNFs incurs additional resource overhead in terms of link bandwidth and latency. Adding to the complexity, VNFs can be shared between slices that are requested simultaneously by various 5G user equipment (UE).

To analyze if a 5G network slice instance can effectively serve its applications, one needs to ensure that the respective VNFs are allocated, scheduled, and routed according to the current network scenario. This is referred to as **dynamic 5G service orchestration**. For instance, when applications of different criticality share network resources, one needs to ensure that all slices facilitate meeting the requirements of considered applications.

* This is an extension of previous work published at ECBS 2021 [12].

Although much research has been devoted to solving the 5G service orchestration problem by providing optimal VNF allocation and routing schemes [19,14], there is a lack of endeavors that provide modeling and formal verification frameworks that can analyze such schemes to provide guarantees of the intended system behavior. In this paper, we propose such a modeling and formal analysis framework that combines user-friendly UML-based modeling [8] with mathematical assurance via exhaustive model checking in UPPAAL [13] and model finding using Z3 [21]. This work builds on our previous results [11], the UML5G-SO framework, which allows for modeling and analysis or VNF allocation and routing, assuming static worst-case scenarios. We augment the UML5G-SO framework to support dynamic system behavior stemming from slice requests from UE, scheduling, link utilization, etc. Our contribution includes the following: (i) extending the UML5G-SO profile with stereotypes to model UE and the controllers for handling and monitoring the dynamic requests, (ii) defining the behavioral view of a system built based on our profile in terms of restricted UML statechart patterns (see Sec. 4.4), (iii) defining pattern-based timed automata semantics for restricted statechart patterns, enabling model-checking UML5G-SO behaviors with UPPAAL [13], (iv) defining an alternative semantics in first-order logic with linear arithmetic for the restricted statechart patterns, as well as (v) implementing tool support for the automatic generation of UPPAAL and SMT models from UML diagrams. To demonstrate our approach, we consider a case study of simultaneous access of shared network resources by two applications of different criticality. This paper is an extension of our earlier work [12], and in this extended journal version we have added a formalization for semantics using first-order logic with linear arithmetic, and support for generating SMT models. While we show that the scalability of SMT models surpass that of model checking, the possibility of simulating traces step-by-step is lost and queries must be expressed in first-order logic (in contrast to TCTL queries). Hence, the two semantics facilitate two complementary approaches for verification.

The goal is to create a framework in *which a user with no experience in formal methods can use to formally verify properties of a 5G service orchestration system*. The usability is our prime concern, and we wish to provide an interface which relates to the service orchestration-problem and hides as many verification details as possible.

The rest of the paper is organized as follows. Sec. 2 details the problem statement, In Sec. 3, we give an overview of UML 2.0 modeling, timed automata, the UPPAAL model checker, first-order logic, and the Z3 tool. In Sec. 4, we present our extended UML5G-SO profile in UML 2.0 allowing UML modeling of dynamic service orchestration in 5G systems. Sec. 5 presents the formal semantics of the UML model in terms of UPPAAL timed automata, while Sec. 6 presents alternative semantics in first-order logic. Thereafter, in Sec. 7, we present our methodology and the G^5 tool that allows the automated verification of system requirements, followed in Sec. 8 by a case study with experimental verification results, continued by a brief discussion of the gained insights. We compare our contribution to related work in Sec. 9, before presenting the concluding remarks and directions of future work, in Sec. 10.

2. Problem Description

This paper aims at providing a modeling and formal verification framework for dynamic 5G service orchestration. Concretely, the framework allows a 5G engineer to model an

existing service orchestration configuration (that is, VNF allocation and routing) of a network slice, and verify if the solution meets its application requirements, under various dynamic behaviors assumed as follows:

- *Dynamic network load:* We produce a dynamic network load by using a set of 5G user equipment (UE) that can request a network slice at any point of time.
- *Dynamic VNF scheduling:* We assume that hosts executes VNFs according to a given scheduling policy, with an execution time within its best- and worst-case time bounds.
- *Dynamic link utilization:* Based on the execution of the VNFs, we model the utilization of each link's bandwidth, to ensure that it is never over-utilized.

Consider an overlay network consisting of virtual machines (hosts). We assume that this network is powered by 5G, which supports end-to-end network slices. A slice consists of a number of virtual network functions (VNF), generally interconnected via a VNF Forwarding Graph, in this paper assumed to be a VNF Forwarding *Sequence*. A virtual network function is defined as a software implementation of a network function, which can be easily deployed on virtual resources [14]. We also assume that the hosts communicate via virtual links, which incurs overheads in terms of bandwidth capacity and latency. For the overlay network to serve various applications, the network slices' VNFs need to: (i) be allocated on hosts, respecting processing, memory, and storage capabilities, and (ii) be routed such that the respective VNF chaining is achieved. In our previous work [11], we have proposed a UML profile called UML5G-SO, and associated static OCL-based analysis of 5G service orchestration solutions, for checking if a given VNF allocation and routing of a slice meets the application's Quality-of-Service (QoS) requirements. However, our analysis considered only static worst-case scenarios, assuming that the system is serving a maximum number of user requests under a maximum load. This is unrealistic if one considers varying network load using hosts and links in a dynamic manner.

We assume that each slice has a certain allocation and routing defined when the system starts its operation. Our aim is to analyze if the given allocation and routing can meet the application's QoS, considering dynamic behaviors as described above. To address this, we provide a modeling and formal verification framework that combines UML-based modeling with model checking and with SMT model finding.

2.1. Running Example

To illustrate our framework we will use a small *running example* in this paper. We consider a small use case where we only have one user equipment subscribing to one slice with a latency requirement of 50 ms. The slice consists of two VNFs, each allocated to a separate host, connected by a single link. The use case is illustrated in Fig. 1, where the single slice S_1 consists of V_1 and V_2 (both of the same type). The requirement is to ensure that executing VNF V_1 on H_1, traversing the link L_1 and executing VNF V_2 on H_2 in total takes less time than the slice deadline of 50 ms.

3. Preliminaries

In this section, we introduce the types of *UML diagrams* that we use in this paper, as well as briefly overview *timed automata* and *satisfiability modulo theories* (SMT).

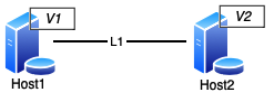


Fig. 1. Illustration of the running example

3.1. UML Diagrams

The Unified Modeling Language (UML) [8] is a modeling language that helps designers to express structural and behavioral artifacts of complex systems. In order to capture the essentials of a 5G-SO system, we specify its structure and behavior, by using our UML5G-SO profile. A *profile diagram* is used to extend existing UML models by defining *stereotypes, tagged values* and *constraints* for capturing domain-specific concepts. For a deeper understanding of UML profiles, we refer the reader to relevant literature [9,8].

We use UML *class diagrams* and *object diagrams* to represent a system's structure, and UML *statecharts* to model a system's behavior. While the former are quite straightforward, we provide a brief explanation of statecharts, as they are central to this work. UML statecharts (or UML state-machine diagrams) depict behavior via *states* and *transitions* between states. While each state simply has a name, a transition includes a *trigger*, a *guard*, and an *action*. The triggers are usually *events* (Ev), and the response actions (A) become the effects on the transitions. The *guard* (G) is a Boolean expression that has to evaluate to true in order for the transition to be fired. The UML syntax for a state transition is $Ev(parameters)[G]/A$. We chose UML statecharts for their effectiveness of capturing the behavior of individual classes (system's components).

In our work, we consider only two kinds of UML events, *time events* and *call events* [8]. We define a time event via the keyword "after", followed by an expression encoding a time value. We model call events as synchronization events that are unicast to other statecharts. The unicast communication offers handshake synchronization between two statecharts, and is blocking, i.e., the synchronization takes place only if both sender and receiver are ready to traverse their edges [24]. In addition, we consider that statecharts can be composed in parallel, defining their interactions via call events [24].

Example 1. Fig .2 shows an example of two statecharts that synchronize when executing in parallel. The initial one is the *controller* that has a state $S1$ waiting for a user equipment (UE) request. On receiving the call event, $rcvUEReq(Slice)$, it queues the respective VNFs to be executed in the host, and generates a call event (action), $sendTrigger(vnfId)$, which is unicast to the *host*. In state Sa, the host is ready to synchronize and receives the event $rcvTrigger(vnfId)$, moving to state Sb, where it executes the respective VNF and moves back to state Sa, when triggered via a time event, modeled by $after(execTime)$.

3.2. Timed Automata and UPPAAL

A Timed Automaton (TA), as used in the model-checker UPPAAL [13], is defined as a tuple, $\langle L, l_0, V, C, A, E, I\rangle$, where: L is the set of finite locations, l_0 is the initial location, V is the set of data variables, C is the set of *clocks*, $A = \Sigma \cup \tau$ is the set of *actions*, where

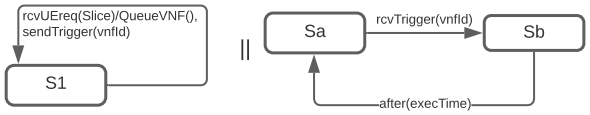


Fig. 2. Parallel Composition of UML Statecharts

Σ is the finite set of *synchronizing actions*($c!$ denotes the send action, and $c?$ the receiving action) partitioned into inputs and outputs, $\Sigma = \Sigma_i \cup \Sigma_o$, and $\tau \notin \Sigma$ denotes internal or empty actions without synchronization, $E \subseteq L \times B(C,V) \times A \times 2^C \times L$ is the set of *edges*, where $B(C,V)$ is the set of *guards* over C and V, that is, conjunctive formulas of clock constraints ($B(C)$), of the form $x \bowtie n$ or $x - y \bowtie n$, where $x, y \in C, n \in \mathbb{N}, \bowtie \in \{<, \leq, =, \geq, >\}$, and non-clock constraints over V ($B(V)$), and $I : L \longrightarrow B_{dc}(C)$ is a function that assigns *invariants* to locations, where $B_{dc}(C) \subseteq B(C)$ is the set of downward-closed clock constraints with $\bowtie \in \{<, \leq, =\}$. Invariants bound the time that can be spent in locations, hence ensuring progress of TA's execution. An edge from location l to location l' is denoted by $l \xrightarrow{g,a,r} l$, where g is the guard of the edge, a is an update action, and r is the clock reset set, that is, the clocks that are set to 0 over the edge. A location can be marked as *urgent* (marked with an U) or *committed* (marked with a C) indicating that the time cannot progress in such locations. The latter is a more restrictive, indicating that the next edge to be traversed needs to start from a *committed* location.

The semantics of TA is a *labeled transition system*. The states of the system are pairs (l, u), where $l \in L$ is the current location, and $u \in R^C_{\geq 0}$ is the clock valuation in location l. The initial state is denoted by (l_0, u_0), where $\forall x \in C,\ u_0(x) = 0$. Let $u \vDash g$ denote the clock value u that satisfies guard g. We use $u + d$ to denote the time elapse where all the clock values have increased by d, for $d \in \mathbb{R}_{\geq 0}$. There are two kinds of transitions:

(i) *Delay transitions*: $< l, u > \xrightarrow{d} < l, u + d >$ if $u \vDash I(l)$ and $(u + d') \vDash I(l)$, for $0 \leq d' \leq d$, and

(ii) *Action transitions:* $< l, u > \xrightarrow{a} < l', u' >$ if $l \xrightarrow{g,a,r} l', a \in \Sigma, u \vDash g$, clock valuation u' in the target state (l', u') is derived from u by resetting all clocks in the reset set r of the edge, such that $u' \vDash I(l')$.

The UPPAAL model checker provides exhaustive model-checking of TA models like the ones overviewed. A real-time system can be modeled as a network of TA (NTA) composed via the parallel composition operator ($||$), which allows an individual automaton to carry out internal actions, while pairs of automata can perform handshake synchronization (via $c!$ and $c?$). The locations of all automata, together with the clock valuations, define the state of an NTA. The properties to be verified by model checking on the resulting NTA are specified in a decidable subset of (Timed) Computation Tree Logic ((T)CTL) [4], and checked by the UPPAAL model checker. UPPAAL supports verification of liveness and safety properties [13]. The queries that we verify in this paper are of the form: **Invariance**, $A\Box p$, stating that p should be true in all reachable states for all paths.

3.3. Satisfiability Modulo Theories and Z3

In this paper, we use a standard approach towards first-order logic using Boolean operators ($\wedge, \vee, \Rightarrow$) as well as quantifiers ($\forall, \exists$). A formula is *satisfiable* if there is a model (an interpretation) that makes the formula true, otherwise it is called *unsatisfiable*. In this work quantifiers can be handled by enumeration as all underlying domains are finite. For an introduction and more we refer the reader to [26].

The satisfiability modulo theories (SMT) problem is posed as: given a formula, together with one (or more) background theories, is the formula satisfiable or not? In this paper we use the Z3 SMT solver [21], which can both answer such queries modulo the theory of linear arithmetic, as well as provide concrete values of a model (which is useful when analyzing answers).

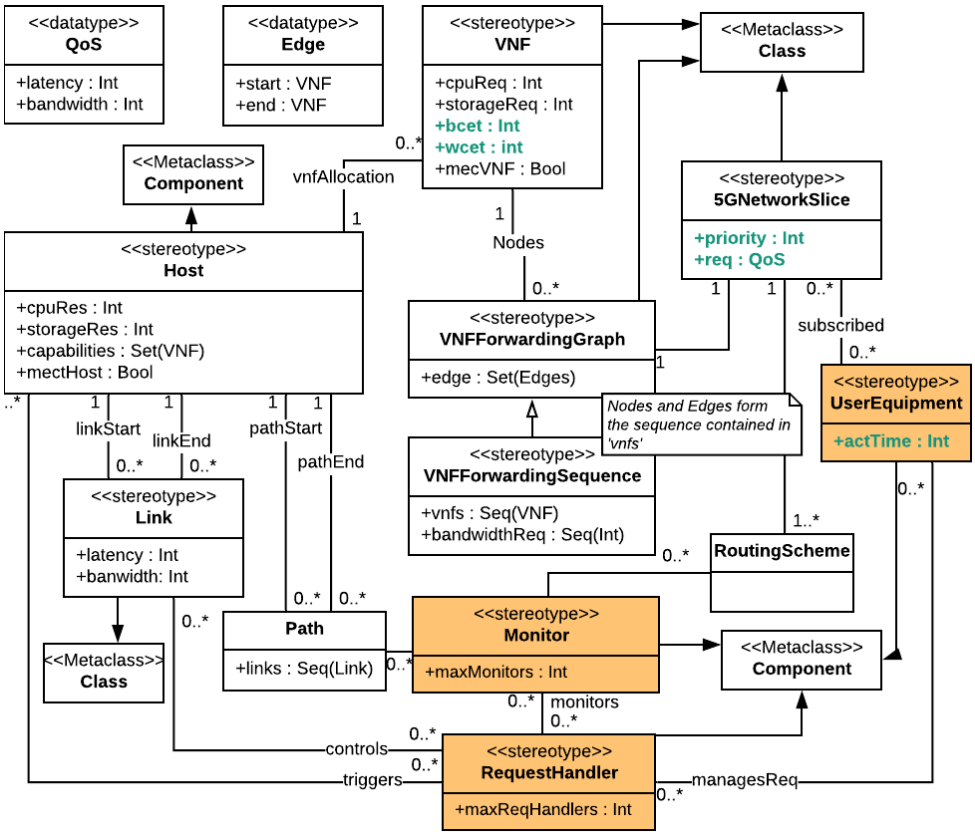


Fig. 3. The UML5G-SO Profile (Extended Version)

4. UML Modeling of Dynamic 5G Service Orchestration

In this section, we exemplify how the UML5G-SO profile can be used for modeling 5G-specific scenarios using our running example.

4.1. The Enhanced UML5G-SO Profile

The enhanced version of the UML5G-SO profile is shown in Fig. 3 (changes highlighted with colors and in bold). We define three stereotypes as follows: (i) *UserEquipment*, extending *UML Metaclass::Component*, models 5G UE. The stereotype has an attribute to specify its activation time (assumed periodic); (ii) *Request Handler*, extending *UML Metaclass::Component*, takes care of parallel UE requests and allows VNFs to be routed according to their VNF forwarding sequence; (iii) *Monitor*, extending *UML Metaclass::Class*, monitors various served requests, and is used to establish if the latter meet their requirements or not. We also replace the execution time attribute of the previous VNF stereotype with two attributes specifying best-case (BCET) and worst-case execution time (WCET).

4.2. Class Diagram Modelling of System Components

The class diagram is used to describe component types for a system. Using the stereotypes from the profile and adding attributes and functions, we introduce a class for each component type (e.g., a slice type, a kind of host). These are used by object diagrams to specify instances of systems, which can be analyzed using the methods in this paper.

Example 2. To illustrate, we show in Fig. 4 a class diagram for a running example. Some of the classes, $MonitorReq$ and $ReqControl$, are defined because they are required to model the full behavior of the system. The rest are introduced to model the components of the system, e.g., the *Ethernet* and *Server*).

In this example there is only one slice type, $5GCamera$ accessing $Slice$. The user equipment class is created by applying the $UserEquipment$ stereotype and contains the attributes $ueId$, sId and $maxReq$, which specify its id, the id of the slice it accesses, and the maximum number of requests the UE can perform, respectively. The user equipment generates the slice request event, $evSliceReq$, each time the UE gets activated, with a period equal to its activation time $actTime$. Similarly, the $Slice$ class are defined by applying the $5GNetworkSlice$ stereotype and adding an attribute for the id.

We also apply stereotypes on $ReqControl$, $MonitorReq$ and VM. The $ReqControl$ class is supplemented with an attribute for its id, and a set of functions: $initialise()$, $queueVNF()$, $consumeBW()$, $releaseBW()$, and $calcLDelay()$. When a UE requests a slice, the $initialise()$ function initializes its VNF forwarding sequence and its routing scheme (note that we assume that all VNFs are allocated). The $ReqControl$ also takes care of queuing the respective VNFs to their hosts via the $queueVNF()$ function. The functions $consumeBW()$, $releaseBW()$, $calcLDelay()$ consume the link bandwidth while routing through it, release it after its usage, and calculate the respective delays in routing across the links. The VM class has one attribute, id and two functions, $scheduleVNF()$ and $dequeueVNF()$, the former encoding the scheduling algorithm, and the latter being responsible for dequeueing the next VNF to be executed.

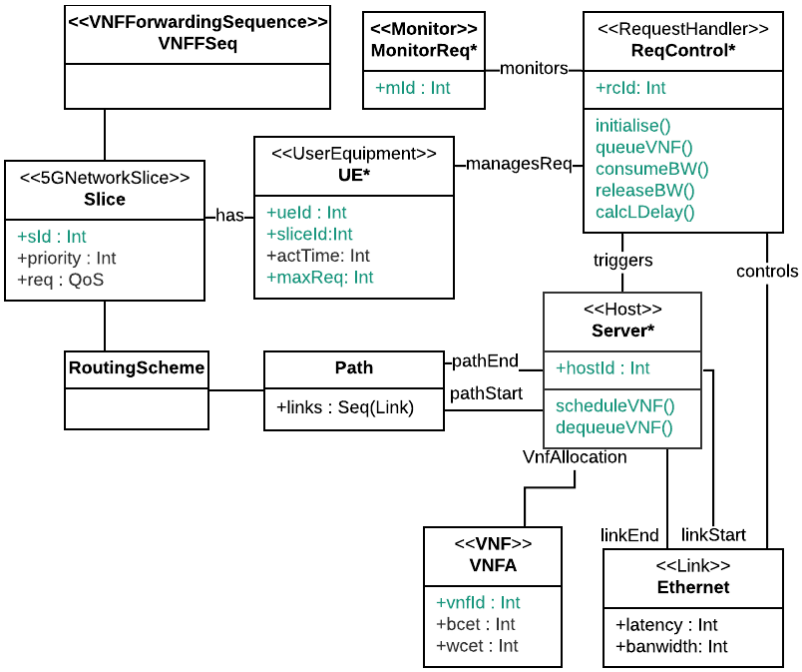


Fig. 4. Class Diagram for our running example

Once the class diagram description of our system is formulated, we encode the behavior of each active class by using a restricted form of UML statecharts. In our examples, the classes that possess behavior are marked with an asterisk, e.g., see Fig. 4.

4.3. Object Diagram Description of System Instance

Given a class diagram, describing each component in the system, it is possible to depict a system instance by means of an *object diagram*. In the object diagram, the classes are instantiated and their attributes assigned values, according to the corresponding use case.

Example 3. In Fig. 5 an object diagram shows the running example system instance.

4.4. Restricted Statechart-based Behavioral Description of our 5G-SO System

In this section, we discuss the behavioral description of our system using UML statecharts restricted to fit our needs. We define the restricted form of statecharts, as follows:

Definition 1. *A* restricted statechart *(RSC) is a UML statechart obeying the following:*

- All states are simple (without hierarchies) and without associated execution history.
- The states can either be the usual simple states of UML statecharts, hereby called "active" states, or pseudostates that represent the initial and final states, only.

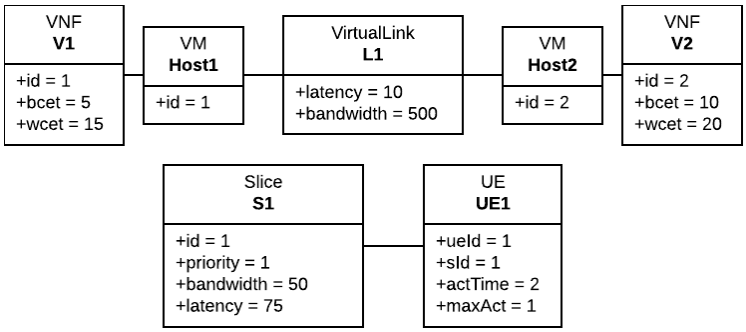


Fig. 5. Object Diagram for our running example

– The transitions follow usual UML syntax, $Ev(params)[G]/A$, where $Ev(params)$ are UML call or time events (Sec. 3), G are Boolean conditions evaluated over system variables, and A include variable assignments and other user-defined functions.

A transition $Ev(params)[G]/A$ is triggered when the event $Ev(params)$ occurs and G is true, or as soon as G is true in case of no triggering event. Moreover, the transitions from psuedostates to active states or vice-versa are considered instantaneous if there are no triggering events or guards enabling them. Different statecharts synchronize via unicast or broadcast synchronizations, defined via a parallel composition of the independent restricted statecharts. In addition, the statecharts follow the run-to-completion execution semantics [5], i.e., a statechart completes processing each event before it can start processing the next one. The statecharts in Fig. 6 are examples of restricted statecharts, as they

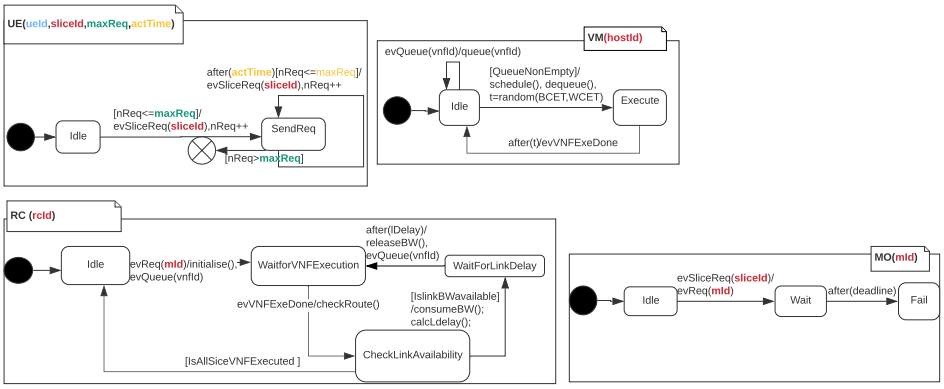


Fig. 6. Restricted statechart representation of the case-study-specific behavior

obey all restrictions in Def. 1. Note that the statecharts UE, RC, VM and MO, correspond to the active classes UserEquipment, ReqControl, VM, MonitorReq, respectively.

Pattern-Based Behavioral Representation To model the behavior of the components, we apply a pattern-based representation. We begin with an auxiliary definition.

Definition 2. *An* assignment *for a set of variables V, is a mapping from variables to values such that each variable is assigned one value. The set of possible assignments for V is denoted as $\mathcal{A}_V$.*

A *restricted statechart pattern, RSC(para)* is a reusable RSC structure used to define repetitive behaviors, and is represented as the following tuple:

$$RSC(para) = \langle S_p, s_{ip}, V_p, Ev_p, F_p, L_p, \rightarrow_p \rangle : para, \tag{1}$$

where:

- $para$ refers to the list of parameters instantiated with values when the pattern is used,
- $S_p = S_{act} \cup S_{psu} = \{s_1, .., s_n\}$, is the set of states, i.e., the set of active states (S_{act}), and the set of pseudostates (S_{psu}), that is, initial and final states,
- $s_{ip} \in S$ is the initial pseudostate,
- V_p is the set of system variables,
- $Ev_p = Ev_t \cup Ev_c$ is the set of events, where Ev_t is the set of time events and Ev_c is the set of call events. Further, $Ev_c = Ev_{trig} \cup Ev_{act}$, where Ev_{trig} is the set of events triggering the transition, and Ev_{act} is the set of generated events,
- F_p is the set of user-defined functions, where for each $f \in F_p, f : \mathcal{A}_{V_p} \mapsto \mathcal{A}_{V_p}$, which maps each variable assignment to a (possibly identical) variable assignment,
- $L_p \subseteq Ev_p \times G_p \times \mathcal{A}_{V_p} \times F_p \times Ev_p$ is the set of labels, where for each label $(ev_t, g, a, f, ev_g) \in L_p$:
  - ev_t is a *triggering* event,
  - g is a guard (i.e., G_p is the set of Boolean expressions over V_p),
  - a is an assignment,
  - f is a user-defined function,
  - ev_g is a *generated* event.
- $\rightarrow_p \subseteq S_p \times L_p \times S_p$, denoted by $s \xrightarrow{L_p}_p s', \{s, s'\} \in S_p$.

Example: In Fig. 6 we present the four patterns corresponding to the active classes in Fig. 10. We exemplify the restricted statechart $RSC_{UE}(ueId, sliceId, maxReq, actTime)$, describing the behavior of UserEquipment:

- $S_p = S_{psu} \cup S_{act}$, where $S_{psu} = \{Initial, Final\}$ and $S_{act} = \{Idle, SendReq\}$,
- $s_{ip} = Initial$,
- $V_p = \{nReq\}$, $para = \{ueId, sliceId, maxReq, actTime\}$,
- $Ev_p = \{evSliceReq(sliceId), after(actTime)\}$,
- $F_p = \{\emptyset\}$,
- $L_p = \{(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset), (\emptyset, nReq \leq maxReq, nReq{+}{+}, \emptyset, evSliceReq(sliceId))$, $(after(actTime), nReq \leq maxReq, nReq{+}{+}, \emptyset,$ $evSliceReq(sliceId)), (\emptyset, nReq > maxReq, \emptyset, \emptyset, \emptyset)\}$,
- $\rightarrow = \{Initial \rightarrow Idle, Idle \xrightarrow{\emptyset, nReq \leq maxReq, nReq{+}{+}, \emptyset, evSliceReq(sliceId)} SendReq,$ $SendReq \xrightarrow{after(actTime), nReq \leq maxReq, nReq{+}{+}, \emptyset, evSliceReq(sliceId)} SendReq,$ $SendReq \xrightarrow{\emptyset, nReq > maxReq, \emptyset, \emptyset, \emptyset} Final\}$

Once the patterns are defined, the patterns can be instantiated by assigning $p \in para$ with actual values. We denote by "$p = v$" the assignment of parameter p with value v, so the instantiated RSC(para) is:

$$RSC_i(v_1, v_2, \ldots) = \langle S_{pi}, s_{ip_i}, V_{pi}, Ev_{pi}, F_{pi}, L_{pi}, \rightarrow_{pi} \rangle : (p_1 = v_1, \ldots) \quad (2)$$

Example 4. Consider the RSC_{UE} pattern. An example of a parameter assignment is:

$$para = \{(ueID, 1), (sliceID, 1), (maxReq, 1000), (actTime, 5)\} \quad (3)$$

The resulting statechart is shown in Fig. 7.

User-Defined Functions. Some of the classes contain user-defined functions, expressing some behavioral aspect of the system. These must be formalized both for the TA model as well as the SMT model. In general, for the TA model we consider user-defined functions to be given as C-like code, as supported by UPPAAL. For the SMT model instead, the effect of executing the function must be formulated in first-order logic. A list of user-defined functions used in our model is found in Table 1.

Example 5. Consider the function $schedule()$ from our running example. Its C-like code is shown in Listing 1.1. A SMT-formalization of its effect is presented later (see Eq. 42).

Listing 1.1. Function for priority-based scheduling

```
int i, j, tmp;
for (i = 0; i<VM.MQ.length; i++) {
    for (j = 1; j<VM.MQ.length; j++) {
        if (VM.MQ[i].Prio > VM.MQ[j].Prio) {
            tmp = VM.MQ[i]
            VM.MQ[i] = VM.MQ[j]
            VM.MQ[j] = temp
        }
    }
}
```

Table 1. User-defined functions

Function	Purpose
$initialise()$	Initializes the slice VNF forwarding sequence, VNF allocation and routing, upon receiving a particular slice request from the UE.
$queue(VNFId)$	Queues the VNF that are ready to be executed into the message queue of the VM where it is allocated.
$schedule()$	Sorts the VNFs in the queue according to priorities of the VNFs such that the head of the queue has the highest priority.
$dequeue()$	Removes the VNF that is executed from the message queue.
$calcLdelay()$	Calculates the sum of latencies over all the links involved a particular path between consecutive VNFs in VNF forwarding sequence.
$consumeBW()$	The required bandwidth of a slice is consumed from the available link bandwidth.
$releaseBW()$	Once a slice has finished utilizing a link, release the respective bandwidth.

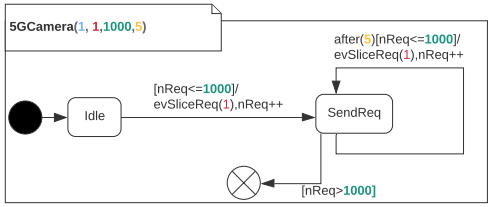


Fig. 7. Example of an instantiated RSC_{UE}

Parallel Composition of RSC(para). A 5G system comprises a number of interacting components, each behaviorally defined by an RSC pattern. Hence, to evaluate how a system is executed, we must compose in parallel the involved RSC patterns. Given a system consisting of n RSC(para), we denote their parallel composition by:

$$RSC_1(para_1) \,||\, RSC_2(para_2) \,||\, \cdots \,||\, RSC_n(para_n) \tag{4}$$

Example 6. The UML statechart-based behavioral model of our running example is defined as the parallel composition of each system component's $RSC(para)$:

$$RSC_{UE1} \,||\, RSC_{MO1} \,||\, RSC_{RC1} \,||\, RSC_{Host1} \,||\, RSC_{Host2} \tag{5}$$

where RSC_{UE1} is the instantiations of the pattern $RSC_{UE}(para)$ for $UE1$ in the system. Similarly, we define instantiations of $RSC_{MO}(para)$, $RSC_{RC}(para)$ and $RSC_{VM}(para)$, representing the UML statechart patterns defined for the active classes corresponding to MonitorReq, ReqControl, and VirtualMachine, respectively.

Request Controllers and Monitors. There are two active classes, request controllers and monitors, which play a special role in the model. These are simulation artifacts that facilitate keeping track of which VNF is currently executed where, and what has to be done next. The model only requires that sufficient instances of these two classes are provided.

5. TA Semantics of RSC Patterns

To formally verify properties of a 5G-SO system, we need to assign formal semantics to the RSC model of component behavior, corresponding to RSC informal semantics. We define semantics in terms of TA that behave according to timed transition semantics (see Sec. 3.2). For each restricted statechart pattern (see Sec. 4.4), we have a corresponding TA pattern. The semantics of the system is defined as the underlying timed transition system of the network of all TA obtained by instantiating the corresponding TA patterns.

5.1. UPPAAL TA pattern

Using the definition of $RSC(para)$ in Sec. 4.4, and the definition of TA (see Sec. 3.2), we define a semantic encoding of the $RSC(para)$ components, respectively, in terms of

$TA(para)$. Like $RSC(para)$, a $TA(para)$ is also defined as a reusable TA structure, for reoccurring behavior, as follows:

$$TA(para) = \langle L_p, l_{0p}, V_p, C_p, A_p, E_p, I_p \rangle : para, \tag{6}$$

where:

- $para$ is a list of parameters that gets instantiated with values when the pattern is used,
- $L_p = \bigcup_{i=1}^{n} La_i \cup \bigcup_{i=1}^{n} Lc_i \cup \bigcup_{i=1}^{n} Le_i$ where La_i correspond to S_{act} in $RSC(para)$, Lc_i is the set of committed locations introduced to handle the simultaneous synchronizations involving the trigger (ev_t) and effect actions (ev_g) occurring in a transition, Le_i is the set of error locations introduced to capture errors of message queue being full, not enough request controllers/monitors to handle the request, or deadline violation,
- $l_{0p} = Idle$ is the initial location,
- V_p is the set of variables defined in the corresponding $RSC(para)$, and other local variables that are used to model the parameter passing (by making a local copy of the parameter passed), and other variables if needed to define the error conditions that lead to Le_i,
- C_p is the set of clock variables that measure the time elapsed for the corresponding Ev_t,
- $A_p = A_{sync} \cup A_{asg} \cup A_{udf}$, where A_{sync} corresponds to the synchronizing events in Ev_c and other urgent synchronizations ($execute?$) defined to trigger a transition as soon as the guard evaluates to true, A_{asg} is the actions involving assignment of variables and clock resets, A_{udf} is the set of user-defined functions in $RSC(para)$,
- $E_p = E_e \cup E_a$, are the TA edges, where E_e refers to the set of edges defined by $\rightarrow_p$ in $RSC(para)$, decorated with L_p in $RSC(para)$ and other guards defined over the clock variables along the transitions enabled with Ev_t events; E_a are the edges defined to connect (La_i and Lc_i) [3] and (La_i and Le_i) [4],
- I_p is the set of invariants that are defined over La_i that generate Ev_t.

Example 7. As an example, we present the TA pattern, $TA_{UE}(para)$, depicted in Fig. 8, which corresponds to $RSC_{UE}(para)$. $TA_{UE}(para)$ is defined as follows:

$$\begin{gathered} TA_{UE}(ueId, sId, maxReq, actT) = \\ \langle L_{ue}, l0_{ue}, V_{ue}, C_{ue}, A_{ue}, E_{ue}, I_{ue} \rangle : (ueId, sId, maxReq, actT), \end{gathered} \tag{7}$$

where:

- $L_{ue} = \{Idle, Start, Select, NoFreeMonitor, NoFreeContr\}$,
- $l0_{ue} = Idle$,
- $V_{ue} = \{ueId, sId, maxReq, actT, nReq, reqSId, avContrs, avMonitors\}$, such that $ueId$ represents the id of the UE, and sId represents the id of the slice it requests, $maxReq$ is the maximum number of slice requests that a UE can make, $nReq$ keeps track of the number of UE requests, $reqSId$ is a variable that copies the sId to reflect the passing of $sliceId$ parameter in RSC_{UE}, $avContrs$, and $avMonitors$, to capture errors if there are no available controllers or monitors to handle the requests,

[3] If both the trigger and effect actions occur simultaneously in a transition, the latter will be transformed to two edges with a committed location in between, in which the former is synchronized with the trigger action and the latter is synchronized with the effect action.

[4] The edge connecting La and Le is decorated with guards that are true when meeting the error conditions.

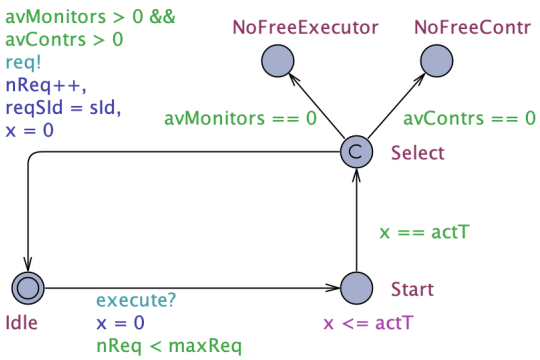


Fig. 8. UPPAAL TA pattern for UserEquipment

- $C_{ue} = \{x\}$ is the one-clock set containing the clock that models the activation time,
- $A_{ue} = \{req!, execute?\} \cup \{reqSId = sId, nReq{+}{+}, x = 0\}$, where A comprises the set of synchronization channels associated with generating a slice request ($req!$) and enforcing UE to not idle ($execute?$), the corresponding variable assignments, and reset action on clock x,
- $E_{ue} = \{Idle \xrightarrow{execute?, nReq<maxReq, x=0} Start, Start \xrightarrow{x==actT} Select,$
  $Select \xrightarrow{avContrs==0} NoFreeContr, Select \xrightarrow{avMonitors==0} NoFreeMonitor,$
  $Select \xrightarrow{req!, (avContrs>0 \&\& avMonitors>0), nReq{+}{+}, reqSId=sId, x=0} Idle\}$,
- $I_{ue} : Start \rightarrow (x \leq actT)$.

Using a similar approach, we generate $TA_{VM}(para)$, $TA_{RC}(para)$, and $TA_{MO}(para)$ patterns corresponding to their RSC counterparts. The instantiation of $TA(para)$ assigns the parameters in $para$ with actual values. Using "$p = v$" to denote the assignment of parameter p with value v, we define the instantiated pattern as:

$$TA_i(v_1, v_2 \ldots) ::= \langle L_{pi}, l_{0pi}, V_{pi}, C_{pi}, A_{pi}, E_{pi}, I_{pi} \rangle : (p_1 = v_1, p_2 = v_2, ...) \quad (8)$$

Example 8. Consider TA_{UE} introduced above. An example parameter assignment is:

$$para = (ueID = 1, sId = 1, maxReq = 1000, actT = 5) \quad (9)$$

Parallel Composition of TA(para). Given a system consisting of n $TA(para)$, that is, $TA_1(para_1), TA_2(para_2), \ldots, TA_n(para_n)$, their parallel composition is denoted as:

$$NTA = TA_1(para_1) \parallel TA_2(para_2) \parallel \ldots \parallel TA_n(para_n) \quad (10)$$

Example 9. Consider the running example. Assuming that there are only one parallel request, the behavior is formalized as a network of timed automata:

$$TA_{UE1} \parallel TA_{MO1} \parallel TA_{RC1} \parallel TA_{Host1} \parallel TA_{Host2} \quad (11)$$

5.2. Queries

Given an instantiated UPPAAL model, we can formulate (T)CTL queries to check properties of the system. For example, the following query is satisfied if there is no scenario where a deadline is violated (i.e., checking that the monitor does not register a missed deadline):

$$A\,\Box\ not\ MO.MissedDeadline \tag{12}$$

Note that the bandwidth requirement is automatically met upon satisfaction of the latency requirement, since we construct the models in such a way that a UE can complete its request only if the necessary bandwidth is available.

6. First-Order Semantics

In the previous section, we assigned semantics to the 5G system in the form of timed automata. As an alternative, in this section we present a formalization of semantics in first-order logic with linear arithmetic. We show how the behavior depicted in the statechart in Fig. 6 can be represented. As mentioned, in this paper we assume that we already have a sound allocation and routing; nevertheless, it is also straightforward to encode this in logic, see our previous work where a formalization was presented in OCL [11].

As a starting point, we consider a given 5G-SO system described by an object diagram. We present a set of formulas that model the intended semantics, and then we add constraints for the desired property to be checked, e.g., "deadlines are not violated".

Example 10. Consider the network given in Fig. 5, consisting of two hosts and one link. On this network, we wish to serve one slice consisting of two VNFs. All requests for slices are generated by a single user equipment.

We start with a formalization of the objects found in an object diagram.

Definition 3. *We define a 5G-SO system as the tuple:* $5GSO = (H, L, S, V, U)$*, where* H *is the set of* Hosts*,* L *is the set of* Links*,* S *is the set of* Slices*,* V *is the set of* VNFs *and* U *is the set of* User Equipment*.*

Each object has a number of properties, e.g., a slice has a *deadline* and a *VNF forwarding sequence*, which are listed in Table 2. In formulas, we use $prop(o)$ to denote the value of the prop assigned to object o. For example, $deadline(s)$ denotes the *deadline* assigned to slice s. All relevant properties and corresponding functions are shown in Table 2.

Example 11. Our running example has $H = \{H_1, H_2\}$, $L = \{L_1\}$, $S = \{S_1\}$, $V = \{V_1, V_2\}$ and $U = \{U_1\}$. Properties from the object diagram are defined accordingly, e.g., $bw(L_1) = 500$, $deadline(S_1) = 75$, $vnffseq(S_1) = [V_1, V_2]$, and $subslice(U_1) = S_1$.

We introduce two core concepts for our model: *requests* and *schedules*. A request describes, for a given slice, what VNFs needs to be executed, when and for how long. Since our model assumes an allocation of VNFs to hosts, and given best-case and worst-case execution times, we can represent a request as a sequence of host and durations.

Table 2. Properties of a 5G system

Object type	Description	Type	Function name
Link	Bandwidth capacity of the link	Integer	$capacity$
Link	Latency of the link	Integer	$latency$
Slice	Latency requirement of slice	Integer	$deadline$
Slice	Constituting sequence of VNFs	Sequence of VNF types	$vnffseq$
VNF	Best-Case Execution Time	Integer	$bcet$
VNF	Worst-Case Execution Time	Integer	$wcet$
User Equipment	Subscribed slice	Slice	$subslice$

Definition 4. *A* request R *is defined as a constraint-triple* $(R_{start}, R_{deadline}, R_{priority})$ *and a sequence of host-duration triples:*

$$R_{exec} = [(h_1, dmin_1, dmax_1), \ldots, (h_n, dmin_n, dmax_n)], \tag{13}$$

where $_{start}, R_{deadline}, R_{priority} \in \mathbb{N}$*,* $h_i \in H$ *and* $dmin_i, dmax_i \in \mathbb{Z}^+$.

The constraint-triple gives the starting time of a request ((R_{start}), its (relative) deadline ($R_{deadline}$), and its priority ($R_{priority}$). The host-duration triple consists of a host, a minimum and a maximum execution time. Intuitively, $(h_1, 5, 15)$ means the request requires host h_1 to execute between 5 and 15 time units. This information is derived from the best- and worst-case execution time of a VNF, as well as to which host it is allocated.

Example 12. In our running example, considering that the user equipment has only two activation, it generates two requests R^1 and R^2 according to S_1, which consists of VNFs $[V_1, V_2]$, and thus: $R^1_{exec} = e$, $R_{start} = 0$, $R_{deadline} = 75$ and $R_{priority} = 1$, and $R^2_{exec} = e$, $R_{start} = 50$, $R_{deadline} = 125$ and $R_{priority} = 1$, where $e = [(H_1, 5, 15), (H_2, 10, 20)]$.

Our second concept describes how a request can be fulfilled. A *schedule* is a sequence of *actions*. Each action is either a computation-based *host*-action, a communication-based *link*-action or one of the *meta*-actions initialize and complete. Each action has a time-tuple that describes when the corresponding is executed.

Definition 5. *A* schedule *for a request* R*, denoted* S_R*, is a sequence of actions, each action being one of the following:*

- $\mathrm{I}(T)$ *- This signifies the start of the request*
- $\mathrm{H}^p_h(T)$ *- This signifies the host* h *computing with priority* p
- $\mathrm{L}^w_l(T)$ *- This signifies link* l *communicating with and bandwidth of* w
- $\mathrm{C}(T)$ *- This signifies the completion of the request*

where $T = (t_0, t_B, t_E)$ *is the timing information for the action. It is interpreted as follows:* t_0 *is the first possible time the action can be performed;* t_B *is the time when the action is* actually *started, and* t_E *is the time when the action is finished. It is required that the schedule begins with an initialize-action and ends with a complete-action.*

Note that for the meta-actions all time-values will be equal, so we can write $\mathrm{I}(0,0,0)$ as $\mathrm{I}(0)$ (and similar for the completion action).

In contrast to Sec. 5 where we have described the semantics from a behavioral point of view, here we describe the possible resulting execution traces. For a particular request, we know which hosts need to execute, for how long, with what priority, as well as what links should be traversed in between with what delay. Therefore, we can for each request introduce a schedule *fulfilling* that request, i.e., it performs the necessary actions.

Example 13. The following schedule fulfills the request R^1 in Example 12:

$$[\mathrm{I}(0), \mathrm{H}_1^1(0,0,10), \mathrm{L}_1^{50}(10,10,20), \mathrm{H}_2^1(20,30,40), \mathrm{C}(40)] \tag{14}$$

The schedule is initialized at time zero, then H_1 computes for ten time units, L_1 communicates for ten time units, and finally H_2 computes for ten time units. Note that the second host-action has a delay between when it was available and when it was started.

For a set of request, we can create a set of schedules fulfilling those requests, but without the timing information. We do not go into detail, but in principle, for each request, a schedule is obtained by converting each host-duration triple to its corresponding host-action, adding link-actions in between (for the corresponding links), and finally prepending an initialize-action and appending a complete-action. The resulting schedule contains no timing information, these are the variables of the model (i.e., the values sought after).

Example 14. Consider our running example. Since the user equipment generates two requests, we will need to create two schedules as follows:

$$[\mathrm{I}(r_{1_b}), \mathrm{H}_1^1(h_{0_0}, h_{0_b}, h_{0_e}), \mathrm{L}_1^{50}(l_{0_0}, l_{0_b}, l_{0_e}), \mathrm{H}_2^1(h_{1_0}, h_{1_b}, h_{1_e}), \mathrm{C}(r_{1_e})] \tag{15}$$

$$[\mathrm{I}(r_{2_b}), \mathrm{H}_1^1(h_{2_0}, h_{2_b}, h_{2_e}), \mathrm{L}_1^{50}(l_{1_0}, l_{1_b}, l_{1_e}), \mathrm{H}_2^1(h_{3_0}, h_{3_b}, h_{3_e}), \mathrm{C}(r_{2_e})] \tag{16}$$

The variables (e.g., $h_{0_0}, l_{1_b}, r_{2_e}$) represents timing information, which is to be determined.

6.1. Constraints

As seen above, given a set of requests, we can introduce a set of schedules fulfilling those requests. First, we present constrains requiring these schedules to be sound (**Sound schedules**). Next, we need to ensure that the schedules are compatible, that is, a host is not used by two schedules at the same time (**No host overlap**), and the sum of bandwidths used on a single link at any point in time does not exceed the link capacity (**No link over-utilization**). We also need to ensure that the scheduling policy is enforced, in this case-study being "first-come first-served" priority-based scheduling (**Scheduling**). Note that we also need to enforce this to routing over links (**Routing**), where priority is ignored.

We begin with some useful predicates and sets, which will be used in the remainder of this section. Furthermore, we assume that $\mathcal{A}$ is a set containing all action-time pairs.

Predicates:

$$\mathit{disjoint}(T^1, T^2) := T_E^1 \leq T_B^2 \vee T_E^2 \leq T_B^1 \tag{17}$$

$$\mathit{in}(t, T) := T_B \leq t < T_E \tag{18}$$

$$\mathit{used}(l, t, \mathrm{L}_{l'}^{bw}(T)) := \mathit{in}(t, T) \wedge l = l' \tag{19}$$

The $disjoint$ predicate is true whenever the actual executing time of the two time-tuples T^1 and T^2 does not overlap; in is true when t is in the actual executing time of T; and finally $used$ is true if link l is used by link-action $\mathrm{L}_l^{bw}(T)$ at time t.

Sets:

$$hostActions(h) := \{\mathrm{H}_{h'}^{p}(T) \mid \mathrm{H}_{h'}^{p}(T) \in \mathcal{A}, h = h'\} \tag{20}$$

$$linkActions(l) := \{\mathrm{L}_{l'}^{bw}(T) \mid \mathrm{L}_{l'}^{bw}(T) \in \mathcal{A}, l = l'\} \tag{21}$$

$$hostEnds(h) := \{t \mid \mathrm{H}_{h'}^{p}(T) \in \mathcal{A}, h = h' \wedge t = T_E\} \tag{22}$$

$$linkStarts(l) := \{t \mid \mathrm{L}_{l'}^{bw}(T) \in \mathcal{A}, l = l' \wedge t = T_B\} \tag{23}$$

$$linkEnds(l) := \{t \mid \mathrm{L}_{l'}^{bw}(T) \in \mathcal{A}, l = l' \wedge t = T_E\} \tag{24}$$

The set $hostActions(h)$ ($linkActions(l)$) contains all host-actions (link-actions) for host h (link l); $hostEnds(h)$ is the set of all times at which a host-action for host h ends executing, and finally $linkStarts(l)$ and $linkEnds(l)$ are defined analogously.

Sound schedules. First, we require that each schedule S_R begins at its start time, thus for each initialize-action $I(T)$:

$$T_0 = R_{start} \tag{25}$$

Next, no action can actually begin executing before it is ready:

$$T_0 \leq T_B \leq T_E \tag{26}$$

The duration of host-actions must be in the interval of the d_{min} and d_{max} of the corresponding action in the request, thus for all $\mathrm{H}_h^p(T)$:

$$d_{min} \leq T_E - T_B \leq d_{max} \tag{27}$$

Durations of link-actions must be equal to the latency of the link, thus for all $\mathrm{L}_l^{bw}(T)$:

$$T_E - T_B = latency(l) \tag{28}$$

We require that for all actions in a schedule no action can begin before the preceding one has ended. Hence, for each pair of successive actions $(a^1, T^1), (a^2, T^2)$ in a schedule:

$$T_0^2 = T_E^1 \tag{29}$$

Example 15. For our running example, the following constraints would be introduced:

$$r_{1_b} = 0, r_{1_b} = h_{0_0}, h_{0_e} = l_{0_0}, l_{0_e} = h_{1_0}, h_{1_e} = r_{1_e} \tag{30}$$

$$r_{2_b} = 75, r_{2_b} = h_{2_0}, h_{2_e} = l_{1_0}, l_{1_e} = h_{3_0}, h_{3_e} = r_{2_e} \tag{31}$$

$$h_{0_0} \leq h_{0_b} \leq h_{0_e}, l_{0_0} \leq l_{0_b} \leq l_{0_e}, h_{1_0} \leq h_{1_b} \leq h_{1_e} \tag{32}$$

$$h_{2_0} \leq h_{2_b} \leq h_{2_e}, l_{1_0} \leq l_{1_b} \leq l_{1_e}, h_{3_0} \leq h_{3_b} \leq h_{3_e} \tag{33}$$

$$5 \leq h_{0_e} - h_{0_b} \leq 15, l_{0_e} - l_{0_b} = 10, 10 \leq h_{1_e} - h_{1_b} \leq 20 \tag{34}$$

$$5 \leq h_{2_e} - h_{2_b} \leq 15, l_{1_e} - l_{1_b} = 10, 10 \leq h_{3_e} - h_{3_b} \leq 20 \tag{35}$$

No host overlap. We must ensure that no two host-actions of the same host overlap.

$$\forall\, \mathrm{H}^{p}_{h_1}(T^1), \mathrm{H}^{p}_{h_2}(T^2) \in \mathcal{A} \quad h_1 = h_2 \Rightarrow disjoint(T^1, T^2) \tag{36}$$

Example 16. For our running example, the following constraints (simplified by removing antecedent and replacing $disjoint$ with its definition) would be introduced (we only write those where the hosts are the same, as the others are trivially satisfied):

$$h_{0_e} \leq h_{2_b} \vee h_{2_e} \leq h_{0_b} \tag{37}$$

$$h_{1_e} \leq h_{3_b} \vee h_{3_e} \leq h_{1_b} \tag{38}$$

No link over-utilization. We must ensure that the sum of the bandwidth usage of active actions at any given point is less than the capacity of the link. However, note that if over-utilization would occur, it would at least occur when a new link-action is started:

$$\forall l \in L \quad \forall t \in linkStarts(l) \left(\sum_{\substack{\mathrm{L}^{bw}_{l}(T) \in \mathcal{A} \\ used(l,t,\mathrm{L}^{bw}_{l}(T))}} bw \right) \leq capacity(l) \tag{39}$$

Example 17. For our running example there are two points in time when a link action begins (l_{0_b}, l_{1_b}), so these two time-points must be checked. For clarity, let $load(cond, bw)$ be a function which returns bw if $cond$ is true otherwise 0. Using this, we can expand the summation sign and explicitly enumerate the two time-points:

$$load(in(l_{0_b}, T(l_{0_0}, l_{0_b}, l_{0_e})), 50) + load(in(l_{0_b}, T(l_{1_0}, l_{1_b}, l_{1_e})), 50) \leq 500 \tag{40}$$

$$load(in(l_{1_b}, T(l_{0_0}, l_{0_b}, l_{0_e})), 50) + load(in(l_{1_b}, T(l_{1_0}, l_{1_b}, l_{1_e})), 50) \leq 500 \tag{41}$$

Scheduling. As mentioned, we present a formalization of first-come first-serve scheduling of VNFs. We can ensure this by enforcing that if a host-action is not started immediately, during all time-steps between T_0 and T_B there is another computing host-action, which was ready earlier or at the same time with higher or equal priority. Note that we do not need to check all time-steps, but it is enough to check that indeed at T_0 this is the case, and then all time-steps where a host-action (of the same host) ends (since this is the only time when the host could become idle):

$$\begin{gathered} \forall(\mathrm{H}^{p^1}_{h}(T^1)) \in \mathcal{A} \quad \forall t \in hostEnds(h) \cup \{T^1_0\} \\ T^1_0 \leq t < T^1_B \Rightarrow \\ \exists \mathrm{H}^{p^2}_{h}(T^2) \in hostActions(h) \quad \mathrm{H}^{p^1}_{h}(T^1) \neq \mathrm{H}^{p^2}_{h}(T^2) \wedge in(t, T^2) \wedge T^2_0 \leq T^1_0 \wedge p^1 \geq p^2 \end{gathered} \tag{42}$$

Example 18. In our running example, there are in total four host-actions (two per request), and for each host-action there is one other host-action of the same host, thus for each host-action there are two time-points to check, the first possible time for the action, and the end-time for the other host-action. For each check, we can replace the existential

quantifier with its enumeration, which in this case is only one action. We show this for the first host-action (the constraints for the other host-actions are analogous):

$$h_{0_0} \leq h_{0_0} < h_{0_b} \Rightarrow in(h_{0_0}, (h_{2_0}, h_{2_b}, h_{2_e})) \wedge h_{2_0} \leq h_{0_0} \wedge 1 \geq 1 \tag{43}$$

$$h_{0_0} \leq h_{2_e} < h_{0_b} \Rightarrow in(h_{2_e}, (h_{2_0}, h_{2_b}, h_{2_e})) \wedge h_{2_0} \leq h_{0_0} \wedge 1 \geq 1 \tag{44}$$

Routing. Routing must also be enforced, and we use the same policy as for scheduling, "first-come first-serve". We can ensure this by enforcing that if a link-action is not started immediately, during all time-steps between T_0 and T_B there are other communicating link-actions, which were ready earlier or at the same time, consuming bandwidth such that the link-action can not be accommodated. Note that we do not need to check all time-steps, but it is enough to check that indeed at T_0 this is the case, and then all time-steps where a link-action (of the same link) ends (since this is the only time when the link could get sufficient free bandwidth):

$$\forall\, \mathrm{L}_l^{bw}(T^1) \in \mathcal{A} \quad \forall t \in \mathit{linkEnds}(l)$$

$$in(t, T^1) \Rightarrow \left(\sum_{\substack{\mathrm{L}_l^{bw'}(T^2) \in \mathcal{A}, \mathrm{L}_l^{bw}(T^1) \neq \mathrm{L}_l^{bw'}(T^2) \\ used(l,t,\mathrm{L}_l^{bw'}(T^2))}} bw' \right) > capacity(l) - bw \tag{45}$$

Example 19. The routing for our running example is similar to the scheduling constraint, except that the existence of another host-action, is replaced by summing the bandwidth usage of all active link-actions (and we use the same function *load* as in no over-utilization constraint). We show for the first link-action, the second case is symmetrical:

$$l_{0_0} \leq l_{0_0} < l_{0_b} \Rightarrow load(in(l_{0_0}, T(l_{1_0}, l_{1_b}, l_{1_e})), 50) \leq 500 \tag{46}$$

$$l_{0_0} \leq l_{1_e} < l_{0_b} \Rightarrow load(in(l_{1_e}, T(l_{1_0}, l_{1_b}, l_{1_e})), 50) \leq 500 \tag{47}$$

6.2. Queries

If all constraints above are satisfied, the underlying set of schedules are compatible. Now, we can add additional constraints corresponding to queries, that is, properties to check.

Deadline violation. To check if any single schedule violates its deadline we add:

$$\bigvee_{S_R \in \mathcal{SC}} T_E^2 - T_B^1 > R_{deadline}, \tag{48}$$

where $\mathcal{SC}$ is the set of all schedules, and $\mathrm{I}(T^1), \mathrm{C}(T^2)$ are the initialize- and complete-action of each schedule S_R, and $R_{deadline}$ is the deadline of the corresponding request.

Example 20. For our running example, the deadline query would be:

$$r_{1_e} - r_{1_b} > 75 \vee r_{2_e} - r_{2_b} > 75 \tag{49}$$

6.3. SMT solving

To practically use these semantics to determine a property for a given 5G-SO system, we create one request for each activation of the user equipment. These are then encoded as schedules, with corresponding constraints, in Z3, together with the property to be checked.

Since the deadline constraint is true if at least one schedule violates its requirement, if the resulting SMT-problem has a solution this means that there is at least one execution trace in which the system fails, thus the configuration is not a solution. Note the inverse of the result: if no SMT model is found then the 5G-SO system is safe.

7. Modeling and Verification Framework

By combining the UML profile (Sec. 4) with the semantics (Sec. 5 and Sec. 6) we obtain a framework which can be used to model various 5G-specific scenarios, and analyze them by employing a suitable back-end (e.g., UPPAAL, Z3). Our framework has two benefits: (1) a practitioner can use an industrially-accepted UML tool to model the system without knowledge of the underlying TA and first-order logic modeling, and (2) the verification results provide guarantees of the 5G-SO modeled behavior.

The intended workflow of the framework is to start from a case study and its requirements in natural language. First, the UML5G-SO is applied profile and a structural representation is created using a class diagram. Some of the classes in the diagram, referred to as active classes, have behavior modeled by UML statecharts. Next, the requirements of the case study are formalized either as (T)CTL queries or SMT formulas. This part is meant to be done by an expert working with both 5G as well as the underlying semantics.

Afterwards, a 5G expert can continue work without knowledge of formal methods. To verify a particular 5G-SO system, the class diagram is instantiated, yielding an object diagram of the system. Secondly, the semantics are applied (using TA templates or SMT formulas). Finally, the corresponding queries or formulas are analyzed to verify the system. We provide a tool, G^5, which can execute the second and third step automatically.

7.1. G^5

As a proof-of-concept of how employing our profile allows for automatic verification of object diagrams, we provide a tool, G^5[5], which facilitates verifying the assertion that all deadlines are met in a specific 5G-SO system. The tool is implemented in Python and takes as input an object diagram, specified in the `soil`-format (of the USE tool [28]); it can automatically generate both a TA model and an SMT model, and use a corresponding back-end to verify that all deadline requirements of slices are met. The user designs their system in an UML tool, saving an object diagram corresponding to the system they wish to analyze. Then in the G^5 tool, the diagram is loaded and by the press of a button, deadline-violations can be checked.

[5] https://github.com/ptrbman/GGGGG

7.2. Comparison of semantics

We have now presented two formalizations of the 5G system semantics, one based on TA and the other on first-order logic. They are formulated independently, but aimed at expressing the same behavior. Consider a counter-example found in the SMT-model of a particular 5G system. We can then extract all the schedules with their action-time pairs and form an execution trace of the system. In the same manner, if the deadline-query of the TA formalization is unsatisfied then there exists a counter-example that can also be seen as a trace.

We conjecture that it is always possible to translate a counter-example trace (from TA) to corresponding schedules with action-time pairs (to SMT) and vice versa. If this conjecture holds, both formalizations should always yield the same answer to the question whether a deadline is violated or not. We explore this further in the experimental evaluation.

8. Case study

As a case study, we consider a robot-assisted surgery application and a video-streaming application, accessing a health slice and a video slice, respectively, within the same small cell (i.e., bandwidth resources are consumed from the same cell tower). Applications access their respective slices via their 5G user equipment (UE). In our case study, a 5G camera accesses an instance of the *health slice*, and a mobile phone accesses an instance of the *video slice*. We consider that health slices have a higher priority over video slices.

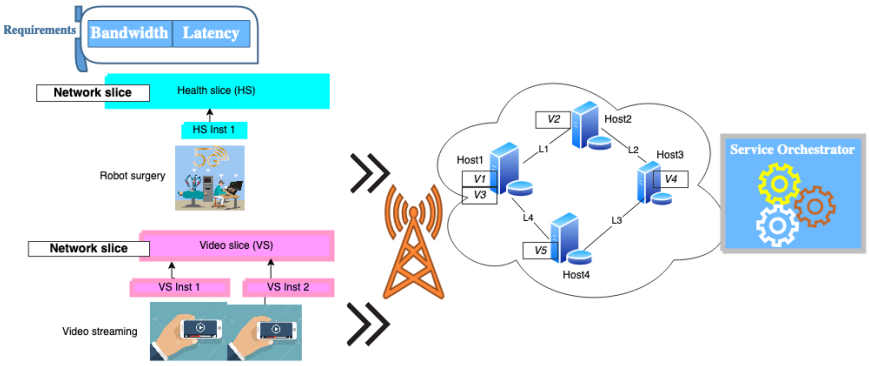


Fig. 9. Case study

We assume that we have one UE accessing the health slice and two UE accessing the video slice. The case study is depicted in Fig. 9. Slices consist of their respective VNF instances, e.g., the health slice consists of VNF sequence $[V_1, V_2]$, and the video slice has the VNF sequence $[V_1, V_3, V_4, V_5]$. Note that the slices share V_1. The overlay network comprises four hosts onto which VNFs are allocated. For instance, as shown in Fig. 9, we consider that V_1 and V_3 are allocated to Host1. Furthermore, the hosts are connected via

virtual links, e.g., Host1 and Host2 are connected via $L1$. However, not all hosts have a one-to-one link connectivity established, e.g., there is no direct link between Host1 and Host3. Hence, any routing scheme from Host1 to Host3 must use the path consisting of links $[L_1, L_2]$, or the alternative path $[L_4, L_3]$. The slices cater to different application requirements, for example while the robotic surgery application requires both low latency and high bandwidth, the video streaming application has high bandwidth requirements, but not critical latency constraints. We focus on two requirements in this case study, the end-to-end latency requirements of the slices are met and that the end-to-end bandwidth requirements of the slices are met.

8.1. Modeling with the Framework

The class diagram description of our case study is shown in Fig 10. We apply stereotypes and add attributes and functions, in the same manner as in Example 4. We model two categories of 5G user equipment, $5GCamera$ and $MobilePhone$, which access the two kinds of slices, $HospitalSlice$ and $VideoSlice$, respectively. We also apply stereotypes on $ReqControl$, $MonitorReq$ and VM in the same manner as in Example 2. We also present an object diagram modeling of the use case. In Fig. 11 all the components of the system instance is shown with their assigned attributes.

We need to define queries corresponding to the requirements. By construction, both formalizations only gives solutions respecting bandwidth requirements. For latency requirements, we use the queries in Eq. 12 for the TA model and Eq. 48 for the SMT model.

8.2. Experimental Evaluation

We conducted an experimental evaluation of our framework on the use case. All experiments are run on a Mac, with 1,4 GHz Quad-Core CPU and 16 GB RAM, using G^5.

Use Case Instantiation To create a TA model we instantiate our TA patterns described in Sec. 5, in correspondence with the object diagram (compare with Fig. 11). In order to verify the model, we assume that the network has enough request controllers and monitors to handle the requests from independent UE (i.e., we find parameters s.t. it is sufficient).

For the SMT model we instantiate the formulas described in Sec. 6. Instead of finding a sufficient number of monitors and executors, we need to create one schedule for each request (i.e., the sum of all maximum activations of all user equipment). We then generate a model as described above to check individual queries.

One crucial common factor of performance between the two models is the maximum activation times. The number of required monitors and executors is heavily dependent on these numbers (as well as the number of schedules). We set the maximum activation of all user equipment to the same value, and will refer to this as MA. To check scalability, we set this value to one and try with increasing values until timeout.

We verify the generated model against the formalized requirement, using the UPPAAL model checker version 4.1.19 [6]. We also add error locations to capture queues being full, and insufficient request controllers or monitors available (e.g., see Fig. 8). In case the

[6] https://www.it.uu.se/research/group/darts/uppaal/download.shtml

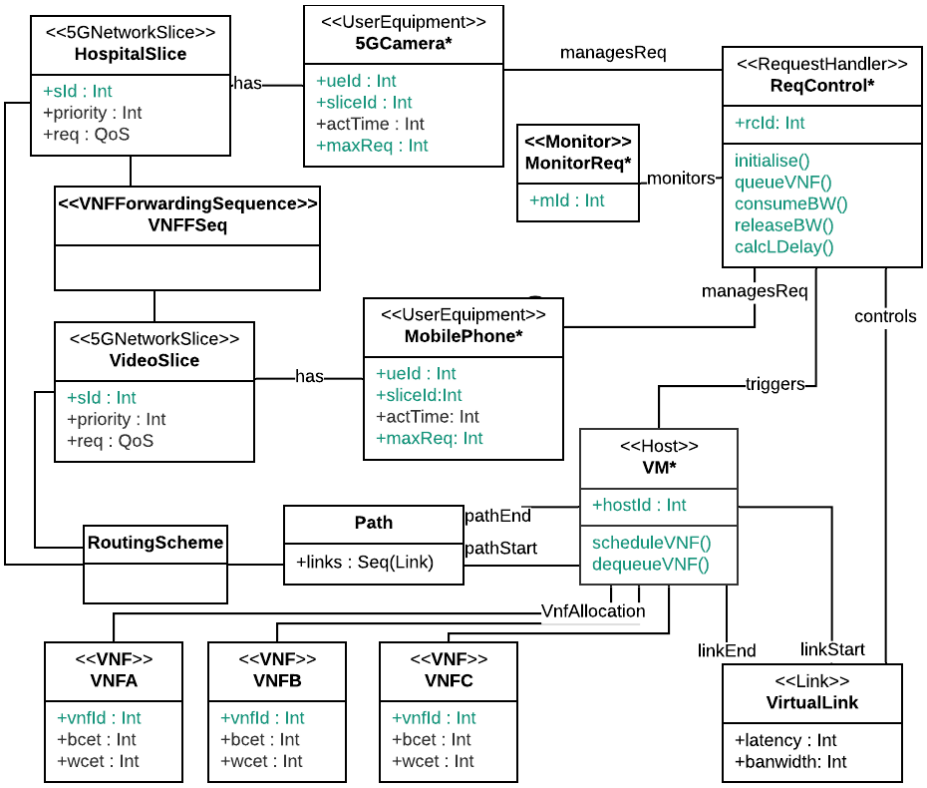


Fig. 10. Class diagram representation of our case study (It inherits all relations and multiplicities in the profile). Active classes are denoted with an asterisk.

verification fails, the mechanism allows for detecting if the model has reached any of the error locations. From Table 3, one can see that the time taken for exhaustive verification of certain queries is not promising, hence we acknowledge that exhaustive model checking may not always be the best solution at hand to verify big complex systems like a 5G-SOS, as it suffers from state-space explosion. However, if one is able to model and verify some critical part of the network and functionalities, it provides guarantees over all possible system behaviors. As shown in Table 3 the scalability of the SMT model is better than for the TA model, but it is still non-linear (as expected, as the number of constraints are quadratic in the number of schedules).

8.3. Discussion

The use case helps us see how it is possible to draw up UML diagrams (class and object) to describe a scenario and then analyzing it using formal methods. Currently, the number of request controllers and monitors needs to be manually by the user, as we have not found a method to calculate a reasonable bound, but other than this most details of the verification process is hidden.

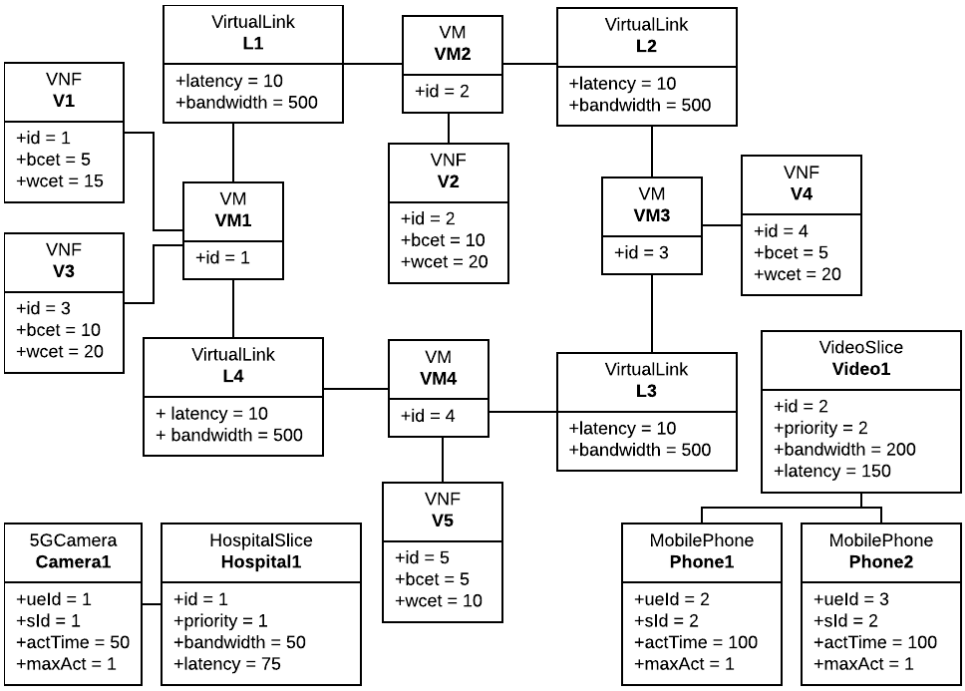


Fig. 11. Object diagram representation of the use case

The performance of both models (SMT and TA) are not enough to analyze systems with a large number of activations of user equipment. Nevertheless, it can be used to analyze systems of small size and there is room for optimizations which could greatly increase the performance. It is also important to emphasize, that in contrast to many other approaches which has a more detailed view of a similar system (see e.g., [10]), we are not aiming at just running or simulating a system but modeling it formally s.t. it is analyzable.

It is clear that the SMT model scales better than the TA model (at least for the use case). However, all these results are satisfied, and the performance could be different when compared on unsatisfiable use cases. Moreover, (T)CTL queries can be easier to work with for formulating properties, and UPPAAL allows for studying a single execution and going back and forth to analyze and make different choices.

Table 3. Results and running-times of checking deadline on the use case with both models for different maximum activations (MA). All times are in seconds.

MA	1	2	3	4	5	6	7	8	9	10	11
TA Time	31.21	T/O	-	-	-	-	-	-	-	-	-
TA Result	sat	-	-	-	-	-	-	-	-	-	-
SMT Time	0.04	0.20	0.82	2.90	6.56	70.87	46.35	903.23	1399.85	321.38	T/O
SMT Result	sat	sat	sat	sat	sat	sat	sat	sat	sat	sat	-

To help establish the equivalence of the models, we conducted a simple experiment generating 100 systems (with four hosts, five links, two slices, eight VNFs and two UE). These were tested using both models checking if their answers agree, and in every case they did. Of course this does not prove the models equivalence but shows support for it.

9. Related Work

Substantial work within the field of 5G service orchestration is aimed at providing optimal VNF placement algorithms and routing schemes [7,18,1,6]. There is also interesting work looking into scheduling of VNFs [2], and slice chain reconfiguration [16]. In [29], Yuan et. al. applies machine learning for resource allocation in network slicing, although this is at a lower level (radio) than what we are looking at in this paper. Outside the scope of 5G, there is work on allocating virtual machines among hosts in an energy-efficient manner [3], which could be adapted and integrated into a VNF placement scheme.

However, not much effort has been invested in modeling and formal analysis of 5G orchestration systems, which in turn would verify if a given VNF placement, resource allocation, and routing meets the application requirements. Nevertheless, there exists interesting work that considers the description of VNFs and VNF chaining in isolation, to analyze if application requirements are met, for instance, the Gym framework [25] and the work by Peuster and Karl [23]. In contrast to our work, these approaches model VNFs and their chaining at a low level, without considering a system perspective, or 5G-specific scenarios. Spinoso et al. [27] employ SMT solvers (e.g., Z3), to verify VNFs and VNF chains against safety and reachability properties. Although the approach is promising, the framework is strictly formal, lacking the bridge to industrially-accepted modeling languages such as UML. In addition, the authors do not consider the service-orchestration problem and the QoS requirements that we study in this paper. In another interesting work [17], Luque-Schempp et al. investigate the use of formal methods in the context of a 5G network, focusing on Software Defined Networking and Network Function Virtualization modeling and verification via selected formal tools like theorem provers, model checkers, and SMT solvers, at different levels of network abstraction, without considering 5G service orchestration. Unlike our approach, the framework does not employ modeling techniques other than those of the formal tools, making it less appealing to practitioners.

Even if placed outside the 5G realm, the work on modeling E-service orchestration by Petri Nets [20] could be extended to use the timed variant of the formalism together with the respective tool support (e.g. TINA model checker) to solve the 5G orchestration problem. However, our UPPAAL-supported approach benefits from the several options for diagnosis, as well as replaying counterexamples in the tool's user-friendly GUI.

In the domain of E-health, there is a proposed framework for fog-assisted monitoring of patients [10]. A prototype testing efficiency of introducing fog/nodes is created and can therefore provide empirical evidence of improvement, while we hope that with a formal methods framework like the one in this paper, such improvements can instead be proven.

The use of UML to model 5G service orchestration is not investigated much in the literature. One recent work [22] models 5G network slices, namely, resource driven, service driven, deployment driven, using different UML diagrams. However the modeling is not backed by formal analysis like the one presented in this paper.

10. Conclusions and Future Work

In this paper, we have proposed a framework to model and analyze dynamic 5G service orchestration systems. Our solution combines the features of user-friendly UML modeling with formal analysis using the UPPAAL model checker, or SMT checking, and provides one with automated support to model and formally verify the structure and behavior of dynamic 5G SOS systems. Using our tool support requires knowledge of UML and of the UML5G profile, but *no experience with timed automata or model checking*. This shows the power of complex automatic reasoning tools when provided to a UML user.

In our current model, we have considered only dynamic behavior arising from simultaneous user requests, VNF sharing, variable link and server utilization, etc. In the future, we would like to consider other factors like host failures, which entails VNF reallocation or rerouting of network traffic through an alternative path.

Acknowledgments. This work is supported by the EU Celtic Plus/Vinnova project, Health5G - Future eHealth powered by 5G, and the KKS synergy project ACICS - Assured Cloud Platforms for Industrial Cyber-Physical Systems, which are gratefully acknowledged.

References

1. Agarwal, S., Malandrino, F., Chiasserini, C.F., De, S.: Joint vnf placement and cpu allocation in 5g. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. pp. 1943–1951. IEEE (2018)
2. Alameddine, H.A., Qu, L., Assi, C.: Scheduling service function chains for ultra-low latency network services. In: 2017 13th International Conference on Network and Service Management (CNSM). pp. 1–9. IEEE (2017)
3. Alsbatin, L., Öz, G., Ulusoy, A.H.: Efficient virtual machine placement algorithms for consolidation in cloud data centers. Computer Science and Information Systems 17(1), 29–50 (2020)
4. Alur, R., Courcoubetis, C., Dill, D.: Model-checking for real-time systems. In: Logic in Computer Science, 1990. LICS'90, Proceedings., Fifth Annual IEEE Symposium. pp. 414–425. IEEE (1990)
5. Börger, E., Cavarra, A., Riccobene, E.: Modeling the dynamics of uml state machines. In: International Workshop on Abstract State Machines. pp. 223–241. Springer (2000)
6. Choyi, V.K., Abdel-Hamid, A., Shah, Y., Ferdi, S., Brusilovsky, A.: Network slice selection, assignment and routing within 5g networks. In: 2016 IEEE Conference on Standards for Communications and Networking (CSCN). pp. 1–7. IEEE (2016)
7. D4.1. Definition of service orchestration and federation algorithms, service monitoring algorithms. http://5g-transformer.eu/index.php/deliverables/, accessed: 2020-07-24
8. Douglass, B.P.: Real time UML: advances in the UML for real-time systems. Addison-Wesley Professional (2004)
9. Gérard, S., Selic, B.: The uml–marte standardized profile. IFAC Proceedings Volumes 41(2), 6909–6913 (2008)
10. Hu, J., Liang, W., Zeng, Z., Xie, Y., Yang, J.E.: A framework for fog-assisted healthcare monitoring. Computer Science and Information Systems 16(3), 753–772 (2019)
11. Kunnappilly, A., Backeman, P., Seceleanu, C.: Uml-based modeling and analysis of 5g service orchestration. In: 27th Asia-Pacific Software Engineering Conference, APSEC 2020, Singapore, December 1-4, 2020. pp. 129–138. IEEE (2020), https://doi.org/10.1109/APSEC51365.2020.00021

12. Kunnappilly, A., Backeman, P., Seceleanu, C.: From UML modeling to UPPAAL model checking of 5g dynamic service orchestration. In: ECBS 2021: 7th Conference on the Engineering of Computer Based Systems, Novi Sad, Serbia. pp. 11:1–11:10. ACM (2021), https://doi.org/10.1145/3459960.3459965
13. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a nutshell. International journal on software tools for technology transfer 1(1-2), 134–152 (1997)
14. Leivadeas, A., Kesidis, G., Ibnkahla, M., Lambadaris, I.: Vnf placement optimization at the edge and cloud. Future Internet 11(3) (2019), https://www.mdpi.com/1999-5903/11/3/69
15. Li, X., Samaka, M., Chan, H.A., Bhamare, D., Gupta, L., Guo, C., Jain, R.: Network slicing for 5g: Challenges and opportunities. IEEE Internet Computing 21(5), 20–27 (2017)
16. Liu, Y., Lu, H., Li, X., Zhao, D.: An approach for service function chain reconfiguration in network function virtualization architectures. IEEE Access 7, 147224–147237 (2019)
17. Luque-Schempp, F., Merino-Gómez, P., Panizo, L., et al.: How formal methods can contribute to 5g networks. In: From Software Engineering to Formal Methods and Tools, and Back, pp. 548–571. Springer (2019)
18. Mahboob, T., Jung, Y.R., Chung, M.Y.: Dynamic vnf placement to manage user traffic flow in software-defined wireless networks. Journal of Network and Systems Management pp. 1–21 (2020)
19. Marchetto, G., Sisto, R., Valenza, F., Yusupov, J.: A framework for verification-oriented user-friendly network function modeling. IEEE Access 7, 99349–99359 (2019)
20. Massimo Mecella, F.P.P., Pernici, B.: Modeling e-service orchestration through petri nets. In: 3rd Int. Workshop on Technologies for E-Services, TES 2002. vol. LNCS 2444, pp. 38—-47. Springer-Verlag (2002)
21. de Moura, L., Bjørner, N.: Z3: An efficient smt solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
22. Papageorgiou, A., Fernández-Fernández, A., Siddiqui, S., Carrozzo, G.: On 5g network slice modelling: Service-, resource-, or deployment-driven? Computer Communications 149, 232–240 (2020)
23. Peuster, M., Karl, H.: Understand your chains: Towards performance profile-based network service management. In: 2016 Fifth European Workshop on Software-Defined Networks (EWSDN). pp. 7–12. IEEE (2016)
24. Ramos, M.A., Masiero, P.C., Penteado, R.A., Braga, R.T.: Extending statecharts to model system interactions. Journal of Software Engineering Research and Development 3(1), 1–25 (2015)
25. Rosa, R.V., Bertoldo, C., Rothenberg, C.E.: Take your vnf to the gym: A testing framework for automated nfv performance benchmarking. IEEE Communications Magazine 55(9), 110–117 (2017)
26. Smullyan, R.M.: First-Order Logic. New York [Etc.]Springer-Verlag (1968)
27. Spinoso, S., Virgilio, M., John, W., Manzalini, A., Marchetto, G., Sisto, R.: Formal verification of virtual network function graphs in an sp-devops context. In: European Conference on Service-Oriented and Cloud Computing. pp. 253–262. Springer (2015)
28. USE: UML-based Specification Environment. https://sourceforge.net/projects/useocl/, accessed: 2020-07-24
29. Yuan, S., Zhang, Y., Qie, W., Ma, T., Li, S.: Deep reinforcement learning for resource allocation with network slicing in cognitive radio network. Computer Science and Information Systems 18(3), 979–999 (2021)

Peter Backeman is an Associate senior Lecturer at Mälardalen University, School of Innovation, Design and Engineering at the Formal Modeling and Analysis of Embedded

systems research group. He holds a Ph.D. in Computer Science from Uppsala University, which focused on working with SMT-solver and theorem provers. Currently, he is working with how to model embedded systems and apply different kinds of formal verification techniques.

Ashalatha Kunnappilly is a Train Traction Control System Engineer at Alstom AB, Västerås, Sweden. She holds a Masters in Embedded Systems (Amrita University, 2013) and a Ph.D. in Computer Science (Mälardalen University, Sweden, 2021). Her research focuses on developing formal models and verification techniques for predictable real-time systems.

Cristina Seceleanu is Associate Professor at Mälardalen University, School of Innovation, Design and Engineering, Networked and Embedded Systems Division, Västerås, Sweden, and leader of the Formal Modeling and Analysis of Embedded Systems research group. She holds a M.Sc. in Electronics (Polytechnic University of Bucharest, Romania, 1993) and a Ph.D. in Computer Science (Turku Centre for Computer Science, Finland, 2005). Her research focuses on developing formal models and verification techniques for predictable real-time and autonomous systems. She currently is and has been involved as organizer, co-organizer, and chair of relevant conferences and workshops in computer engineering, and is a member of the Editorial Board of the International Journal on Software Tools for Technology Transfer (STTT), Springer, and the Frontiers in Computer Science: Theoretical Computer Science.

Received: October 01, 2021; Accepted: September 01, 2022.

Computer Science and Information Systems 20(1):359–380 https://doi.org/10.2298/CSIS210923060M

Blockchain-based model for tracking compliance with security requirements*

Jelena Marjanović, Nikola Dalčeković, Goran Sladić

Faculty of Technical Sciences, Trg D. Obradovića 6,
21000 Novi Sad, Serbia
{jelena.stankovski, nikola.dalcekovic, sladicg}@uns.ac.rs

Abstract. The increasing threat landscape in Industrial Control Systems (ICS) brings different risk profiles with comprehensive impacts on society and safety. The complexity of cybersecurity risk assessment increases with a variety of third-party software components that comprise a modern ICS supply chain. A central issue in software supply chain security is the evaluation whether the secure development lifecycle process (SDL) is being methodologically and continuously practiced by all vendors. In this paper, we investigate the possibility of using a decentralized, tamper-proof system that will provide trustworthy visibility of the SDL metrics over a certain period, to any authorized auditing party. Results of the research provide a model for creating a blockchain-based approach that allows inclusion of auditors through a consortium decision while responding to SDL use cases defined by this paper. The resulting blockchain architecture successfully responded to requirements mandated by the security management practice as defined by IEC 62443-4-1 standard.

Keywords: industrial control systems, secure development lifecycle, blockchain.

1. Introduction

With technological improvements that are permeated through various aspects, software engineers and everyone involved in product development have become more aware of the impact a potential bug can produce on everyday life. While bugs in the production can lead to disrupted availability of the service or a product, a vulnerability could also lead to loss of confidentiality and integrity of the system. Those vulnerabilities have a greater impact if they were to occur in an ICS, as they perform data acquisition and real-time control [1] and the root cause of those vulnerabilities must be addressed [2]. A prime example of ICS is the Supervisory Control and Data Acquisition (SCADA), and if a vulnerability was to enter such a system, it may cause blackouts, thus leaving cities without power. A flaw in the energy management system led to a blackout in the northeastern U.S. in 2004, which could have been prevented if the code audit had been done in the implementation phase [3]. Another worrying evidence is that critical infrastructures have become a target for various cyber-attacks, where threat actors vary from competitors, hacktivists, cyber-criminals, nation-states. The impact of cyberattacks

* This is an extended version of the ECBS 2021 conference paper ”Improving Critical Infrastructure protection by Enhancing Software Acquisition Process Through Blockchain”.

that have happened to ICS, starting with the first publicly known SCADA cyberattack [4] to the latest Colonial pipeline malware attack [5] is ever-increasing, which has also been noted by several authors [6], [7], [8]. Cyber-attacks in the global oil supply chain were previously recognized as a potential issue, so authors of [9] have analyzed cyber threats and have provided immediate countermeasures. A model and an algorithm to optimize the survivability of a mission-critical system under attacks for a certain time duration by maintaining redundancy of components can be used when designing such a system [10]. Vulnerabilities in industrial control systems have shown that cybersecurity posture must improve, and the root cause of ICS vulnerabilities should be addressed [11].

Performing the root cause analysis for issues that have occurred is the way to minimize future mistakes and learn in the process, but a more beneficial approach is the shift left approach, i.e., implementing security checks from the early development. Such an approach can be complemented by implementing the industry best practices from a secure development lifecycle process. The standard IEC 62443-4-1 [12], named Secure product development lifecycle requirements, SDL for short, helps industrial automation and control systems (IACS) increase their security posture, by implementing security best practices in every aspect of the product development lifecycle. The IEC 62443-4-1 standard is divided into eight practices, addressing security requirements definition, secure design, secure implementation (including coding guidelines), verification and validation, defect management, patch management, and product end-of-life. [12]. As the IEC 62443-4-1 standard is written in the form of 47 requirements, the process of requirement engineering is of great importance. This process has been utilized by various industries, as it allows requirements to go through several stages and can be tracked through their phases. Requirement engineering practices has been analyzed and improved [13], [14], [15], [16], evaluated for startups [17] and adjusted for cyber-physical systems [18], [19].

Requirement engineering is a process that follows the lifecycle of a requirement. An approach to keep information dated and versioned, while at the same time have a tamper-proof resolution that guarantees that information that was written has not been altered, is to utilize features that blockchain technology provides. While some industries require that information stored on the blockchain is made publicly available and is required that the information is publicly verified, most of the industries have decided to keep some or all the information available only to interested parties. Private blockchain networks are suitable for corporations that need to utilize blockchain technology but the information that is stored on the blockchain cannot be publicly available. Hyperledger Fabric is a distributed ledger, used for creating blockchain solutions that require a private permissioned blockchain network, a network that is created and maintained by a pre-authorized set of members. An overview of blockchain classification was done by Golosova et al. [20], where the difference between public, private, permissioned, and permissionless blockchain was provided. Hyperledger Fabric has smart contracts, transactions, peers, consortiums as other blockchain implementations, but it has also introduced terms such as organization, ordering service, and channel [21] [22].

A step forward was made in requirement tracking, as the author proposed requirement tracing utilizing blockchain technology [23]. Blockchain technology provides immutability of information being stored on the distributed ledger. Demi et al. [24] claim that blockchain has the potential to enhance the immutability, trust, visibility, and

traceability of requirements throughout the software development lifecycle (SDLC). What we see as an appropriate extension to Demi's hypothesis [23], that would be beneficial to ICS wanting to improve its security posture, is to have a private permissioned, blockchain-based model, so that organizations can track and manage security requirements throughout the secure development lifecycle, which would allow ICS to promote cooperation and trust among different parties. The extension we made is to utilize only private-permissioned blockchain networks, as ICS will not set its requirements and their compliance on a public blockchain, that would be accessible to anyone with appropriate tools, and to focus on security requirements for ICS. We propose a blockchain-based model for tracking compliance with security requirements, as blockchain technology provides timestamped and tamper-proof information that is stored on the ledger, which is beneficial to parties auditing the process. Additionally, a universal blockchain architecture that can be used within the proposed model is offered. We chose to evaluate and provide details of this model, on Security Management practice, as a governing practice that ensures all other practices in the IEC 62443-4-1 standard are executed appropriately.

The remainder of this paper is organized into five sections. Section 2 provides review of related work. Section 3 introduces a proposed blockchain-based model and architecture. Section 4 evaluates the proposed blockchain model for security management requirements, combined with supply-chain management architecture. Section 5 concludes with a final discussion on the solution and future steps.

2. Related Work

The issue of addressing security practices from the beginning of a product lifecycle has been discussed by several authors [25], [26] relying on various standards and guidelines that provide knowledge on incorporating security into the product. Secure development lifecycle (SDL), whether product or software is the intended area of applicability, is a process of building secure products or software, by encompassing security and privacy considerations throughout all phases of the development process, helping developers to build highly secure software while addressing security compliance requirements, and reducing development costs [27]. Security standards and guidelines change over time, as seen in the case of Comprehensive, Lightweight Application Security Process-CLASP, which has been put to archive, but its segments are incorporated into IEC 62443-4-1 standard [12], [28]. One of the segments that were not incorporated directly into the IEC 62243-4-1 standard is roles and their responsibilities. Instead, only a requirement for defining roles and responsibilities is added (SM-2 Identification of responsibilities). Apart from CLASP, NIST Special Publication 800-64 Rev. 2 [29], named Security Considerations in the System Development Life Cycle, has also been withdrawn, guiding readers to refer to NIST SP 800-160 Volume 1 [30]. Microsoft's SDL has been guiding software developers over the last two decades [31], [32], [33]. Differences between CLASP and Microsoft's SDL have been addressed several times [34], [35], [36] which is beneficial to those gaining broader knowledge. Another standard that takes into consideration security from the very begging of the product development lifecycle is IEC 62443-4-1 standard. The IEC 62443-4-1 standard is one of 13 standards included

in the IEC 62443 series, developed by the ISA99 committee. Standards within series are grouped in general, policies and procedures, system and component groups [12] covering a broad area of security for industrial automation and control systems, from Terminology, concepts, and model (IEC 62443-1-1) to Technical security requirements for IACS components (IEC 62443-4-2). IEC 62443 is a source of common understanding of cybersecurity-related issues for industrial and automation control system (IACS) owners, component developers, and service providers [12]. This paper focuses on the IEC 62443-4-1 standard and Security Management practice, presented as a first and crown practice, containing 13 requirements, ranging from the development process to continuous improvement.

The demand for secure development lifecycle practices has been identified by academia and industry, but the applicability and justification of resources, both human and financial, remained a debate. This issue with additional cost that security practices are believed to be adding to the development was discussed in [37], which concluded that, at the time, few cost-estimation models that take security into account have been proposed and that the existing models were not properly validated. While the argument that security practices introduce excessive overhead in terms of time and money, authors [32] showed that Microsoft's Secure Development Lifecycle can be used even on a small team, that consists of one developer, but argue that the proper cost-benefit analysis of implementing a robust framework on a small team, should be conducted. Another group of authors [38] assumes that security will introduce overhead in terms of time and additional human resources. There are some challenges when secure practices are left out of software development and they need to be introduced, particularly in agile web development relying on SCRUM methodology. Such development is based on fast feature production, which usually lasts less than 30 days. The authors argue that such a short period does not leave time for security practices to be implemented at full scale and propose a secure SCRUM process, allowing "agile" security activities to be introduced to the process. The process was evaluated by a team of developers, describing the process as "medium" agile and "medium" cost-effective. As authors assumed, such a process introduced overhead in terms of time, but the analysis of not applying secure practices, which could lead to security issues such as breach and DDoS, has not been discussed. An alternative perspective is that companies take security and secure practices differently, depending on the industry, size, and organizational structure. Also, security experts in companies have various roles, from security engineers, consultants, or auditors. The work done by security auditors was presented by authors [39] in form of interviews, providing insight into security practices, such as static code analysis, and penetration tests. As the authors have concluded, a combination of organizational processes, developer training, and tools is needed for improving application security. The traceability for the processes has not been discussed and that is gained by utilizing our proposed blockchain-based model, since the possibility of tracking all information that has been put on the decentralized ledger, digitally signed and tamper-proof information, is an out-of-the-box feature of the blockchain.

Further research shows that one direction in securing products is incorporating security tools, such as AttackSurface Host Analyzer (AHA), allowing continual monitoring and improving ICS, but such activity is not sufficient for an overall increase in product security posture [40]. An argument for poor SDL implementation in the industry may lay in the interviewees' perspective of lacking security experts [41], there is

a mechanism that can contribute to additional security practitioners in the early stages of developers' working life. Walden et al. [42] have recognized the need for secure development lifecycle courses at the undergraduate level, as adopting base principles of SDL at the begging of higher education, provides future engineers concepts and the importance of securing development lifecycle. The course introduced 10 modules and a web project for demonstration, allowing students both conceptual and practical knowledge of SDL, but the research lacked results from this introduction. Also, the importance of introducing quality materials and advanced techniques in teaching the value of secure development lifecycle was recognized by authors [43] where teaching security design analysis in a hybrid flipped classroom was introduced in the class of 2015/2016. The proposed framework showed that the newer generation had better learning outcomes, reflected in system understanding and dataflow diagram quality. Since the framework showed an increase in students' understanding, it could lead to creating additional security courses using the hybrid flipped classroom method, which would further increase overall understanding of the importance of a secure development lifecycle.

With our approach of enabling compliance for securing the product development lifecycle, other ICS, such as smart grids, that are highly regulated, can append their blockchain implementations, for their compliances. Smart grids in the USA are regulated by NERC CIP standards, that every utility is obligated to follow. Authors from [44] have recognized the potential that blockchain technology has in terms of making information available, secure and tamper-proof, which can be considered a prerequisite for standard compliance. Using blockchain proof of authority as a mechanism for providing widely witnessed evidence on what can be considered the truth, while not relying on a single party, authors in [44] have proposed a supply chain blockchain solution, which is per NERC Critical Infrastructure Protection 13 standard. Blockchain security controls that enable compliance with NERC CIP 13 standard, also enable Customers, Manufacturers, Hardware, and Software Suppliers, as identified actors, to exchange information in a secure, transparent, traceable, and tamper-proof manner. A similar group of authors [45] has also analyzed blockchain utilization for other NERC CIP standards: CIP 007-5, CIP 008-5, CIP 009-5, CIP 010-1, and CIP 011-1. Realization of compliance is suggested through using keyless signature blockchain infrastructure, while NERC CIP requirements were fulfilled through 18 critical controls. An in-depth analysis on facilitating compliance with NERC CIP 010 standard, which is aimed at configuration management and vulnerability assessment, is provided by authors in [46]. Requirements and their measures from NERC CIP 010 standard can be fulfilled by seven identified blockchain controls. While the authors from [44], [45], [46] describe how NERC CIP compliance can be achieved utilizing various blockchain implementations, they also point out that such an implementation should be thoroughly analyzed as it is still in the nascent stage [44].

The most comprehensive analysis of the IEC 62443-4-1 standard presented in several papers, from a similar group of authors [47], [48], [49]. The standard was analyzed in terms of applicability, deliverables, CMMI and authors have also addressed the topic of integrating IEC 62443-4-1 standard with agile software engineering. This analysis was done through several aspects, from creating a BPMN that includes SDL and agile process to interviewing key stakeholders of those processes. In the paper [49], authors made a self-assessment tool that can provide to development teams an insight to current

compliance with the IEC 62443-4-1 standard, without the need for additional, costly, involvement of external auditors. The tool consists of eight assessment sheets, each corresponding to practice in the IEC 62443-4-1 standard. Through three research questions, this tool was evaluated by Siemens employees, with different backgrounds in IEC 62443-4-1 standard and expertise in security compliance assessment. While utilizing this tool, the results can be explicitly tracked to the 4-1 requirements delivering a common ground for auditors and project participants, we argue that such an approach lacks traceability and is not tamper-proof.

Compared to other solutions, our proposed blockchain-based model differs in that it takes into consideration that ICS, which will follow certain SDL practices, wish to keep their information private, tamper-proof, and easily auditable. As the private-permissioned blockchain provides a tamper-proof solution, enabling only a pre-authorized set of users to participate in the process, our proposed model enables those ICS to fully commit to implementing security requirements that are part of the SDL process, while at the same time, have a way of incorporating auditing opportunities.

3. Requirement's Compliance Tracking Model and Architecture

Hyperledger Fabric is a private-permissioned blockchain network that enables only preauthorized users to participate in the network. This feature, combined with blockchain out-of-the-box traceability and tamper-proof features, can be utilized for tracking compliance with various security standards. While other solutions provide private permissioned blockchains, Hyperledger Fabric is focused on enterprise-level solutions, covering a wide range of industries that can utilize their solution. Given that the proposed model is oriented towards ICS that wish to comply with security requirements from Secure Development Lifecycle, such blockchain implementation seems suitable.

The proposed blockchain-based model is shown as an activity diagram in Figure 1, illustrating which steps the Project Manager, Security Advisor, and Blockchain Administrator should do to create a platform that would enable other participants to contribute to compliance with chosen standard requirements. In the beginning, the Project Manager chooses the standard to be implemented and Security Advisor checks for the applicability of that security standard, as the security expertise is within that role. If the standard is not applicable, this activity diagram is finished. Next, the Security Advisor should define use case diagrams for the requirements that are in the scope. Security Advisor can consult the Project Manager if any assistance is needed. All other steps are for the Blockchain Administrator, and that is to determine the number of channels, define organizations and consortiums based on uses case, and determine whether Inter Planetary File System (IPFS) should be utilized. Certain solutions, that involve blockchain network, require files to be uploaded. As the blockchain network is not created for storing files, an Interplanetary File System (IPFS) can be used. The IPFS is a peer-to-peer hypermedia protocol designed to make the internet faster, safer, and more open [50]. In a peer-to-peer network such as IPFS, if one node is down, other nodes in the network can serve needed files. Utilizing IPFS for storing files, the blockchain network remains solely for storing transactions and maintaining the world

state. Versioned documents will be available for download from IPFS, while the information about the version number, last update, and last modifier can be stored within smart contracts. Finally, the architecture, which depicts all organizations, consortiums, applications, and channels is created and deployed on the Hyperledger Fabric.

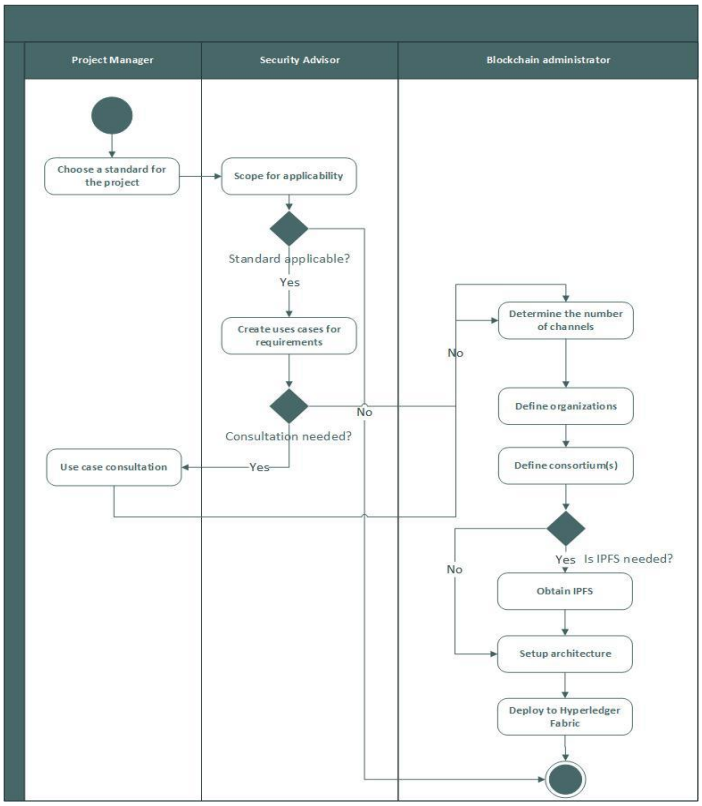


Fig. 1. Blockchain-based model for tracking compliance with security requirements

In addition to the proposed blockchain-based model for tracking compliance with security requirements, a customizable architecture is also proposed. This customizable architecture, that combines Hyperledger Fabric architecture and IPFS, enables the Blockchain Administrator to easily adjust use cases that have been created by the Security Advisor into a deployable architecture. The number of channels is the number of presented use cases, the organizations within Hyperledger Fabric are presented as actors in use cases and consortiums are presented as a use case that certain actors can update. Figure 2 shows a customizable architecture that can be adjusted to the required number of channels, organizations, and consortiums.

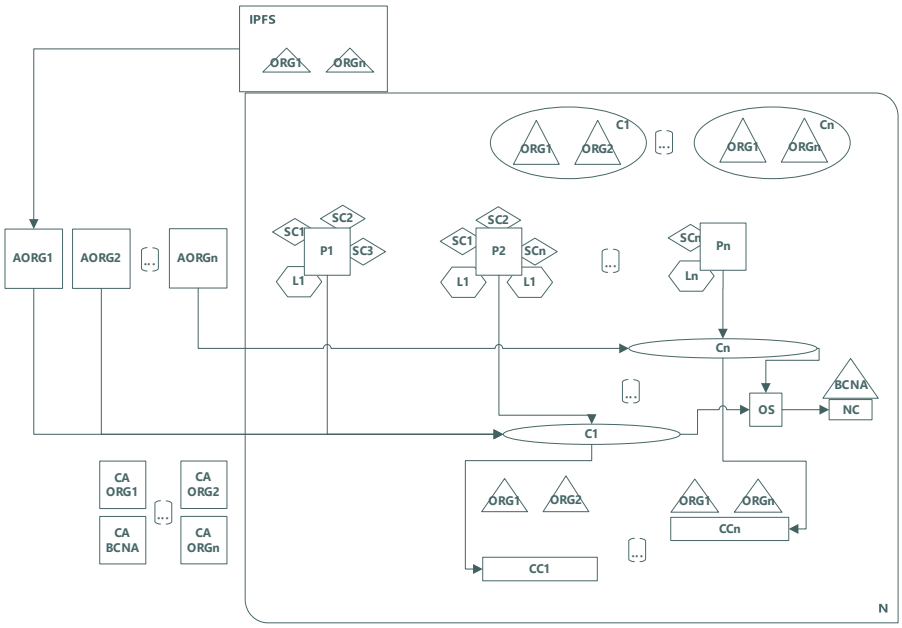


Fig. 2. Customizable architecture for tracking requirement compliance combining Hyperledger Fabric architecture and IPFS

The customizable architecture shown in Figure 2 represents the blockchain network N, with related Certificate Authorities, applications for organizations, and IPFS. The blockchain network N contains one Ordering Service (OS), one Network Configuration, and Blockchain administrator organization (BCNA). The number of consortiums (Cn), channels (CHn), channel configurations (CCn), organizations (ORGn), peers (Pn), smart contracts (SCn), ledgers (Ln), certificate authorities (CAn), and applications (AORGn) is defined through use cases and their actors. If IPFS is identified in the use cases, it can be added externally to the blockchain network and connected to the corresponding applications. Following this universal architecture, an architecture that is appropriate for the supply-chain management use case described in the next section will be discussed.

4. Model evaluation through Security Management practice

The blockchain-based model that has been proposed in the previous chapter will be evaluated on a security standard, guiding how to interpret the discussed steps. We have chosen IEC 62443-4-1 standard for Secure Product Development Lifecycle to evaluate how this model can be used for demonstrating compliance, as every information on the blockchain network is timestamped and digitally signed. Particularly, Security Management practice has been chosen for this paper, as the overall practice in IEC 62443-4-1 standard, whose implementation is a prerequisite for other practices. We also believe that the proposed model can be applied to other security standards that would require similar evidence for compliance.

The first step from the proposed model was for the Project Manager to choose a standard to be compliant with. That step was done by choosing all 13 requirements from Security Management practice, from IEC 62443-4-1 standard. The requirements are named with the prefix SM (Security Management) and have an assigned incremental number, as well as the name of the requirement, e.g., the SM-1 Development process is the first requirement from Security Management practice, named Development process. In the following sections, activities presented in the proposed model, from choosing the scoping applicability to setting up the architecture, are discussed.

4.1. Security Advisor activities

The first step for the Security Advisor, presented in Figure 1, is to perform the scoping and applicability decisions. The applicability of the standard requirements can be tracked on the blockchain, and it will be discussed later as part of the Security Management practice. The following step is to create use cases for the requirements and throughout this activity, Security Advisor can consult with the Project Manager. Although the requirements from IEC 62443-4-1 standard are grouped into practices, for Security Management practices, we have divided requirements into four divisions, as it offers better organization of channels and consortiums that need to be defined on a blockchain network. The divisions are:

1. Team and project management: this division covers requirements SM-2 through SM-5, as they require teams to have defined roles and responsibilities, team members should complete assigned trainings, and applicability of this standard to the product in scope should be done.
2. Development environment: these requirements are aimed at securing the development environment (SM-7), considering the development process (SM-1), while at the same time ensure the controls for private keys (SM-8) and ensure file integrity (SM-6).
3. Supply chain management: this division is for requirements SM-9 and SM-10, that are directed at vendor's supply chain and engaged 3rd party company, that provides custom-developed components.
4. Quality assurance: this division is created to gather requirements SM-11, SM-12, and SM-13, as they are focused on tracking security bugs to closure (SM-11). This proposed framework is essentially how the SM-12 requirement, named Process verification can be fulfilled. The requirement that focuses on increasing the SDL process maturity ultimately increasing software quality and security is Continuous improvement (SM-13).

These four divisions are the use cases that would satisfy the Security Management requirements, described below.

Team and project management use case

For this use case, which is named *Team&project management* and presented in Figure 3, actors Team and Auditors have been defined, where the Actor Team is a generalization for actors Project Manager, Security Advisor, and Software Component Manager. Every

actor must authenticate to the network, while the Project Manager and Security Advisor create and maintain a consortium.

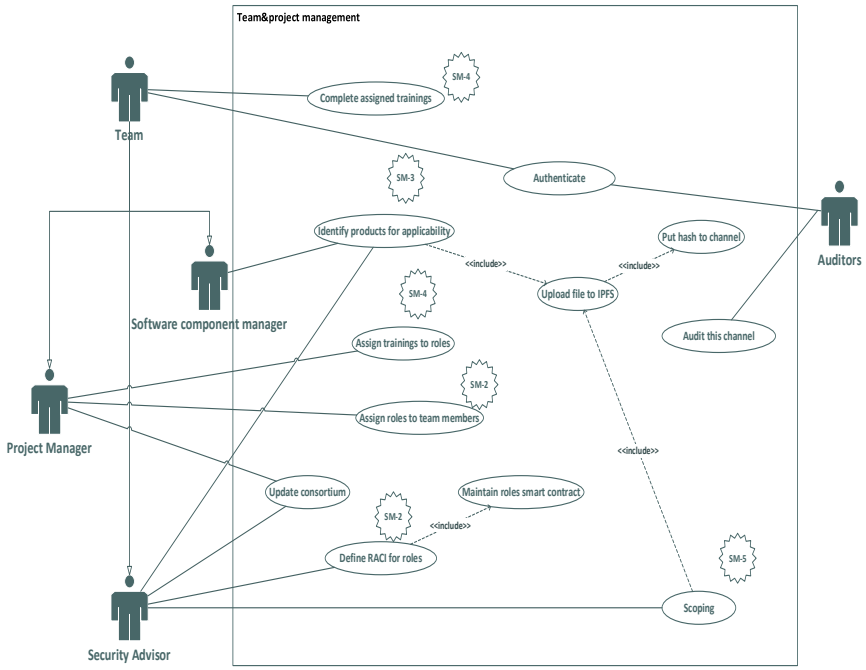


Fig. 3. Team&project management

The actors contribute to this use case through the following scenarios:

1. The actor Team: the generalization of the Project Manager, Security Advisor, and Software Component Manager is presented as the actor Team, as everyone involved in the SDL must complete assigned trainings. The completion of assigned trainings is required by the requirement SM-4 Security expertise. Through this requirement, Project Managers are expected to assign trainings to identified roles. The identification of roles, i.e., assigning roles to team members is expected from the Project Manager and is needed by requirement SM-2 Identification of responsibilities. This requirement includes defining RACI (responsible, accountable, consulted, informed) matrix, and that is done by Security Advisor, as that role possesses the knowledge. The Security Advisor is responsible for performing Scoping, which is directed by the requirement SM-5 Process scoping. As shown on the activity diagram in Figure 2, the Security Advisor will scope a standard and select requirements that the product should be made compliant with. This is also applicable for the requirement SM-3, Identification of applicability, where the Software Component Manager will provide information to which components SDL should be applied.
2. The actor Auditors: presented here as the role of auditing this channel, i.e., seeing that roles have been delegated, every role has completed the assigned

trainings, the processes artifacts for identifying applicability and scoping are created.

Development environment use case

The Development environment use case is created for grouping Software Component Manager's, Security Advisor's, and Project Manager's responsibilities, respectively, which are in line with the requirements SM-1 and SM-6 through SM-8. All actors that are presented in this use case, must authenticate to the blockchain network. The project Manager and Security Advisor will form a consortium, that will enable them to add or remove new organizations to the channel, as well as to limit their rights. The actor Auditors can be present at any channel, to inspect the compliance with those requirements.

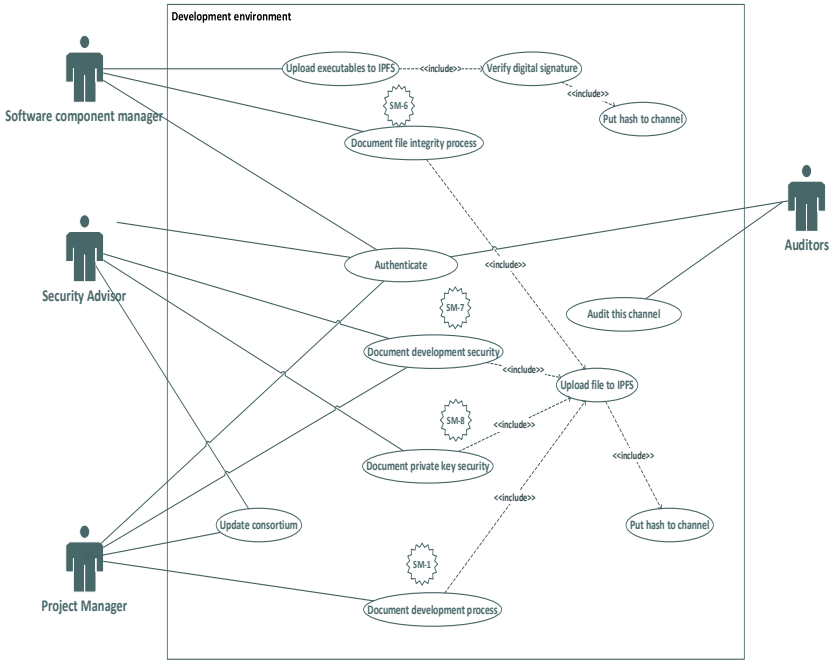


Fig. 4. Development environment channel

Following actors and scenarios contribute to this use case:

1. The actor Software Component Manager: responsible for uploading executable files to the IPFS, as the requirement SM-6 File integrity, requires the product to have a mechanism that shows that files have not been altered. Although the blockchain network is used for verifying that information has not been altered, this requirement is aimed at files and executables, which should not be placed on a blockchain network, rather they should be placed on IPFS. The upload of executable files should include verification of the supplier's digital signature, and

such information should be included in the channel, through a designated smart contract. This process can be described through a document and uploaded to the IPFS. Once the file is uploaded to the IPFS, the obtained hash should be stored on the smart contract.

2. The actor Security Advisor: the role of Security Advisor in this use case is seen through requirements SM-7 Development environment security and SM-8 Controls for private keys. Both requirements expect both technical and procedural controls to be put in place, but since technical controls, such as Hardware Security Modules for private keys should not be used by the blockchain network itself, both these requirements are focused on implementing the processes.
3. The actor Project Manager: this role is similar to the one Security Advisor has, as the requirement SM-1 Development process is aimed at documenting and enforcing product development processes, for configuration management, requirement engineering, implementation practices, etc. The process can be seen as a document that can be uploaded to the IPFS and that hash should be placed to the channel, through a smart contract.
4. The actor Auditors: the role of actor Auditors is to verify that requirements SM-1, SM-6, SM-7, and SM-8 have been met and that the traceable documentation is created, which can be done through inspecting the Development environment channel on the blockchain network. All information that is put on the blockchain network has timestamped and signed changes, which eases the auditing of channels and requirements.

Supply chain management use case

Participants in the process of managing the supply chain, seen through the integration of software components procured from different vendors, are defined as actors in the use case diagram shown in Figure 5. An actor called Software Component Vendor represents companies that provide software that can be custom-made for a specific customer or can be commercial-off-the-shelf (COTS) components. The actor Purchaser is the company that will be using the software that Software Component Vendor provides. The Purchaser actor is a generalization for actors named Software Component Manager, Security Advisor, and Project Manager. Also, the actor Auditors is presented in the figure which will be able to inspect the whole process of tracking software components' supply chain.

Every actor in the use case must authenticate to the blockchain network to be able to participate in any activity. Following actors contribute to this use case:

1. The actor Software Component Vendor: the actor Software Component Vendor is also responsible for filling the security questionnaire, which includes uploading the file to Interplanetary File System (IPFS). The security questionnaire is a document used for assessing the security posture of companies whose components will be integrated into the system. In case the Software Component Vendor is producing a tailor-made software component, besides filling the security questionnaire, the code should be deployed in a predefined repository, which is managed by a Software Component Manager of the

Purchaser. Through this security questionnaire, the requirement SM-9 Security requirements for externally provided components can be fulfilled. The security questionnaire will provide enough information to the Security Advisor, which is responsible for analyzing the questionnaire, to determine whether Software Component Vendor security posture is adequate.

2. The actor Purchaser: the actor Purchaser, as a generalization for actors Software Component Manager, Security Advisor, and Project Manager, can update a consortium for adding and deleting organizations. The actor Software Component Manager can manage the software component repository which is utilized by the Software Component Provider. The role of the Security Advisor is to perform questionnaire analysis, which includes obtaining the file from the IPFS. Through this generalization, requirements SM-9 and SM-10, can be fulfilled. The Security Advisor will inspect the security questionnaire and the Software Component Manager will manage the software component repository, which is in line with the SM-9 requirement, while the Project Manager will audit Software Component Vendor, which is defined by the requirement SM-10 Custom developed components from third-party suppliers.
3. The actor Auditors: if an external auditor wants to inspect the process of managing the software components that are either COTS or tailor-made for a Purchaser, that can be done by updating the consortium between Software Component Vendor and Purchaser. For Auditors to be able to inspect any of the ledgers, authentication must be performed against their CA. After successful authentication, Auditors can proceed with inspecting the security questionnaire. Once the audit is finished, Software Component Vendor and Purchaser can update the consortium so that Auditors are deleted from the channel, thus losing the capability to inspect the ledger.

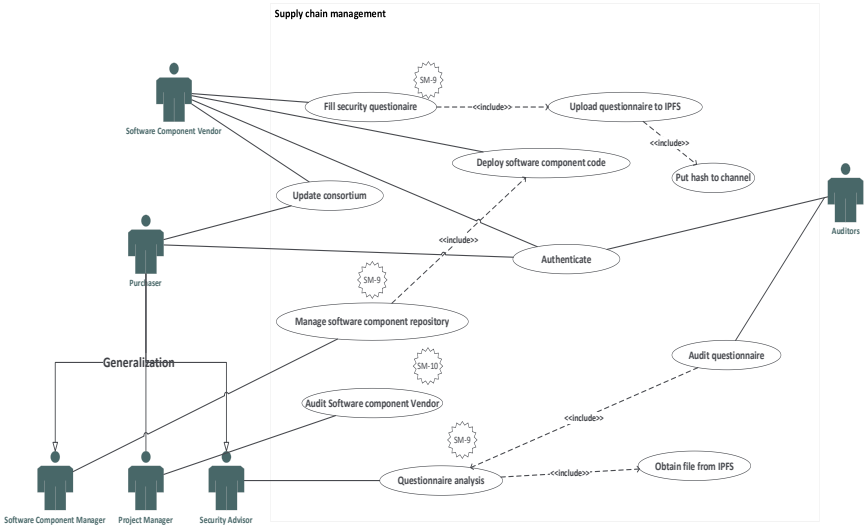


Fig. 5. Supply-chain management channel

Quality assurance use case

The actors Test analyst, Project Manager, Security Advisor contribute to the quality by following requirements SM-11 through SM-13. While the quality itself is permeated through several requirements in multiple practices in SDL, these requirements from the Security Management practice are focused on increasing the quality through management. As in previous use cases, all actors must authenticate to the channel. Future organizations, that are shown as Project Manager and Security Advisor actors in this use case, form a consortium, which enables them to manage the channel configuration and add or remove other organizations, such as Auditors, to the channel.

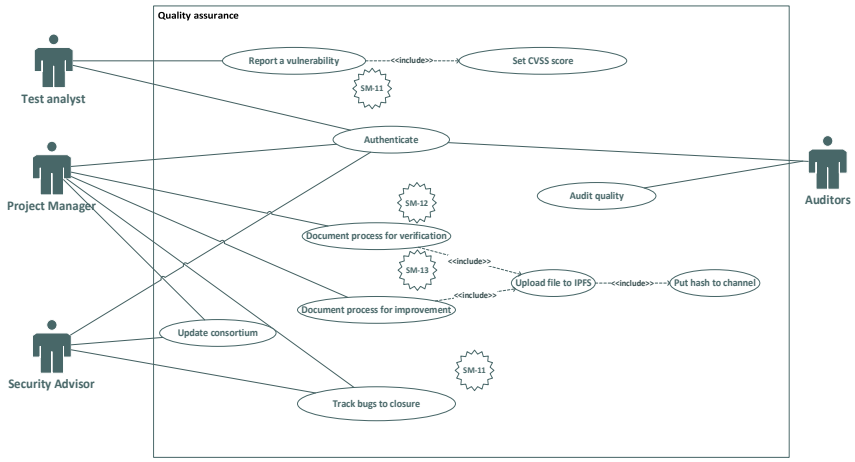


Fig. 6. Quality assurance channel

The following actors and scenarios are part of this use case:

1. The actor Test Analyst: the requirement SM-11 Assessing and addressing security-related issues is aimed at verifying that the product has not been released with security-related issues, but the role of the Test analyst is to report such vulnerabilities, which includes setting the correct CVSS score. Setting proper CVSS score will allow other participants in the channel to be aware of the critically and make prioritizations if such action is needed.
2. The actor Project Manager: this actor is responsible for tracking security-related issues to closure, which is defined by the requirement SM-11. Also, the Project Manager should contribute to the documents that should be made for compliance with SM-12 Process verification and SM-13 Continuous improvement. While the requirement SM-12 itself is incorporated into every use case diagram and enabled by the later proposed architecture, the document that describes the process itself can be uploaded to the IPFS, and the document hash should then be put to the appropriate smart contract on the channel. Similarly, the requirement SM-13 is incorporated into every use case diagram, as it allows the actor Auditors to inspect the processes, which enables them to suggest further improvements. The document that describes the improvement process can be uploaded to the IPFS and the hash can be put to the smart contract.

3. The actor Security Advisor: the role of Security Advisor is presented as a support for tracking security-related issues to closure, as this role contains the needed knowledge about security-related issues and can advise on prioritization if such is needed.
4. The actor Auditors: as in previous use cases, the actor Auditors can inspect the channel, once the consortium enables them read rights to the channel. This audit includes verification that all security bugs are tracked to closure, that Test Analysts have reported vulnerabilities with correct CVSS scores, and that proper documentation, is uploaded to the IPFS and can be tracked.

4.2. Blockchain Administrator activities

Due to paper limitations, only the blockchain architecture for the supply-chain management use case will be presented. The blockchain network architecture for tailor-made or COTS components is shown in Figure 7. This architecture is the proposed solution of how Purchaser can track software components that are developed specifically for that system, i.e., tailor-made or COTS components which are incorporated as-is in the system. The architecture is created as a specification of the customizable architecture, applied to the Supply-chain management use case. With such architecture, compliance with requirements SM-9 and SM-10 can be proven.

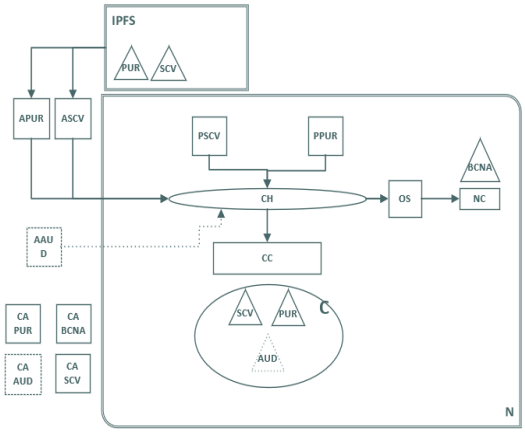


Fig. 7. Supply-chain management network architecture

Upon blockchain network creation, Network Configuration (NC), Organization Blockchain Network Administrators (BCNA), and Ordering Service (OS) are added. NC is the set of rules and policies allowing organization BCNA to maintain the network, add consortiums, new organizations, create channels, add new ordering services and peers. Consortium (C) is created between organization Purchaser (PUR) and Organization Software Component Vendor (SCV), which have formed a Channel (CH) through Channel Configuration (CC). This consortium is formed for organizations to be able to create their channel, with the necessary smart contracts to carry out the tracking of

software components through security questionnaire. Also, by leveraging organizations, only preauthorized members can write and read from the channel, following the need-to-know basis. Every organization in this solution has its own Certificate Authority (CA), which is presented as CA BCNA, CA PUR, and CA SCV. Also, organizations SCV and PUR have their peers who host the ledger, named PSCV and PPUR, respectively. Since organizations SCV and PUR have formed a consortium, they can add additional organizations to the channel, but only if both organizations agree on such activity. This is necessary in case an external auditor must examine the process. By changing the channel configuration, organizations PUR and SCV can add organization Auditors (AUD). For organization AUD to be able to participate in the channel, an application AAUD is added. As the use case diagram shows before any changes are made on the network, participants must authenticate, while the CA for Auditors is created (CA AUD). The AUD, AAUD, and CA AUD are represented by dotted lines as it is added by the organizations SCV and PUR when necessary and is removed once the audit has been finished. Creation of organizations that will be part of the blockchain network, reduces the risk of tampering with software components that are incorporated into the system as COTS or tailor-made components, as all actors are known in advance and are authenticated, while the blockchain technology provides a tamper-proof solution that guarantees that no information is changed once written on the ledger. All these features combined, allow compliance with requirements SM-9 and SM-10.

Applications APUR and ASCV contribute to the security questionnaire by uploading the document on the IPFS, after which they will receive a file hash-code which is then stored on Channel CH. Utilizing IPFS for storing files, the blockchain network remains solely for storing transactions and maintaining the world state. Versioned security questionnaire documents will be available for download from IPFS, while the information about the version number, last update, and last modifier will be securely stored on the Channel CH, within the smart contract Security Questionnaire. Both organizations SCV and PUR can contribute to security questionnaire through applications ASCV and APUR, which have interfaces for communication with IPFS and blockchain network.

Application and Smart Contracts

Following the Supply-chain management network architecture shown in the previous chapter, a small ExpressJS application was created. ExpressJS is a NodeJS framework, used for creating server-side web applications. The Ethereum network is a public blockchain network that provides vast number of tools available for creating and verifying blockchain smart contracts. One of the most popular test networks for Ethereum is the Ropsten network, allowing users to test their smart contracts without the need to invest in any platform or online service. For this prototype implementation, QuickNode as an Ethereum node was used, connected to the Ropsten test network. Utilizing Ropsten test network for prototyping, provides users an easy and unrestricted option of interacting with the blockchain. The architecture presented in Figure 7 includes IPFS, as a method of uploading documents, so for the prototype, a local IPFS node was used.

The smart contract named `Document` is a representation of the document that will be stored on the IPFS. On the ledger, the uploaded hash, as well as the document name and owner will be stored. `Document` constructor is created for instantiating a new `Document` with the given `documentName_` and `uploadedIPFSHash_`, while the `owner` of that created `Document` will be extracted from the global attribute `msg.sender`. The contract `SecurityQuestionnaires` is created for keeping the collection of documents, implemented as a `mapping` of `string` to a `Document`. Function `upload_questionnaire_hash` is called once the document hash is obtained from the IPFS and the `queryDocumentByName` function is called for retrieving document details. Following code snippet is written in Solidity programming language and provides insight on how smart contracts for `SecurityQuestionnaires` and `Document` are implemented.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract Document {
    address owner;
    string documentName;
    string uploadedIPFSHash;

    constructor(string memory documentName_, string memory
uploadedIPFSHash_){
        owner = msg.sender;
        documentName = documentName_;
        uploadedIPFSHash = uploadedIPFSHash_;
    }
}

contract SecurityQuestionnaires {
    mapping (string => Document) documents;

    function upload_questionnaire_hash(string memory documentHash_,
string memory documentName_) public {
        documents[documentName_] = new Document(documentHash_,
documentName_);
    }

    function queryDocumentByName(string memory documentName_) public
view returns (Document) {
        return documents[documentName_];
    }
}
```

The following snippet is written in ExpressJS and displays libraries for interaction with IPFS, Ethereum network and file system, which are introduced in the beginning of snippet and initialized with appropriate configurations. Provided `filePath` is used for creating a buffer which is uploaded to IPFS. The `url` is the QuickNode's endpoint to the Ropsten test network, which is used to create a `customHttpProvider` which is needed for creating a `signer`. The `signer`, together with the `address` of the smart contract and `abi`, is used for creating an object `contract`, which is used for calling the methods from the smart contract previously explained. Once the upload of the document is complete, IPFS returns the hash of that document, which is then uploaded to the `security_questionnaires` smart contract.

```
  var ipfsClient = require('ipfs-http-client');
  var ethers = require('ethers');
  var fs = require('fs');
  var ipfs = ipfsClient.create('http://localhost:5001')
  var url = QUICK_NODE_ENDPOINT;
  var customHttpProvider = new ethers.providers.JsonRpcProvider(url);
  var address = 'SMART_CONTRACT_ADDRESS';
  var abi = ABI_JSON;
  var signer = new ethers.Wallet(ethers.Wallet.fromMnemonic("PRIVATE_KEY,
customHttpProvider);
  var contract = new ethers.Contract(address, abi, signer);

  const uploadDocument = async function (filePath) {
      var testFile = fs.readFileSync(filePath);
      var testBuffer = Buffer.from(testFile);
      var uploadResult = await ipfs.add(testBuffer);
      contract.upload_questionnaire_hash(filePath, uploadResult);
  }
```

The snippets provided for the smart contracts and the ExpressJs display the most important part of the code which is needed for supporting the basic use case of uploading the document to the IPFS and storing the obtained hash on the ledger.

4.3. Discussion

The proposed model for tracking security requirements proposes utilization of Hyperledger Fabric, as a private permissioned blockchain network that enables only preauthorized users to contribute to the network. Though several papers [47], [48], [49] propose how requirements from the IEC 62443-4-1 standard can be tracked, our proposed blockchain architecture enables tamper-proof solution which is supported by blockchain basis. The authors [23] explore the idea of utilizing blockchain technology for tracking security requirements, but the paper doesn't address security requirements from IEC 62443-4-1 standard. The value of our work lies on the idea of utilizing private permissioned blockchain network for tracking security requirements that are critical when implementing secure lifecycle development for ICS. The ICS, such as smart grids, can benefit from this approach as it provides a solution which cannot be tampered with, leading to improved security posture, as well as plainer certifications. The certification process can be assisted by our solution, giving the auditors a timestamped and verifiable information about compliance with security requirements that have been stored on the blockchain. Private permissioned blockchain network should be utilized as the information stored on the blockchain should remain available only to preauthorized set of users. Though this paper presented a prototype implementation on the Ropsten network, i.e., one of Ethereum's test networks, the same principles can be applied to the Hyperledger Fabric. The public availability of smart contracts that have been added to the Ropsten test network enables easier verification of the prototype and simpler collaboration in this prototype phase. As this prototype implementation lacks the configuration that is needed for creating consortiums, which is only available in the Hyperledger Fabric, such improvements are planned in future work.

5. Conclusion

The ever-increasing cybersecurity threats that are emerging from various inputs, can be addressed by utilizing security practices, which are incorporated into the security development lifecycle (SDL) requirements. Implementation of those requirements involves providing evidence that the compliance has been met. The tamper-proof traceability that blockchain technology provides out-of-the-box, enables various interested parties, such as Auditors, to verify that the compliance with requirements, has been met. In this paper, we have presented a private permissioned, blockchain-based model, so that organizations can track and manage security requirements throughout a secure development lifecycle, which would allow ICS to promote cooperation and trust among different parties. Apart from the proposed blockchain-based model and universal architecture, an evaluation of the model was done against 13 requirements from the Security Management practice, from IEC 62443-4-1 Secure product development lifecycle. That evaluation involved discussion of the activities that have been set by the proposed model for the Project Manager, Security Advisor, and Blockchain Administrator. As the Project Manager has chosen the Security Management practice, the Security Advisor created four use cases that correspond to the grouped requirements from the SM practice. Those use cases were the inputs for the Blockchain Administrator to create an architecture for the supply-chain management use case. Within each use case, a possibility of allowing the Auditors to inspect the compliance with the requirements was given through modifying the consortiums, allowing Auditors to read tamper-proof information stored on the ledger. By utilizing private-permissioned blockchain technology, participants in this network are pre-authorized and have predefined rights. The future work shall include the design of guidelines for smart contracts that would enable further development of the architecture among all parties that are part of the supply chain process verification. Also, the usability of this framework for other security related standards, such as ISO 27001, shall be discussed.

References

1. Zhivich, Michael, and Robert K. Cunningham. "The real cost of software errors." IEEE Security & Privacy 7.2 (2009): 87-90.
2. Graham, J., Hieb, J., & Naber, J. (2016, June). Improving cybersecurity for industrial control systems. In 2016 ieee 25th international symposium on industrial electronics (isie) (pp. 618-623). IEEE.
3. Neumann, Peter G. "Risks to the public in computers and related systems." ACM SIGSOFT Software Engineering Notes 29.2 (2004): 8-16.
4. McLaughlin, Stephen, et al. "The cybersecurity landscape in industrial control systems." Proceedings of the IEEE 104.5 (2016): 1039-1057.
5. Smith, Don C. "Cybersecurity in the energy sector: are we really prepared?." (2021): 265-270.
6. Morris, Thomas H., and Wei Gao. "Industrial control system cyber attacks." In 1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1, pp. 22-29. 2013.

7. Drias, Zakarya, Ahmed Serhrouchni, and Olivier Vogel. "Analysis of cyber security for industrial control systems." In 2015 international conference on cyber security of smart cities, industrial control system and communications (ssic), pp. 1-8. IEEE, 2015.
8. Maglaras, Leandros A., et al. "Cyber security of critical infrastructures." Ict Express 4.1 (2018): 42-45.
9. Nasir, Muhammad Ali, Shizra Sultan, Samia Nefti-Meziani, and Umar Manzoor. "Potential cyber-attacks against global oil supply chain." In 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), pp. 1-7. IEEE, 2015.
10. Al-Haija, Qasem Abu, and Swastik Brahma. "Optimization of Cyber System Survivability Under Attacks Using Redundancy of Components." In 2019 53rd Annual Conference on Information Sciences and Systems (CISS), pp. 1-6. IEEE, 2019.
11. Graham, James, Jeffrey Hieb, and John Naber. "Improving cybersecurity for industrial control systems." In 2016 ieee 25th international symposium on industrial electronics (isie), pp. 618-623. IEEE, 2016.
12. IEC: 62443-4-1. Security for industrial automation and control systems Part 4-1 Product security development life-cycle requirements (2018)
13. Haley, Charles B., Jonathan D. Moffett, Robin Laney, and Bashar Nuseibeh. "A framework for security requirements engineering." In Proceedings of the 2006 international workshop on Software engineering for secure systems, pp. 35-42. 2006.
14. Pandey, Dhirendra, Ugrasen Suman, and A. Kumar Ramani. "An effective requirement engineering process model for software development and requirements management." In 2010 International Conference on Advances in Recent Technologies in Communication and Computing, pp. 287-291. IEEE, 2010.
15. Mishra, Deepti, Alok Mishra, and Ali Yazici. "Successful requirement elicitation by combining requirement engineering techniques." In 2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), pp. 258-263. IEEE, 2008.
16. Fiorineschi, Lorenzo, et al. "Testing a new structured tool for supporting requirements' formulation and decomposition." Applied Sciences 10.9 (2020): 3259.
17. Gupta, Varun, et al. "Requirements engineering in software startups: A systematic mapping study." Applied Sciences 10.17 (2020): 6125.
18. Mengist, Alachew, Lena Buffoni, and Adrian Pop. "An Integrated Framework for Traceability and Impact Analysis in Requirements Verification of Cyber–Physical Systems." Electronics 10.8 (2021): 983.
19. Rehman, Shafiq Ur, and Volker Gruhn. "An effective security requirements engineering framework for cyber-physical systems." Technologies 6.3 (2018): 65.
20. Golosova, Julija, and Andrejs Romanovs. "The advantages and disadvantages of the blockchain technology." In 2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE), pp. 1-6. IEEE, 2018.
21. https://hyperledger-fabric.readthedocs.io/en/release-2.3/glossary.html, accessed August 2021
22. https://developer.ibm.com/technologies/blockchain/articles/blockchain-basics-hyperledger-fabric/, accessed August 2021.
23. Demi, Selina. "Blockchain-oriented requirements engineering: A framework." In 2020 IEEE 28th International Requirements Engineering Conference (RE), pp. 428-433. IEEE, 2020.
24. Demi, Selina, Ricardo Colomo-Palacios, and Mary Sánchez-Gordón. "Software Engineering Applications Enabled by Blockchain Technology: A Systematic Mapping Study." Applied Sciences 11.7 (2021): 2960
25. Woon, Irene MY, and Atreyi Kankanhalli. "Investigation of IS professionals' intention to practise secure development of applications." International Journal of Human-Computer Studies 65.1 (2007): 29-41.

26. Weider, D. Yu, and Kyle Le. "Towards a secure software development lifecycle with square+ r." In 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops, pp. 565-570. IEEE, 2012.
27. https://www.microsoft.com/en-us/securityengineering/sdl, accessed August 2021.
28. https://us-cert.cisa.gov/bsi/articles/best-practices/requirements-engineering/introduction-to-the-clasp-process, accessed August 2021.
29. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-64r2.pdf, accessed August 2021.
30. https://csrc.nist.gov/publications/detail/sp/800-160/vol-1/final, accessed August 2021.
31. Lipner, Steve. "The trustworthy computing security development lifecycle." In 20th Annual Computer Security Applications Conference, pp. 2-13. IEEE, 2004.
32. Kainerstorfer, Michael, Johannes Sametinger, and Andreas Wiesauer. "Software security for small development teams: a case study." In Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services, pp. 305-310. 2011.
33. Rindell, Kalle, Sami Hyrynsalmi, and Ville Leppänen. "Aligning security objectives with agile software development." In Proceedings of the 19th International Conference on Agile Software Development: Companion, pp. 1-9. 2018.
34. Gregoire, Johan, Koen Buyens, Bart De Win, Riccardo Scandariato, and Wouter Joosen. "On the secure software development process: CLASP and SDL compared." In Third International Workshop on Software Engineering for Secure Systems (SESS'07: ICSE Workshops 2007), pp. 1-1. IEEE, 2007.
35. Rindell, Kalle, Sami Hyrynsalmi, and Ville Leppänen. "Aligning security objectives with agile software development." In Proceedings of the 19th International Conference on Agile Software Development: Companion, pp. 1-9. 2018.
36. Roudiès, Ounsa. "Benchmarking SDL and CLASP lifecycle." In 2014 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14), pp. 1-6. IEEE, 2014.
37. Venson, Elaine, Xiaomeng Guo, Zidi Yan, and Barry Boehm. "Costing secure software development: A systematic mapping study." In Proceedings of the 14th International Conference on Availability, Reliability and Security, pp. 1-11. 2019.
38. Maier, Patrik, Zhendong Ma, and Roderick Bloem. "Towards a secure scrum process for agile web application development." In Proceedings of the 12th International Conference on Availability, Reliability and Security, pp. 1-8. 2017.
39. Thomas, Tyler W., Madiha Tabassum, Bill Chu, and Heather Lipford. "Security during application development: An application security expert perspective." In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1-12. 2018.
40. Hahn, Adam, Ali Tamimi, and Dave Anderson. "Securing your ics software with the attacksurface host analyzer (aha)." In Proceedings of the 4th Annual Industrial Control System Security Workshop, pp. 33-39. 2018.
41. Moyón, Fabiola, Daniel Méndez, Kristian Beckers, and Sebastian Klepper. "How to integrate security compliance requirements with agile software engineering at scale?." In International Conference on Product-Focused Software Process Improvement, pp. 69-87. Springer, Cham, 2020.
42. Walden, James, and Charles E. Frank. "Secure software engineering teaching modules." In Proceedings of the 3rd annual conference on Information security curriculum development, pp. 19-23. 2006.
43. Luburić, Nikola, et al. "A framework for teaching security design analysis using case studies and the hybrid flipped classroom." ACM Transactions on Computing Education (TOCE) 19.3 (2019): 1-19.
44. Mylrea, Michael, and Sri Nikhil Gupta Gourisetti. "Blockchain: Next generation supply chain security for energy infrastructure and nerc critical infrastructure protection (cip) compliance." Resilience Week 16 (2018).

45. Mylrea, Michael, Sri Nikhil Gupta Gourisetti, Randy Bishop, and Matt Johnson. "Keyless signature blockchain infrastructure: Facilitating nerc cip compliance and responding to evolving cyber threats and vulnerabilities to energy infrastructure." In 2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), pp. 1-9. IEEE, 2018.
46. Mylrea, Michael, and Sri Nikhil Gupta Gourisetti. "Blockchain for supply chain cybersecurity, optimization and compliance." In 2018 Resilience Week (RWS), pp. 70-76. IEEE, 2018.
47. Moyon, Fabiola, Kristian Beckers, Sebastian Klepper, Philipp Lachberger, and Bernd Bruegge. "Towards continuous security compliance in agile software development at scale." In 2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE), pp. 31-34. IEEE, 2018.
48. Dännart, Sebastian, Fabiola Moyón Constante, and Kristian Beckers. "An assessment model for continuous security compliance in large scale agile environments." In International Conference on Advanced Information Systems Engineering, pp. 529-544. Springer, Cham, 2019.
49. Moyón, Fabiola, Christoph Bayr, Daniel Mendez, Sebastian Dännart, and Kristian Beckers. "A light-weight tool for the self-assessment of security compliance in software development–an industry case." In International Conference on Current Trends in Theory and Practice of Informatics, pp. 403-416. Springer, Cham, 2020.
50. Nyaletey, Emmanuel, et al. "BlockIPFS-blockchain-enabled interplanetary file system for forensic and trusted data traceability." 2019 IEEE International Conference on Blockchain (Blockchain). IEEE, 2019.

Jelena Marjanović (née Stankovski) received her B.Sc. in 2015. and her M.Sc. in 2016 from the Faculty of Technical Sciences, University of Novi Sad. From 2016, she is a PhD student on Faculty of Technical Sciences, University of Novi Sad. Currently, she is a Teaching Assistant on the Faculty of Technical Sciences, University of Novi Sad, and a Senior Cybersecurity Analyst in the Energy Management Software Solutions Industry. Her research interests include blockchain, software engineering and cybersecurity.

Nikola Dalčeković received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Faculty of Technical Sciences, University of Novi Sad, in 2012, 2013, and 2019, respectively. Currently, he is a Product Security Officer in the Energy Management Software Solutions Industry. He worked as a Teaching Assistant in computer science with the Faculty of Technical Sciences, University of Novi Sad, from 2014 to 2020. At the same Faculty, he continued his career as an Assistant Professor from 2020 until 2022 when he decided to move to California, United States to help with industry innovation in the cybersecurity area. His research interests include cybersecurity, software engineering, and distributed computing.

Goran Sladić received the Ph.D. degree from the University of Novi Sad, in 2011. He is currently a Professor of computer science with the Faculty of Technical Sciences, University of Novi Sad. He has published over 80 articles and participated or lead in more than 20 projects. His research interests include cyber security, blockchain, software engineering, software architectures, and context-aware computing.

Received: September 23, 2021; Accepted: August 04, 2022.

Computer Science and Information Systems 20(1):381–404 https://doi.org/10.2298/CSIS220227061S

The Application of Machine Learning Techniques in Prediction of Quality of Life Features for Cancer Patients*

Miloš Savić[1], Vladimir Kurbalija[1], Mihailo Ilić[1], Mirjana Ivanović[1], Dušan Jakovetić[1], Antonios Valachis[2], Serge Autexier[3], Johannes Rust[3], and Thanos Kosmidis[4]

[1] Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad
Trg D. Obradovića 3, Novi Sad, Serbia
{svc, kurba, milic, mira, dusan.jakovetic}@dmi.uns.ac.rs
[2] Department of Oncology, Faculty of Medicine and Health, Örebro University
SE 70182 Örebro, Sweden
antonios.valachis@oru.se
[3] German Research Center for Artificial Intelligence GmbH, Cyber-Physical Systems
Bremen, Germany
{serge.autexier, johannes.rust}@dfki.de
[4] Care Across Ltd
London, England
thanos.kosmidis@careacross.com

Abstract. Quality of life (QoL) is one of the major issues for cancer patients. With the advent of medical databases containing large amounts of relevant QoL information it becomes possible to train predictive QoL models by machine learning (ML) techniques. However, the training of predictive QoL models poses several challenges mostly due to data privacy concerns and missing values in patient data. In this paper, we analyze several classification and regression ML models predicting QoL indicators for breast and prostate cancer patients. Three different approaches are employed for imputing missing values, and several settings for data privacy preserving are tested. The examined ML models are trained on datasets formed from two databases containing a large number of anonymized medical records of cancer patients from Sweden. Two learning scenarios are considered: centralized and federated learning. In the centralized learning scenario all patient data coming from different data sources is collected at a central location prior to model training. On the other hand, federated learning enables collective training of machine learning models without data sharing. The results of our experimental evaluation show that the predictive power of federated models is comparable to that of centrally trained models for short-term QoL predictions, whereas for long-term periods centralized models provide more accurate QoL predictions. Furthermore, we provide insights into the quality of data preprocessing tasks (missing value imputation and differential privacy).

Keywords: Quality of Life, Cancer Patients, Predictive Models, Federated Learning, Breast Cancer, Prostate Cancer

* This is an extended version of a MEDES 2021 conference paper [1].

1. Introduction

Nowadays in modern society people of all ages suffer from different chronic and critical diseases. The one of important trends in medical domains is to find out adequate services that will help patients to keep, or even if it is possible to increase, their quality of life and health parameters.. Information and Communication Technologies offers powerful tools, devices and services for collection of wide range of patient's data: digital health records, wearable devices, dashboards, unobtrusive sensors and other types of smart devices. Using advanced and effective techniques of Artificial intelligence (AI) and Machine learning (ML) [2] such data are processed and can be used to improve patient's health status and different quality of life (QoL) aspects [3, 4].

Predictive analytics in medicine and healthcare plays essential role and recently is more concentrated on obtaining personalized interventions, treatment and improvement of QoL issues. However, health QoL issues depend on type of disease and patients status (beginning disease phase, after surgery, during treatment, follow-ups, etc.) and in this paper we concentrate of various QoL issues for breast and prostate cancer such as depression, anxiety, insomnia, etc.

Research results presented in the paper are achieved within ASCAPE project: Artificial intelligence supporting cancer patients across Europe (https://ascape-project.eu/) where two the most prevalent types of cancer are considered: breast and prostate cancer. One of the main purposes of the project is to employ powerful AI/ML mechanisms to support cancer patients' health status and QoL.

Apart from patients' data collected from different sources like health records, wearable devices, environmental data, within ASCAPE also different validated questionnaires will be used to capture the particular QoL issues depending on type of cancer. Such questionnaires are good instruments to minimize the risk for measurement bias, and to enable the better ASCAPE ability for suitable predictions and interventions. Clinical ASCAPE partners identified 15 QoL issues for breast and 12 for prostate cancer that will be predicted through AI-based models.

To the best of our knowledge, ASCAPE is a unique research project that prospectively investigate an AI-based approach, towards a personalised follow-up strategy for cancer patients focusing on their QoL issues. The aim of the project is to offer substantial benefits for after-treatment health-related QoL improvements.

Importance of applying adequate approaches to properly process huge amounts of sensitive patients' data is evident in different medical disciplines and areas [5–8]. Two approaches to train predictive models are prevalent: centralized and distributed. In the centralized approach patients' data collected from different sources are stored locally in clinics and model training is performed locally as well. However, recently popular decentralized ML technique is Federated learning (FL) which enables the quality use and learning from decentralized data. Patients' data collected by clinics could be adequately processed by applying FL approach as it enables the training of shared global models with a central server, while keeping all the sensitive data in local. Moreover, raw data could not be used directly for ML model training since it needs some additional activities for curation and pre-processing. In this paper we will examine the importance and the impact of two pre-preocesing techniques which are required with existing form of data: missing value imputation (MVI) and differential privacy (DP). The primary purpose of

FL in ASCAPE is to enable democratized access to ML models promoting patient QoL without revealing private or sensitive patient data.

Presented experimental results are based on already existing retrospective datasets of an ASCAPE clinical partner. The aim of conducted experiments was to identify appropriate ML models for breast and prostate cancer patients. However, numerous ML techniques which are utilized during project showed to be very promising and we will continue our research activities on prospectively collected data expecting accurate and reliable behavior of developed models. The experimental evaluation of the ML models presented in this paper is based on datasets containing QoL indicators derived from prescribed medications and LISAT-11 questionnaires. As far as we are aware, this is the first analytical study of various ML models predicting such QoL features. Another important novelty of our study is that the selected ML models are not examined only in a centralized learning scenario, but also in two different federated learning settings. Medical datasets usually contain missing values, while ensuring privacy during predictive model training is of the uttermost importance. Thus, as the last contribution of our paper, we study the impact of appropriate pre-processing techniques (MVI and DP) to the quality of investigated ML models.

The rest of paper is organized as follows. The second Section is devoted to related work. The central, third Section, is focused on application of ML techniques for predictive QoL models. Section four is devoted to the description of experimental datasets. Experimental results are discussed in Section five. The last Section brings concluding remarks.

2. Related Work

The tracking and monitoring of QoL parameters has attracted many research attention recently. This has a particular importance in chronic diseases for the prevention and early detection of symptoms and signs. The proper monitoring should provide the positive effect on patient's quality of life, economic impact and resource management. As expected, substantial amounts of data on quality of life are combined into clinical trials using a variety of instruments [9].

Regular assessment of QoL parameters of chronic patients has an impact on physician-patient communication and result in benefits for some patients, who had a better QoL [5]. Furthermore, continuous collection of medical attributes and growth of the database could allow the selection of proper medical variables and the selection of adequate models for a more accurate prediction of QoL [9].

Several recent studies emphasized the importance of applying AI and data mining (DM) techniques in successful prediction of QoL issues. For example, in [10] authors used QoL issues to monitor patients' clinical condition and health status for the patients with ultra-rare autosomal recessive disease Alkaptonuria. One of the main problems in the follow-up of patients with ultra rare diseases is the lack of a standardized methodology to assess disease severity or response to treatment and QoL scores could be a successful way to monitor such patients. Also, different machine learning approaches (Linear Regression, Neural networks and k-nearest-neighbor) were implemented with the aim to perform a prediction of QoL scores based on clinical data.

The authors in [3] present the basis of the DIAL system which represents an early warning system for the early detection of a deteriorating QoL score in the hemodialysis population using machine learning algorithms. Here two models (classification tree and Naïve Bayes) were generated to predict an increase or decrease of 5% in a patient's QoL score over one month. The classification tree was selected as the better model with an area under curve (AUC) of 83.3% and accuracy of 81.9%. The authors concluded that their system DIAL, if implemented on a larger scale, is expected to help patients in terms of ensuring a better QoL and a reduction in the financial burden in the long term.

The predictor of quality of life called the Better Life Index (BLI) was used for QoL assessment in [11]. It is based on the measurement of different aspects of human life in the whole population: environment, jobs, health, civic engagement, governance, education, access to services, housing, community, and income. The paper presents a supervised machine-learning analytical model that predicts the life satisfaction score using several DM models like: decision tree, elastic net, neural network, random forest, support vector machine, etc. The results showed that the ensemble model based on the stacked generalization framework is a significantly better predictor of the life satisfaction of a nation, compared to base models.

QoL are studied in different cancer related diseases since they present a common chronic conditions nowadays. For example, in [12] authors presented the project which tries to design a new patient reported outcome measure to assess QoL issues for patients with locally recurrent rectal cancer. The authors identified the fact that it is very tedious to administer simultaneously several questionnaires required for QoL assessment especially in patients with major limitations caused by the disease. So it is essential to find alternatives which largely replace questionnaires, and their approach involves use of biometric devices with the help of DM techniques.

Some successful applications of QoL prediction in cancer diseases also includes: the comparison of the pre and the post treatment quality of life in cervical cancer patients [13], construction of machine learning and statistical models for prediction of gastro-urinary symptoms and QoL issues following prostate radiation treatment [14], and the prediction of 5-year lung cancer survival on the basis of QoL issues [4].

To the best of our knowledge, this study is the first one which tries to develop a QoL prediction model for breast and prostate cancer patients. The study [14] also investigates prostate cancer patients but focus only on one type of intervention – radiation treatment. Here we will try to encompass both types of cancer patients (breast and prostate) with as many as possible common attributes, with all types of common interventions, and with a set of relevant QoL attributes.

3. Machine Learning Techniques for Predictive QoL Models

The QoL indicator of a cancer patient can be predicted either by a classification or a regression machine learning model depending on the type of the indicator. In this work we consider two types of QoL indicators: (1) binary indicators indicating whether the patient will experience QoL related symptoms after diagnosis (e.g., anxiety and depression) and (2) numeric indicators indicating overall QoL of the patient reported by filling an appropriate questionnaire (e.g., the QoL score of the LISAT-11 questionnaire [15]). Binary

classification models are the most appropriate for the first type of QoL indicators, while for the second type of QoL indicators predictions can be obtained by regression models.

For predictive QoL models (as well as for medical predictive models in general) it is important to distinguish two practical scenarios: centralized and decentralized (federated). In the centralized scenario all anonymized training data coming from different medical organizations is collected to a central place where machine learning models are trained. However, in majority of cases medical organizations are not willing to share anonymized patient data, or it is not even allowed for them to do so according to governmental regulations and laws. One solution in this case is to apply federated learning [16] of predictive QoL models in which several edge nodes collectively train predictive QoL models by exchanging locally updated machine learning models instead of sharing training datasets. Edge nodes are computational devices owned by medical organizations and deployed within their computational infrastructure, so training datasets never leave the boundaries of data owners.

Before training AI models, the available datasets must be pre-processed. Some pre-processing steps are mandatory, while the others only improve the predictive power or security/privacy of trained models. In this paper we will consider two pre-processing approaches: missing values imputation (MVI) and differential privacy (DP). MVI is obligatory step since the model training can not be performed with incomplete data. DP is optional since its role is to protect the privacy of patients data.

3.1. Centralized QoL Predictive Models

The following machine learning algorithms are examined for training centralized binary classification models predicting binary QoL indicators: NB (Naive Bayes), kNN (k nearest neighbors), SVM (support vector machines), DT (decision trees) and RF (random forests).

NB is a probabilistic classification algorithm learning a predictive model giving the most probable class (positive or negative in the case of binary classification) for a given data instance (in our case a patient described by a set of features). Class probabilities are computed using conditional probability estimates learned from a training dataset under the assumption that features describing data instances are conditionally independent.

kNN is a lazy learning algorithm. The class for a given data instance is predicted by majority voting from the classes of the k closest data instances belonging the training dataset according to some distance functions (e.g., the Euclidean or Manhattan distance).

SVM classifiers are based on the idea of using linear models to identify non-linear boundaries of classes. This is achieved by transforming data instances into a new higher dimensional space using a non-linear mapping. Quadratic programming algorithms are then employed in the higher dimensional space to determine the maximum margin hyperplane separating instances from different classes.

Decision-tree based classifiers make predictions according to decision trees constructed from the training dataset by a recursive divide-and-conquer algorithm utilizing some information theoretic measure (e.g, information gain or the Gini impurity). This means that decision trees are formed in the divisive manner from the root of the tree to its leafs. The underlying information theoretic measure is used to find and select the most discriminative feature to form a corresponding node in the tree and split training data to recursively form its subtrees.

A random forest is an ensemble of decision trees learned from bootstrapped samples of the training dataset. The RF algorithm employs a feature bagging procedure to determine a random subset of features for learning individual decision trees. The class for a given input data instance is then determined as the most frequent class predicted considering all decision trees in the ensemble.

For predicting numeric QoL indicators we examine the following algorithms for learning regression models: LINEAR (linear regression), RIDGE (ridge regression), LASSO (lasso regression), ELASTICN (elastic net regression), KRIDGE (kernel ridge regression), SVM (regression by support vector machines), RF (regression by random forests), and kNN (k-nearest neighbours regression).

The linear regression algorithms determine coefficients of a linear model by minimizing the residual sum of squares (RSS) between real values of the target variable and predictions derived from the model. Ridge, Lasso and Elastic Net find linear models by minimizing RSS with incorporated regularization penalties: Ridge incorporates the L2 regularization penalty, Lasso is based on the L1 regularization penalty, while Elastic Net uses both previously mentioned penalties. Kernel Ridge regression performs Ridge regression in a space obtained by a non-linear mapping of the training dataset. SVM, RF and kNN are adaptations of the corresponding classification algorithms for regression tasks.

In our experiments we have used the Scikit-learn machine learning library [17] to develop a set of Python modules for training centralized QoL predictive models.

3.2. Federated QoL Predictive Models

A federated model is a machine learning model collectively trained by several edge nodes running federated learning clients. Each federated learning client has its own dataset for training the model and those local training datasets are never exchanged among federated learning clients participating in federated learning. The federated learning process is coordinated by a federated learning server. The main purpose of the federated learning server is to enable the exchange of locally updated federated models among federated learning clients.

Two basic federated learning schemas are incremental and concurrent. Let C_1, C_2, .., C_k denote k federated learning clients each having its own training dataset D_i ($i \in [1..k]$). In the incremental federated learning scheme federated learning clients incrementally build a machine learning model from the first to the last client. This means that C_1 creates M on D_1 and sends it to the federated learning server. Then, C_2 retrieves M from the federated learning server, updates it on D_2 and returns the updated model back to the server. Each next federated learning client does exactly the same until the last client C_k.

In the concurrent federated learning scheme M is collectively trained in parallel. In the first step each federated learning client C_i creates its own model M_i on D_i. All models are then sent to the federated learning server which averages M_1 to M_k into a single model M. Once all edge nodes have submitted their local models, the global model is updated using the Federated Averaging [18] approach. The federated learning server then sends M to all federated learning clients which update M on their local datasets and the updated models are returned back to the federated learning server for the second averaging. The previous operation is repeated for an arbitrary number of learning rounds and the averaged model after the last learning round is the final federated model.

Neural networks are the most natural model choice for federated learning for the following two reasons: (1) neural networks can be incrementally updated, and (2) neural networks can be easily averaged by averaging edge weights and biases. Since we deal with two types of predictive problems we also have two types of federated neural networks:

1. Federated neural networks for regression. The last layer of such neural networks contains exactly one node activated by the linear function. In our work we use the mean squared error (MSE) as the loss function when training regression neural networks.
2. Federated neural networks for binary classification. In this case, one node contained in the last layer is activated by the sigmoid function. Output values higher than 0.5 indicate the positive class, while values lower than 0.5 correspond to the negative class. The binary cross-entropy function is used at the loss function when optimizing parameters of binary classification neural networks.

Nodes in hidden layers of both types of federated neural networks are activated by the ReLU activation function. We also consider two mechanisms to prevent overfitting: dropout and regularization strategies (the kernel, bias and activation regularization).

To compare federated QoL models to centralized QoL models we have developed a federated learning simulator based on the Tensorflow machine learning library [19]. The realized simulator supports both incremental and concurrent federated learning mode for an arbitrary number of simulated edge nodes (federated learning clients). The architecture of a federated neural network can be specified by providing its type (regression or binary classification), the number of hidden layers and the number of nodes per hidden layer. The user can also specify the number of epoch (learning rounds) and the batch size (the number of training instances propagated through the network when updating model parameters). At the beginning, the simulator divides training data (training folds when the k-fold cross-validation is applied to evaluate models) into p stratified parts, where p is the number of simulated edge nodes. Then, simulated edge nodes use their part of training data when creating or updating Tensorflow neural networks by the Adam optimization algorithm [20]. In the case of concurrent federated learning, the averaging of Tensorflow neural networks formed by simulated edge nodes is performed after each epoch. More precisely, we utilized the callback mechanism provided by Tensorflow to implement the federated client-server communication. A socket connection is open between an edge node and the coordinating server once a client wishes to update a global model, and at key training steps such as the end of each epoch, the edge node sends its model updates to the server.

3.3. Missing Value Imputation (MVI)

Datasets for training machine learning models often contain missing values. This is also the case with the experimental datasets used in this work. To train machine learning models it is necessary to infer and fill missing values of predictor features or, alternatively, to remove data instances containing missing values. For the MVI process we use three methods provided by the Scikit-learn library: (1) simple MVI, (2) iterative MVI and (3) nearest neighbours imputation.

The simple MVI fills missing values using a simple approach: all missing values for a feature f are filled with the mean of existing values in f.

The iterative MVI is based on the idea to train a regression model for each feature containing missing values [21]. The regression model for feature f is trained based on values of other predictor features. Then, missing values for f are filled based on predictions of its regression model. After obtaining predictions for all missing values, the iterative MVI repeats the whole procedure for predefined number of times in a round-robin fashion (i.e., predicted missing values of f together with known values are then used to retrain the regression model for other features also containing missing values). The predictions of the final round are then used to fill missing values. It is important to emphasize here that we do not use a datasets' target attribute for the training of regression model. If the target attribute was used, the regression models of the iterative imputer would be fit with target values, which must not happen since it would introduce erroneous dependence between predictors and target attribute.

In nearest neighbours imputation a Euclidean distance metric is used to find the nearest neighbours. Each missing feature is imputed using values from k (parameter of the method) nearest neighbours that have a value for the feature. The features of the neighbours could be averaged uniformly or weighted by distance to each neighbour.

3.4. Differential Privacy (DP)

The task of DP [22] is to preserve the privacy of patients data and to prevent model inversion attacks [23]. Using these attacks an attacker can reconstruct some of the training data from AI models (under particular circumstances). Differential privacy is based on the idea that the outcome of the query posed to protected database (or machine learning model) is essentially equally likely independent of whether any individual joins or refrains from joining the database. In such a way the private data about a particular patient is protected since the system returns the result with the same probability whether a particular patient was involved in the analysis or not.

There are several mechanisms to implement DP, but the most common are: Laplace mechanism, Gaussian mechanism and Exponential mechanism. Laplace mechanism is most commonly used for numeric types of data, and it consists of adding Laplacian noise (a noise which follows Laplace distribution) to the data model. Furthermore, in machine learning algorithms noise could be added in different ways: (1) to the training dataset, (2) to the prediction model itself (for example in edge weights in neural network), and (3) to the predicted results of the model. Here, first option will be applied as the simplest one, but still sufficiently general and robust.

As expected, more noise means more privacy, but a lot of noise could lead to the reduction of model performance/accuracy. So, the main challenge in introduction of privacy preserving techniques in machine learning models is the optimal balance between privacy and the utility of models. The amount of privacy is controlled through parameter ϵ. Smaller value of ϵ means stronger privacy, which is achieved by adding more noise to the training data. The value 0 represents total privacy, but the usability of such an algorithm is none since in that case training data represents pure randomness.

3.5. Model Evaluation

To estimate errors of the examined regression models we use the mean absolute error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}|, \tag{1}$$

where n is the number of data instances in a test dataset (or a test fold), y_i the real value of the target variable for i-th instance and $\hat{y_i}$ is the predicted value of the target value according to a regression model.

In our experimental evaluation of QoL regression-based predictive models, MAE estimates of the examined regression models are also compared to MAE estimates of the so-called DUMMY regression algorithm. The DUMMY regression algorithm always predict the same value for a target feature: the mean of the target feature from a training dataset (or training folds).

Binary classifiers predicting QoL issues are evaluated using accuracy, precision, recall and F_1 scores. The accuracy of a binary classifier is equal to the number of correctly classified instances divided by the total number of instances in a test dataset (or a test fold). Precision and recall metrics are defined per class. The precision for a class c (c is the positive or negative class) is the number of instances correctly classified to c divided by the total number of instances classified to c. On the other hand, the recall for c is the number of instances correctly classified to c divided by the total number of instances that belong to c. Precision and recall score for the whole binary classifier are obtained by averaging precision and recall scores per class. Since precision and recall measure two different aspects of classifier's performance it is useful to aggregate them into a single score. The usual way to aggregate precision and recall is to compute the F_1 score which is the harmonic mean of precision (P) and recall (R):

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}. \tag{2}$$

All above-mentioned model evaluation metrics are estimated by the 10-fold cross validation procedure. In the 10-fold cross-validation, an input dataset is split into 10 folds. Then, 10 iterations of training and evaluation are performed. In the i-th iteration, i-th fold is used to compute evaluation metrics of models trained on the remaining folds. In this way we obtain 10 estimates of evaluation metrics that are then averaged into final estimates.

The effects of missing value imputation to the performance of predictive QoL models is investigated by comparing models trained on datasets to which the simple MVI is applied, to models trained on datasets obtained after the iterative MVI and after the MVI based on kNN rule. The impact of differential privacy is measured by comparing models trained without and with applied differential privacy with different values of parameter ϵ.

The evaluation methodology described in this section may also be utilized by practitioners for selecting the best predictive ML model for any other medical dataset with numeric or discrete target variables. After applying MVI and DP techniques to a dataset, each ML model from a list of candidates is validated by the 10-fold cross validation that results with a metric indicating the overall predictive ability of the model. In case of numeric target variables, the regression-based model with the lowest MAE is the best one.

On the other hand, the model with the largest F_1 is the best performing model in the case of discrete target variables.

4. Experimental Datasets

Previously described machine learning models are evaluated on datasets formed from two databases: ORB and BcBase. BcBase is a population-based research database containing data about early breast cancer patients from three healthcare regions in Sweden (accounting for nearly 60% of the total population) in terms of patient and tumor characteristics, treatment strategies and prescribed medications. The database does not include QoL indicators captured through questionnaires, but from prescribed medications it is possible to derive the presence of certain QoL issues. From BcBase we formed 4 datasets for training and evaluating binary classifiers deciding whether a patient will suffer from anxiety (BcBase-Anxiety), depression (BcBase-Depression), insomnia (BcBase-Insomnia) and pain (BcBase-Pain) after breast cancer treatment. All BcBase datasets contain 18988 data instances (patients) described by 97 predictor features.

ORB is a database containing data about patients with localised prostate cancer treated with radiotherapy in the Örebro healthcare region. It includes data on patient and tumour characteristics, treatment approaches, dosimetric parameters regarding radiotherapy, side effects based on direct questions or validated questionnaires (IPSS, IIEF-5) and QoL issues based on a validated questionnaire (LISAT-11). ORB contains 2466 health records with follow-up examinations repeated at six months intervals and LISAT-11 QoL scores at the time of the diagnosis and three different times relative to the date of diagnosis at months 36, 60 and 120. From ORB data we created six datasets with the naming scheme ORB-n-m for training regression models predicting the LISAT-11 QoL score at month m considering all patient data collected up to month n. The created datasets are: ORB-30-36, ORB-30-60, ORB-30-120, ORB-54-60, ORB-54-120 and ORB-108-120. For example, ORB-30-36 is used to train regression models predicting the LISAT-11 QoL score in month 36 considering data collected up to month 30 as predictor variables. The number of instances (patients) in ORB datasets ranges from 1138 in ORB-30-36 to 610 in ORB-108-120, while the number of predictor variables ranges from 96 in ORB-30-36 to 158 in ORB-108-120 (the number of patients decreases due to recoveries, dropouts and deaths, while the number of predictors increases due to longer time intervals. Also some patients got their diagnosis less than 10 years ago. They haven't had their follow-up yet).

5. Results and Discussion

In this section we first present the evaluation of centralized machine learning models trained on experimental datasets described in previous section. The influence of missing value imputation and differential privacy to the performance of those models is then discussed. Finally, we examine federated QoL machine learning models and compare them to their centralized counterparts.

5.1. Evaluation of Centralized QoL Models

The performance of binary classification models trained on BcBase datasets obtained by the 10-fold cross-validation are summarized in Tables 1, 2, 3, and 4. All classifiers were

trained and evaluated on BcBase datasets obtained after missing value imputation by the iterative MVI method.

Table 1. Evaluation of binary classification models on BcBase-Anxiety

Classifier	Accuracy	Precision	Recall	F_1
RF	0.673	0.484	0.535	0.514
SVM	0.698	0.411	0.349	0.5
NB	0.629	0.552	0.553	0.552
KNN	0.682	0.458	0.529	0.507
DT	0.583	0.511	0.511	0.511

Table 2. Evaluation of binary classification models on BcBase-Depression

Classifier	Accuracy	Precision	Recall	F_1
RF	0.677	0.473	0.524	0.509
SVM	0.702	0.413	0.351	0.5
NB	0.566	0.534	0.543	0.551
KNN	0.688	0.462	0.536	0.509
DT	0.589	0.515	0.515	0.515

Table 3. Evaluation of binary classification models on BcBase-Insomnia

Classifier	Accuracy	Precision	Recall	F_1
RF	0.538	0.526	0.535	0.532
SVM	0.541	0.533	0.539	0.537
NB	0.555	0.554	0.555	0.554
KNN	0.521	0.502	0.516	0.514
DT	0.516	0.515	0.515	0.515

The SVM classifier exhibits the highest accuracy on three out of four BcBase datasets (BcBase-Anxiety, BcBase-Depression and BcBase-Pain). However, precision and recall scores of SVM on those three BcBase datasets are significantly lower compared to other classifiers. A more detailed examination of precision and recall per class revealed that SVM has zero precision and zero recall for the positive class (the presence of anxiety,

Table 4. Evaluation of binary classification models on BcBase-Pain

Classifier	Accuracy	Precision	Recall	F_1
RF	0.698	0.486	0.554	0.518
SVM	0.714	0.417	0.357	0.5
NB	0.53	0.517	0.553	0.564
KNN	0.699	0.457	0.528	0.506
DT	0.604	0.522	0.522	0.522

depression and pain after treatment) and that it dominantly predicts the negative class (no negative QoL related symptoms). The highest accuracy of SVM on those three dataset is a consequence of its bias towards the negative class on class imbalanced datasets in which approximately 70% of the patients belong to the negative class and 30% to the positive class. Therefore, it can be concluded that accuracy is not an appropriate measure for comparing binary classification models trained on BcBase datasets. F_1 score is more adequate measure since it takes into account precision and recall of both classes. It can be observed that SVM exhibits the lowest F_1 score on those datasets where it has the highest accuracy. KNN has the lowest F_1 score on BcBase-Insomnia and the second lowest on other BcBase datasets. Thus, it can be concluded that those two methods are the worst performing binary classification models for BcBase datasets.

The largest F_1 score on three BcBase datasets (anxiety, depression and insomnia) is achieved by the NB classifier. The best model for the fourth dataset (pain) is DT, but its F_1 score is very close to the F_1 score of NB. Consequently, it can be concluded that NB is the best choice to train centralized QoL predictive models for BcBase datasets.

The results of the evaluation of centrally trained regression models on the ORB datasets are shown in Table 5 including the DUMMY regressor as the baseline. The best model (the lowest MAE) for ORB-30-36 is RF. For the rest of ORB datasets, the best performing model is LASSO. KNN is the worst performing regression algorithm on all ORB datasets: predictions made by this model are even more erroneous than predictions made by DUMMY. Excluding kNN, all others considered models exhibit smaller prediction errors compared to DUMMY except in one case: DUMMY is better than linear regression on ORB-30-120. The prediction errors of the best performing model are in the range [4.84, 6.47], which is an acceptable level of prediction errors taking into account that the target variable (the LISAT QoL index) is in the range [11, 66].

The improvement of LASSO (the best performing regression model) over DUMMY are significant for short term QoL predictions (30-36, 54-60, 108-120) when the reduction of the MAE score ranges from 20% to 30%. For medium term QoL predictions (30-60, 54-120) the improvements are between 10% and 15%. As expected, the lowest improvement is for long term QoL predictions on ORB-30-120 where the reduction of MAE scores is slightly higher than 5%.

Table 5. MAE scores of regression models on ORB datasets (best value per column is bolded)

Regressor	30-36	30-60	30-120	54-60	54-120	108-120
DUMMY	6.541	6.89	6.909	6.89	6.909	6.909
LINEAR	5.311	6.129	7.003	5.238	6.899	6.524
RIDGE	5.1	5.925	6.652	5.07	6.356	5.977
LASSO	5.089	**5.886**	**6.478**	**4.84**	**6.18**	**5.437**
ELASTICN	5.126	5.913	6.504	4.859	6.216	5.448
KRIDGE	5.147	5.958	6.75	5.115	6.492	6.155
SVM	6.519	6.773	6.871	6.772	6.859	6.875
RF	**5.051**	6.015	6.685	5.009	6.357	5.635
KNN	6.72	6.968	7.133	6.906	7.128	7.033

5.2. Influence of Missing Value Imputation

To evaluate the impact of MVI and DP to the accuracy of the examined ML models we focus on one dataset – ORB-30-36[5]. The following MVI techniques are applied to ORB-30-36 in order to obtain a complete dataset with missing values: (1) Simple imputer, (2) Iterative imputer, and (3) Imputer base on kNN rule. The value of parameter k in kNN imputer is varied from 1 to 10.

After imputation of missing values all regression models described in Section 3.1 are trained using datasets with imputed values. The performance of these models is measured using MAE regression evaluation metrics.

Table 6 shows the MAE values for all 3 imputer approaches, and for 10 values of the parameter k in kNN approach. Since there are a lot of regression methods, table is split on two sub-tables.

In order to better visualize the impact of MVI techniques, the values of MAE of two most promising regressors (LASSO and ElasticNet) are plotted in Figure 1 and Figure 2. The presented behaviour is similar to all regressors, although it is not the same in all circumstances. Considering kNN approach, better values (smaller MAE) are achieved for higher values of k. Simple and iterative imputers are usually in the range of lower MAE values of kNN approach. In some cases, simple imputer is better, while in some other cases iterative imputer shows better performance.

Since the differences are quite small, and since the simple imputer is significantly faster than other two, it can be concluded that the choice of the MVI method does not make a significant influence to the performance of examined models and that the simple MVI can be as equally effective as the iterative or kNN MVI.

5.3. Influence of Differential Privacy

Our experimental datasets contain missing values. Consequently, it is first necessary to impute missing values before applying differential privacy. The experimental results presented in the previous section shows that the choice of MVI imputer does not significantly

[5] The results of MVI and EP evaluation for other datasets are highly similar to the results obtained on ORB-30-36.

Table 6. MAE values for different MVI techniques and for different regressors on ORB-30-36 dataset

MVI	Linear	Ridge	Lasso	ElasticNet	KernelRidge
simple	5.3040	5.2366	5.0814	5.1084	5.2619
iterative	5.2494	5.1141	5.1344	5.1912	5.1501
knn_1	5.4895	5.4254	5.3072	5.3204	5.4575
knn_2	5.3644	5.3084	5.1617	5.1774	5.3374
knn_3	5.2343	5.1727	5.1313	5.1435	5.1998
knn_4	5.2735	5.2117	5.1567	5.1619	5.2364
knn_5	5.2565	5.1927	5.1525	5.1695	5.2207
knn_6	5.2625	5.1948	5.1038	5.1134	5.2306
knn_7	5.2732	5.2021	5.1201	5.1132	5.2353
knn_8	5.2619	5.1885	5.0952	5.0958	5.2258
knn_9	5.2584	5.1773	5.0950	5.1004	5.2179
knn_10	5.2682	5.1836	5.0807	5.0873	5.2240

MVI	SVM	RF	KNN	AdaB	TFNN
simple	6.5197	5.1142	6.4550	5.5006	6.0614
iterative	6.5000	5.1079	6.7064	5.3787	6.0978
knn_1	6.5110	5.2839	6.6407	5.6586	6.3338
knn_2	6.5164	5.1627	6.7020	5.4869	6.0745
knn_3	6.5148	5.1237	6.7146	5.4681	6.0500
knn_4	6.5200	5.1076	6.7535	5.5114	6.1146
knn_5	6.5208	5.1107	6.7520	5.4943	5.9661
knn_6	6.5148	5.1442	6.7349	5.4876	6.6542
knn_7	6.5183	5.0847	6.7675	5.4033	5.9290
knn_8	6.5201	5.1352	6.7311	5.4286	5.8820
knn_9	6.5205	5.1653	6.7735	5.3508	5.7765
knn_10	6.5204	5.1455	6.7483	5.3892	6.0941

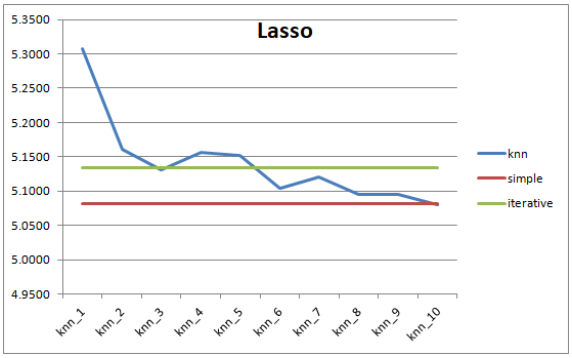


Fig. 1. MAE values for Lasso regressor for different MVI techniques

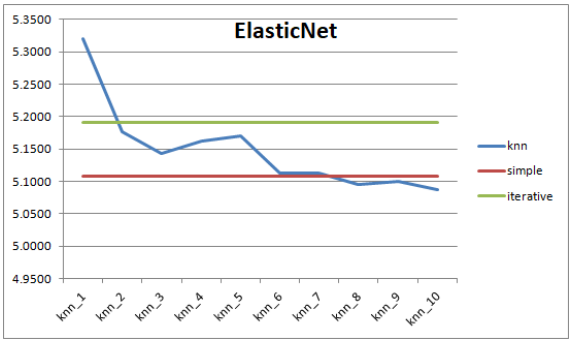


Fig. 2. MAE values for ElasticNet regressor for different MVI techniques

influence the predictive ability of the ML models. Thus, in the experimental evaluation of DP, we impute missing values in the ORB-30-36 dataset by the simple imputer. Then, the dataset without missing values is treated with the DP component which adds a specified amount of noise. The amount of noise is expressed through the ϵ parameter. The noise is added only on numerical features while the categorical features, Boolean features, one-hot-encoding features and class features stayed untouched. The following values of DP parameter (ϵ) were used for initial experiments: $\{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$. These initial experiments should provide an insight in the behaviour of ML models with regard to the amount of added noise.

All obtained results for all ML regression models behave in a similar manner. These results could be summarized as shown in Figure 3 and Figure 4 (here the results for the most prominent regressors LASSO and RF are shown). For a very small value of ϵ (e.g., 0.1), the MAE is notably high. After that, MAE significantly drops with an increase of ϵ up to the value of approximately 1. For the values of $\epsilon > 1$, MAE remains approximately constant on the level of MAE of models trained with no DP data (baseline level).

Since the lower ϵ parameter means more privacy, the goal here was to find the lowest value of ϵ with satisfactory MAE value. The conclusion from this preliminary set of experiments is that the value of ϵ should be in the neighborhood of value 1. For the values of

$\epsilon > 1$ the models do not have significantly higher performance (do not have lower MAE), and the level of privacy is decreased.

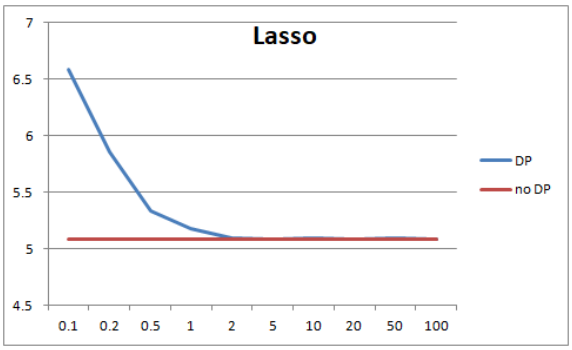


Fig. 3. MAE values for Lasso regressor for different values of DP parameter.

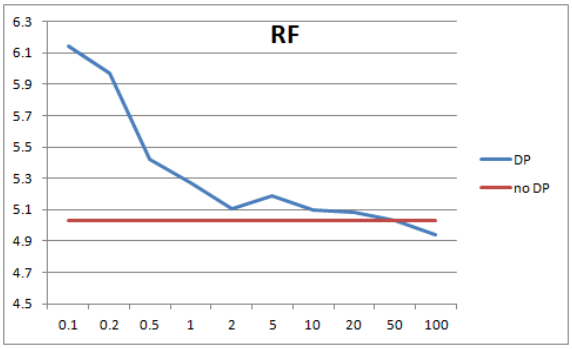


Fig. 4. MAE values for RF regressor for different values of DP parameter.

Since the privacy of the patient data is particularly important we performed additional experiments to find an optimal value of ϵ (lowest value of ϵ with satisfactory MAE value of regressors). We tested the performance of all regressors with the following values of ϵ (with finer granularity around value 1): $\{0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2\}$. The results are shown in Table 7.

The MAE values of the most prominent regressors are plotted in Figure 5 and Figure 6 (LASSO and RF). It is evident that in this interval of ϵ values there are no significant changes in trend of regressors' performance. Same behaviour is also observed in case of other regressors. However, as already mentioned, we are interested in the smallest value of ϵ with satisfactory MAE value. From previous experiments we know that the MAE value steadily drops for the small values of ϵ. And now it is evident from these results that this constant drop-down is present from $\epsilon = 0.1$ to $\epsilon = 0.7$ which is applicable for all

Table 7. MAE values for different values of DP parameter and for different regressors on ORB-30-36 dataset.

EPS	Linear	Ridge	Lasso	Elastic Net	Kernel Ridge
0.5	5.4861	5.4378	5.3345	5.3501	5.4648
0.6	5.4210	5.3804	5.2231	5.2518	5.4001
0.7	5.3296	5.2879	5.1340	5.1646	5.3153
0.8	5.3002	5.2633	5.2317	5.2624	5.2781
0.9	5.3366	5.2832	5.1960	5.2313	5.3093
1	5.2954	5.2535	5.1809	5.2119	5.2568
1.1	5.5842	5.5404	5.4678	5.4899	5.5640
1.2	5.2935	5.2500	5.1456	5.1799	5.2737
1.3	5.5551	5.5114	5.4355	5.4589	5.5350
1.4	5.2051	5.1577	5.0898	5.1276	5.1847
1.5	5.3807	5.3366	5.2439	5.2728	5.3582
1.6	5.3220	5.2722	5.1698	5.2036	5.2958
1.7	5.3602	5.3156	5.2295	5.2603	5.3371
1.8	5.3633	5.3184	5.2390	5.2699	5.3392
1.9	5.3696	5.3239	5.2398	5.2706	5.3454
2	5.2583	5.2039	5.0817	5.1219	5.2276
NO DP	5.3109	5.1000	5.0895	5.1259	5.1474

EPS	SVM	RF	KNN	AdaB	TFNN
0.5	6.5186	5.4221	6.7054	5.5476	6.3555
0.6	6.5186	5.2985	6.7123	5.4608	6.3131
0.7	6.5186	5.2177	6.7159	5.4039	6.7360
0.8	6.5186	5.3210	6.7152	5.5178	6.2085
0.9	6.5186	5.1730	6.7250	5.4338	6.0307
1	6.5186	5.2733	6.7239	5.4399	6.1016
1.1	6.5186	5.4779	6.7139	5.6432	6.3989
1.2	6.5186	5.2679	6.7204	5.4214	6.2083
1.3	6.5186	5.4569	6.7145	5.6210	6.3799
1.4	6.5186	5.1505	6.7270	5.3515	5.8816
1.5	6.5186	5.3059	6.7173	5.4841	6.2614
1.6	6.5186	5.2953	6.7212	5.4282	5.8575
1.7	6.5186	5.2939	6.7194	5.4745	6.2064
1.8	6.5186	5.3016	6.7198	5.4815	6.1535
1.9	6.5186	5.2996	6.7202	5.4779	6.1480
2	6.5186	5.1055	6.7194	5.4234	5.7977
NO DP	6.5186	5.0332	6.7200	5.3623	6.0351

regressors. After value $\epsilon = 0.7$ the MAE value starts to fluctuate without no significant trend of decrease. Therefore, the value 0.7 is imposed as the appropriate choice for DP parameter ϵ.

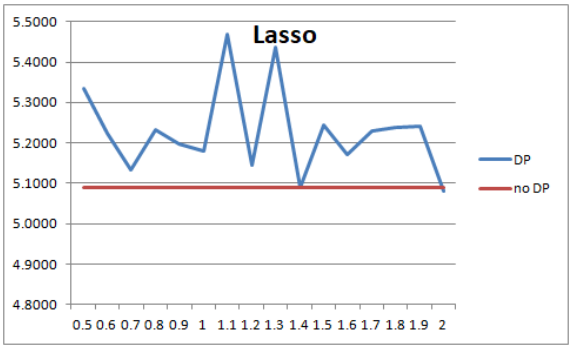


Fig. 5. MAE values for Lasso regressor for different values of DP parameter (with finer granularity).

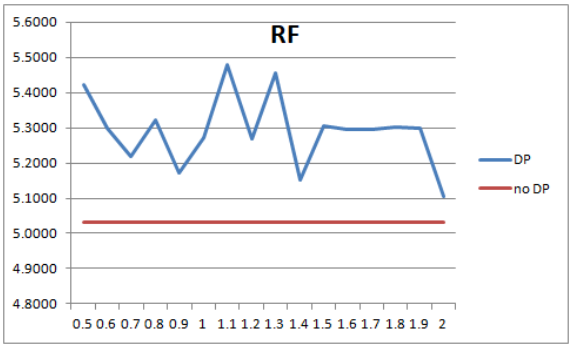


Fig. 6. MAE values for RF regressor for different values of DP parameter (with finer granularity).

5.4. Evaluation of Federated QoL Models

In experiments with simulated federated models, we have used different neural network architectures for BcBase and ORB datasets. A preliminary investigation, in which we have varied the number of hidden neural network layers between 1 and 10 and the batch size in the set $\{16, 64, 128, 256, 512\}$, showed that shallow neural networks (a small number of hidden layers) trained with a large batch size are more suitable for BcBase datasets, while deeper neural networks (a larger number of hidden layers) trained with a small batch size result with better predictive models for ORB datasets. We have simulated from 2 to 4 edge nodes training models in both incremental and concurrent federated learning mode.

Performance metrics for simulated federated models were obtained in the same way as for centrally-trained predictive models (by the 10-fold cross validation procedure).

The architecture of neural networks for federated binary classification models on the BcBase datasets consists of 4 hidden layers each having 20 nodes. Federated neural networks were trained in 200 epochs per simulated edge node with batch size equal to 512. The comparison of F_1 scores of centrally trained and simulated federated binary classification models is presented in Table 8. As baselines we use a centrally trained TensorFlow-based neural network (TFNN), the best centrally trained non-neural network classifier (Best C) and the worst centrally trained non-neural network classifier (Worst C). INC-k and CON-k denote federated neural network binary classification models trained in the incremental (INC) and concurrent (CON) learning mode for k simulated edge nodes.

Table 8. F_1 scores of federated binary classification models on BcBase datasets.

	Anxiety	Depression	Insomnia	Pain
INC-2	0.536	0.512	0.546	0.542
INC-3	0.542	0.507	0.529	0.542
INC-4	0.539	0.515	0.538	0.532
CON-2	0.522	0.504	0.542	0.548
CON-3	0.512	0.519	0.55	0.534
CON-4	0.53	0.509	0.542	0.545
Best C	0.552	0.534	0.554	0.522
Worst C	0.411	0.413	0.502	0.457
TFNN	0.438	0.53	0.54	0.542

For BcBase-Anxiety, Depression and Insomnia datasets, we have observed that federated models are significantly better than the worst performing local model (SVM and kNN depending on the dataset). F_1 scores of federated models are close to F_1 scores of NB which is the best performing centrally trained model. Federated models trained on BcBase-Pain have higher F_1 scores compared to the best performing centrally trained model on that dataset (DT). It is also important to emphasize that there are no significant differences in F_1 scores of incremental models and concurrent federated models. Additionally, the performance of federated models does not tend to significantly drop with the number of simulated edge nodes.

For federated regression models trained on the ORB dataset, we have used neural networks with 10 hidden layers each with 40 neurons. The training was performed in 200 epochs per simulated edge node with the batch size equal to 32. The obtained MAE scores are summarized in Table 9. For all six datasets, the best local model (LASSO) has lower prediction errors than simulated federated models. Large differences between federated models trained in different federated learning modes are absent. In contrast to federated models trained on BcBase datasets, here we can observe a tendency of increasing errors with the number of simulated edge nodes. Federated models are better than the DUMMY baseline for ORB-30-36, ORB-30-60, ORB-54-60 and ORB-108-120, but

worse than DUMMY for ORB-30-120 and ORB-54-120. This result implies that different neural network architectures should be employed for short term and long term QoL predictions. Therefore, our subsequent work will be to examine a wider range of neural network architectures for federated regression and determine architectures providing satisfactory results for long-term QoL predictions.

Table 9. MAE scores of federated regression models on ORB datasets.

	30-36	30-60	30-120	54-60	54-120	108-120
INC-2	6.012	6.775	7.488	5.931	7.188	6.625
INC-3	6.472	6.751	7.226	5.867	7.169	6.43
INC-4	6.595	7.042	7.463	6.22	7.206	6.484
CON-2	5.881	6.705	7.444	5.904	7.193	6.463
CON-3	6.404	6.826	7.427	5.986	7.098	6.327
CON-4	6.534	6.883	7.538	6.269	7.221	6.652
DUMMY	6.541	6.89	6.909	6.89	6.909	6.909
LASSO	5.089	5.886	6.478	4.84	6.18	5.437
TFNN	5.783	6.572	7.323	5.811	7.206	6.562

6. Conclusions and Future Work

In this paper we have examined several classification and regression machine learning algorithms for training models predicting binary and numeric QoL indicators, respectively, for breast and prostate cancer patients within ASCAPE project. The focus of our experimental evaluation was on two types of predictive models: (1) centrally-trained QoL models that are relevant either for individual data owners or for multiple data owners when it is allowed to collect training data in a central location, and (2) federated QoL models trained in distributed environments encompassing multiple data owners without data sharing. We also examined the influence of three different MVI algorithms and the privacy-accuracy trade-off parameter in DP approach on the performance of examined predictive models.

In the MVI pre-processing task we tested 3 approaches: simple imputer, iterative imputer and imputer based on kNN rule. The kNN imputer is further tested with 10 values of the parameter k, from 1 to 10. It is notable that kNN imputer has almost steady down trend (in the MAE values) while the k is increasing, obtaining the best values for $k = 10$. However, these differences are not particularly important, and the simple and iterative imputers obtain similar results as kNN for higher values of k. Our analysis of simple, iterative and kNN MVI algorithms showed that the choice of the MVI does not make a significant influence to the performance of examined models. Therefore, our choice for future analysis will be the simple imputer since it achieves comparable results as the other two and is much faster than the other two.

In the differential privacy task the goal was to add a controlled amount of noise to the training data in order to preserve the privacy of the patients. More noise means more

privacy, but lower performance of AI models (measured with MAE and other evaluation metrics values). The amount of noise is controlled through parameter ϵ, and the goal here was to find the value of ϵ with highest privacy and with satisfying regression performance. After extensive set of experiments our conclusion is to further utilize value of DP parameter $\epsilon = 0.7$.

Our experimental evaluation on real datasets showed that numeric QoL indicators can be accurately predicted by both centrally-trained and federated models for short term future periods. Centrally-trained regression models provide also accurate long term predictions. On the other hand, federated regression models exhibit prediction errors close to the dummy regression model indicating that different neural network architectures should be employed for learning regression-based federated models providing short term and long term QoL predictions.

For classification-based models predicting binary QoL indicator it was observed that centrally-trained and federated models have comparable prediction performances. However, for both types of model we noticed that they achieve relatively low precision and recall scores for the minority class due to imbalanced training datasets indicating that appropriate data sampling techniques should be examined to form more class-balanced training datasets prior to model training.

In our future work, we will also examine various feature selection techniques to identify the most relevant features for making QoL predictions and examine the performance of predictive QoL models trained on selected features. Furthermore, it would be interesting to examine federated learning approaches which are not based on neural networks like: decision trees [24] or logistic regression [25].

Acknowledgments. This research was supported by the ASCAPE project. The ASCAPE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 875351. The authors would also like to thank the anonymous reviewers for their insightful suggestions and comments that helped improve the quality of the paper.

References

1. Savić, M., Kurbalija, V., Ilić, M., Ivanović, M., Jakovetić, D., Valachis, A., Autexier, S., Rust, J., Kosmidis, T.: Analysis of Machine Learning Models Predicting Quality of Life for Cancer Patients, p. 35–42. Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3444757.3485103
2. Sidey-Gibbons, J., Sidey-Gibbons, C.: Machine learning in medicine: a practical introduction. BMC Medical Research Methodology 19 (03 2019)
3. Saadat, S., Aziz, A., Ahmad, H., Imtiaz, H., Sohail, Z., Kazmi, A., Aslam, S., Naqvi, N., Saadat, S.: Predicting quality of life changes in hemodialysis patients using machine learning: Generation of an early warning system. Cureus 9 (09 2017)
4. Sim, J., Kim, Y., Kim, J., Lee, J., Kim, M.S., Shim, Y., Zo, J., Yun, Y.H.: The major effects of health-related quality of life on 5-year survival prediction among lung cancer survivors: applications of machine learning. Scientific Reports 10, 10693 (07 2020)
5. Velikova, G., Booth, L., Smith, A., Brown, P., Lynch, P., Brown, J., Selby, P.: Measuring quality of life in routine oncology practice improves communication and patient well-being: A randomized controlled trial. Journal of clinical oncology : official journal of the American Society of Clinical Oncology 22, 714–24 (03 2004)

6. Singh, A., Pannu, H.S., Malhi, A.: Explainable information retrieval using deep learning for medical images. Computer Science and Information Systems 19(1), 277–307 (01 2022)
7. Šušteršič, T., Peulić, M., Peulić, A.: FPGA Implementation of Fuzzy Medical Decision Support System for Disc Hernia Diagnosis. Computer Science and Information Systems 18(3), 619–640 (06 2021)
8. Bratić, B., Kurbalija, V., Ivanović, M., Oder, I., Bosnić, Z.: Machine learning for predicting cognitive diseases: Methods, data sources and risk factors. J. Med. Syst. 42(12) (oct 2018), https://doi.org/10.1007/s10916-018-1071-x
9. Sinha, R., Heuvel, W.: A systematic literature review of quality of life in lower limb amputees. Disability and rehabilitation 33, 883–99 (06 2011)
10. Spiga, O., Cicaloni, V., Fiorini, C., Trezza, A., Visibelli, A., Millucci, L., Bernardini, G., Bernini, A., Marzocchi, B., Braconi, D., Prischi, F., Santucci, A.: Machine learning application for development of a data-driven predictive model able to investigate quality of life scores in a rare disease. Orphanet Journal of Rare Diseases 15 (12 2020)
11. Kaur, M., Dhalaria, M., Sharma, P., Park, J.: Supervised machine-learning predictive analytics for national quality of life scoring. Applied Sciences 9, 1613 (04 2019)
12. Gonçalves, J., Faria, B.M., Reis, L.P., Carvalho, V., Rocha, A.: Data mining and electronic devices applied to quality of life related to health data. In: 2015 10th Iberian Conference on Information Systems and Technologies (CISTI). pp. 1–4 (2015)
13. Kumar, S., Rana, M., Verma, K., Singh, N., Sharma, A., Maria, A., Singh, G., Khaira, H., Saini, S.: Prediqt-cx: Post treatment health related quality of life prediction model for cervical cancer patients. PloS one 9, e89851 (02 2014)
14. Yang, Z., Olszewski, D., He, C., Pintea, G., Lian, J., Chou, T., Chen, R.C., Shtylla, B.: Machine learning and statistical prediction of patient quality-of-life after prostate radiation therapy. Computers in Biology and Medicine 129, 104127 (2021), https://www.sciencedirect.com/science/article/pii/S0010482520304583
15. Melin, R., Fugl-Meyer, K., Fugl-Meyer, A.: Life satisfaction in 18-to 64-year-old swedes: In relation to education, employment situation, health and physical activity. Journal of rehabilitation medicine : official journal of the UEMS European Board of Physical and Rehabilitation Medicine 35, 84–90 (04 2003)
16. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. 10(2) (Jan 2019), https://doi.org/10.1145/3298981
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
18. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Eichner, H., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., Zhao, S.: Advances and open problems in federated learning (2021)
19. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. p. 265–283. OSDI'16, USENIX Association, USA (2016)

20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
21. Buuren, S., Groothuis-Oudshoorn, C.: MICE: Multivariate imputation by chained equations in R. Journal of Statistical Software 45 (12 2011)
22. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography. pp. 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
23. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. p. 1322–1333. CCS '15, Association for Computing Machinery, New York, NY, USA (2015), https://doi.org/10.1145/2810103.2813677
24. Li, Q., Wen, Z., He, B.: Practical federated gradient boosting decision trees (2019)
25. Yang, S., Ren, B., Zhou, X., Liu, L.: Parallel distributed logistic regression for vertical federated learning without third-party coordinator (2019)

Miloš Savić is an Associate Professor with the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, where he received his B.Sc., M.Sc., and Ph.D. degrees in computer science in 2010, 2011 and 2015, respectively. His research interests are related to complex network analysis and machine learning. He is/was a member of several international and bilateral research projects including three H2020 projects. He published more than 40 papers in international journals and proceedings of international conferences.

Vladimir Kurbalija holds the position of Full Professor from 2021 at the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia, where he received his B.Sc., M.Sc. and Ph.D. degrees. He was/is a member of several international projects supported by DAAD, TEMPUS, Horizon, and bilateral programs. Vladimir (co)authored over 40 papers in Case-Based Reasoning, Time-Series Analysis, Medical decision-support systems, and related fields. He was a member of Program Committees of several international conferences, and a reviewer in more than 20 international journals.

Mihailo Ilić currently works as a Teaching Assistant at the Faculty of Sciences, University of Novi Sad, Serbia, where he is also pursuing his PhD. His main fields of interests are agent technologies and machine learning, with a focus on federated learning techniques. As a member of the ASCAPE H2020 project, his work is focused on the implementation of federated learning in the medical domain. In addition, Mihailo has notable experience from working in industry as both a software developer responsible for system design and a data scientist in the field of smart agriculture.

Mirjana Ivanović holds the position of Full Professor at Faculty of Sciences, University of Novi Sad, Serbia. She is a member of National Scientific Committee for Electronics, Telecommunication and Informatics within Ministry of Education, Science and Technological Development, Republic of Serbia. Prof. Ivanovic is author or co-author of 14 textbooks, several international monographs and more than 450 research papers in

areas: agent technologies, intelligent techniques, applications of machine learning techniques in medical domains and technology enhanced learning. She is member of Program Committees of more than 300 international conferences and Program/General Chair of several international conferences. Mirjana Ivanovic delivered several keynote speeches at international conferences and visited numerous academic institutions all over the world as visiting researcher. Currently she is Editor-in-Chief of the Computer Science and Information Systems.

Dušan Jakovetić received the Dipl. Ing. Degree from the School of Electrical Engineering, University of Belgrade, Belgrade, Serbia, in 2007, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, and the Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal, in 2013. Dr. Jakovetic is an associate professor at the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia. Prior to that, he worked as an assistant professor with the same faculty, a research fellow at the BioSense Institute, Novi Sad, Serbia, and a postdoctoral researcher at IST. Dr. Jakovetic's research interests include distributed inference, learning, and optimization.

Antonios Valachis is Associate Professor of Oncology since 2018 and senior consultant at the Department of Oncology, Örebro University Hospital. His clinical work and research is focused on breast cancer and melanoma.

Serge Autexier studied and received his doctorate (2003) in computer science with a focus on AI. He is a principal researcher at at the German Research Center for Artificial Intelligence (DFKI), which he joined in 2002, and his research areas are formal methods, cyber-physical systems, human-environment/-robot interaction, and active and healthy living. Since 2014 he is head of the Bremen Ambient Assisted Living Lab and DFKI Research Fellow since 2017 and was and engaged as PI or Co-PI in more than 15 national projects and EU projects.

Johannes Rust received the Master of Science degree in Systems Engineering from the University of Bremen in 2020. Since then he worked in the department for Cyber-Physical Systems of the German Research Center for Artificial Intelligence (DFKI) at the site Bremen. His main research topics are explainable AI and computer vision.

Thanos Kosmidis is cofounder and CEO of CareAcross, a digital health company focusing on cancer, providing personalised support with evidence of improving patients' quality of life. Thanos has been at Imperial College, Carnegie Mellon University, Massachusetts Institute of Technology. Before CareAcross he worked in leading technology companies in the USA and Europe.

Received: February 27, 2022; Accepted: August 22, 2022.

Computer Science and Information Systems 20(1):405–421 https://doi.org/10.2298/CSIS2202227062I

Internet of Things and Agent-based System to Improve Water Use Efficiency in Collective Irrigation*

Abdelouafi Ikidid, Abdelaziz El Fazziki, and Mohamed Sadgal

Computer Science Dept, Computing Systems Engineering Laboratory (LISI).
Cadi Ayyad University Marrakesh, Morocco
a.ikidid@gmail.com
elfazziki@uca.ma
sadgal@uca.ma

Abstract. The efficient management of water resources is a major issue in the field of sustainable development. Several models of solving this problem can be found in the literature, especially in the agricultural sector which represents the main consumer through irrigation. Therefore, Irrigation management is an important and innovative field that has been the subject of several types of research and studies to deal with the different activities, behaviors, and conflicts between the different users. This article introduces an intelligent irrigation system based on smart sensors that can be used moderately and economically to monitor farms by integrating some connected electronic devices and other advantageous instruments widely used in the field of IoT, it determines the water requirement of each farm according to the water loss due to the process of evapotranspiration. The water requirement is calculated from data collected from a series of sensors installed in the plantation farm. This project focuses on smart irrigation based on IoT and agent technology, it can be used by farmer associations whose endowments and irrigation planning are defined according to the need and quantity of water available in the rural municipality. The system includes a microcontroller with the integration of sensors, actuators, and valve modules where each node serves as an IoT device. Environmental parameters are monitored directly through a multi-agent system that facilitates the control of each node and the configuration of irrigation parameters. The amount of water calculated for irrigation is based on the Penman model for calculating the daily evapotranspiration baseline. Compared to the conventional irrigation method, it is expected that the proposed irrigation model would contribute to saving water use and distributing it impartially without compromising its production.

Keywords: Agent technology, smart irrigation, sensors, internet of things, Evapotranspiration.

1. Introduction

In Morocco, irrigation systems are very diverse. The State is in the process of setting up a new strategy for water resources management. Also, different projects are being conducted to improve the collective management of irrigation. They are intended to be

* Extended version of a conference paper, MEDES '21, November 1–3, 2021, Tunisia.

participatory and aim to take charge of the management of irrigation networks according to the same model of association of agricultural water users.

Currently, we note that gravity-fed irrigation systems based on motorized wells have several limitations and cause a great loss of water. Good management of these irrigation systems is mainly characterized by better management and planning of water between the different actors and is therefore necessary. However, this management is subject to several constraints: the water allocation granted, the type of crop for each plot, the sowing date of each crop, climatic data, optimal irrigation planning, etc.

The most common irrigation technique used in Morocco is gravity-fed irrigation, which has several limitations since it does not take into account the type of crop and soil, the climatic conditions, and the actual water needs. Thus, for a better allocation of water resources in this type of irrigation system, it is necessary to involve all the actors concerned to establish negotiations between them on the planning of sowing, crop rotation, the water needs of the crop, and the water allocation.

The IoT (Internet of things) offers the possibility to optimize the irrigation of farms, the maintenance of agricultural machinery, the analysis and the remote control, and so on. The connected sensors placed in the field allow the optimization of irrigation by performing a complete analysis of the plant's elements.

To address this issue of water resource allocation in rural communities, we propose to model irrigation network management systems in an intelligent context based on a multi-agent approach coupled with IoT, which will form the basis for the development of a tool to assist in the negotiation of allocations and the planning of intelligent irrigation, which aims to improve profitability and better manage the distribution of water resources before the start of each agricultural season.

The system constructs a real-time irrigation decision based on the predicted soil moisture estimated at the moment of precipitation. The soil moisture prediction is performed depending on the analysis of the data sensed by the soil moisture sensor and the evaporation prediction. The evaporation is predicted using five factors (air temperature, wind speed and direction, and humidity). Furthermore, according to [1] an IoT-based Smart Greenhouse system is designed with a novel monitoring combination including warning, automation, disease forecast, and cloud repository; by employing a readily deployable complete package. It continually maintains dynamic conditions such as temperature, humidity, and soil moisture state to improve the crop yield and to guarantee an instantaneous reaction in the event of abnormal conditions.

The main reason why we opted for a smart solution is the complexity of managing in real time the water distribution operations that arrive asynchronously and dynamically and to be reactive and adaptive to the dynamic and unpredictable events that characterize the domain. In this work, we propose the design of a functional prototype based on the calculation of the water needs of different crops of farmers, a NodeMCU, a soil moisture sensor, a temperature sensor and humidity, a relay module, and a motor. The whole system is integrated into a multi-agent system in such a way that it monitors all the components of the system, allocates the endowments through negotiation with the farmers, monitors the irrigation plans and periods, and supports the endowments.

The rest of the paper is organized as follows: The second section analyzes and discusses the related works about intelligent irrigation systems. The third section gives a global overview of the irrigation problem and the multi-agent system model. The fourth

section details the proposed approach. The fifth provides the detailed results of the simulation and tests. Finally, we conclude in the Sixth section.

2. Related Works

Smart farming improves yields by using minimal resources such as water, fertilizers, and seeds. Farmers can easily deploy sensors to remotely monitor their crops, conserve resources, and reduce the impact of climate changes on crops [2]. Several values parameter detection technologies are used in this agriculture for providing data and helping farmers to monitor and optimize their crops [3], as well as to adapt to environmental change factors, including location, electrochemical, mechanical, airflow sensors, agricultural weather stations, humidity, and PH.

One of the main sensors in smart farming is that of soil moisture. It's used for measuring the present volumetric water content (humidity) in the soil. The threshold value is fixed, and the level of soil moisture value is measured and verified with the upper and lower thresholds at the necessary levels. Irrigation is the vital need of agricultural activities, there are three classic irrigation methods of which we can cite canal irrigation, sprinkler irrigation, and drip irrigation responding to the needs of plants, these three methods are used. Regarding the intelligent irrigation system, the researchers in [4] have shown that water consumption is minimized when an automated irrigation system relies on soil moisture as an implementation parameter. Among these irrigations, that of drip is the one where farmers can save the most water because it will provide water in the form of droplets directly on the plant root and the soil surface.

Intelligence farming had been implemented and many of them were incorporated with Artificial Intelligence and Internet of Things technologies to enhance agricultural production and optimize resource use. Some research reports using smartphones and sensors to remotely monitor the soil condition and enable smart irrigation [5]. A more complex system combines IoT and artificial intelligence techniques such as machine learning [6], Fuzzy logic [7], deep Q-learning [8], artificial neural network [9] and Multi-Agent systems [10] [11] , and to handle diverse aspects, e.g., irrigation, fertilization, or pesticide treatment and so on. In what follows we analyze and discuss succinctly several studies that use a multi-agent system and IoT to perform intelligent farming systems.

Since IoT deals with a large data set received from heterogeneous sensors and modeled in various formats, the necessity of a homogeneous data interpretation claims new and challenging methods for data treatment. To address such challenges, it is necessary to integer AI technologies (such as agent systems, MAS), allowing us to automate the uptake and analysis of the various sets of data that come from different sensors [12]. Many authors [13] use AI to develop data processing in agriculture by obtaining knowledge from the data collected through heterogeneous data sensors and acting continually and automatically and accordingly based on the collected data. In [14] propose an agent-based system is proposed to automate the data management process and provide an optimized irrigation system, the data collected from various sensors is through MAS to make a knowledge base used in the decision-process to develop an

irrigation strategy that meets the environment needs and save the resource improving the agriculture production.

T. Wanyama and B. Far [15] combine Multi-Agent and Fuzzy Logic techniques to introduce a smart irrigation system, this study uses Fuzzy logic to deal with the uncertainty that characterizes the information that affects the irrigation schedule. Other research uses the MAS negotiation mechanism to improve the water allocation [16], agents represent different frames to calculate how much water is needed in the farm. The frame with excess water shares their water with those with other farmers needing water to ensure efficient water distribution.

The evapotranspiration (ET) estimation is at the queen's interest in the hydrologic cycle studies but still lay on the line to uncertainties. Hence, Estimates and predictions of the ET constitute an essential step of irrigation management over the world. In this context, a multitude of studies is made to improve ET Estimation. Especially, in [17] the study used Bowen ratio and eddy covariance methods for calibrating, extracting, and calculating relevant parameters necessary for guessing ET of tea canopy for the whole growing season, to provide further predictions about the adoption of this method for scheduling other crops irrigation as well. Also, another comparative study performed a comparison between three ET estimating methods, in particular the eddy covariance (EC) and Bowen ratio-energy balance (BREB), and the soil water balance (SWB) to measure their efficiency during the cropping season of corn in [18]. Moreover, the FAO Penman-Monteith (FAO PM) has been declared as the standard method to estimate ETo for over the last decades. This method takes into account many climatic variables linked to the evapotranspiration activity such as the net radiation (Rn), the air temperature (T), the vapor pressure deficit (Δe), and the wind speed (U); and its results are very satisfactory referring to [19].

3. Multi-agent Model for Irrigation Management

3.1. IoT and Multi-agent system

The Internet of Things is based mainly on sensors and connected objects placed in physical infrastructures. These sensors will transmit data that will be sent using a wireless network on IoT platforms. Thereafter, these data will be analyzed and enriched to get the most out of them. These data management and data visualization platforms are the new IoT solutions allowing territories, companies, or even users to analyze data and draw conclusions to be able to adopt practices and behaviors.

The proposed solution is based on microcontrollers and a multiagent platform. This choice is based on their computing power, their cost, and their scalability. With the use of various sensors, the variable parameters will be continuously monitored and the irrigation adapted to the type of crop.

This approach focuses on using multi-agent and IoT techniques to develop an efficient system for water management in a collective irrigation scheme. Farms are controlled by a MAS that can calculate the total water requirement of the farm based on

the environment variable which are temperature, soil moisture, and rain, and control the water allocated to each farm. The multi-agent environment and the model of the multi-agent water management system are depicted in **Fig. 1** [20].

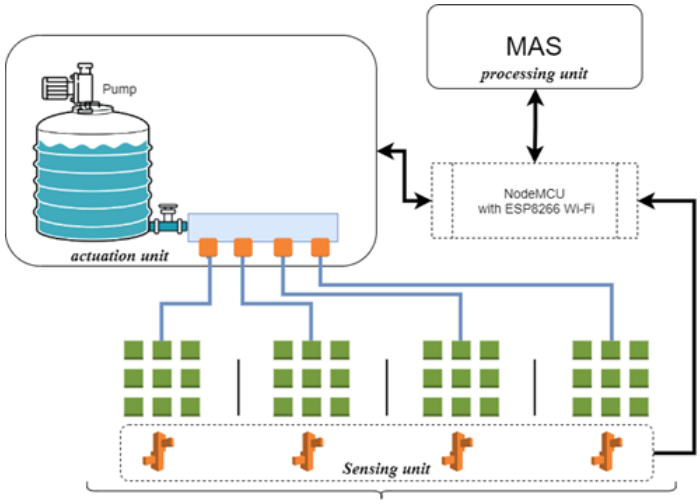


Fig. 1. Multi-agent irrigation management system

3.2. Materials And Methods

The system benefits from the communication rapidly growing in the Internet o things in recent years. This system consists of a wireless sensor network that collects different variables of the environment, the actuation unit that controls a mechanism of irrigation and water storage applying the strategy determined by the MAS, the NodeMCU smart gateway that connects the sensing unit and actuation unit with the agent-based management system.

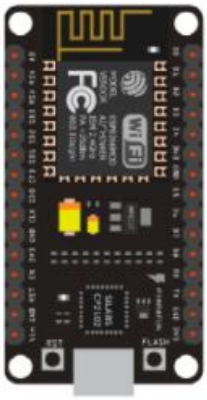
The sensors placed in the farm field, measure continuously the soil humidity and temperature values, the measurement of rainfall, and the water tank level, then send these values through the network to MAS, that information is then sent to MAS that uses an intelligent algorithm to analyze the collected data and to extract insights that support decision-making.

The model comprises the NodeMCU (Fig. 2 (a)) that operates synchronously with the sensor nodes that are physically installed in the farms. The sensors are deposited in the farm in the manner that they operate as the end devices of a point-to-point network. The end devices have the following features:

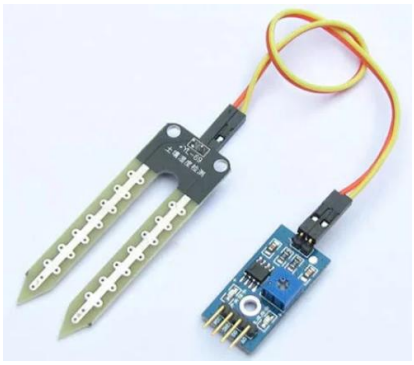
- be able to transfer data by connecting to a local area network
- allow the connection of any device.
- only send data and unable to edit this data.

NodeMCU is an open-source IoT development platform that contains a firmware system development board operating on the ESP8266 Wi-Fi SoC. It is a small

microcontroller with a Wi-Fi chip that allows to establish a communication between two ESP8266, when they are connected on the same network. The NodeMCU provides advanced hardware interfaces that take the complexity of hardware configuration and registry operations out of the hands of application developers.



(a) NodeMCU based on the ESP8266 development board. [1]



(b) The soil moisture sensor.[2]

(c) Temperature sensor.

(d) Water flow sensor (SeaYF-S201). [3]

Fig. 2. The infrastructure components.

The model also consists of sensor devices dedicated to detecting environmental parameters and sending the measured information to the NodeMCU. The proposed smart farm monitoring system consists of the following sensors resources:

[1] https://www.aranacorp.com/
[2] https://www.instructables.com/
[3] https://www.hobbytronics.co.uk/

The soil moisture sensor: illustrated in Fig. 2 (b) which has a working range of (0 to 1023 ADC value) and is used to measure the soil moisture content of the farm. It consists of two conductive probes that can perceive the soil moisture content in proportion to the change in resistance between the two conductive plates.

The temperature sensor (DS18B20): illustrated in Fig. 2 (c) which is a digital temperature sensor that has a working range from -55°C to +125°C with ±0.5°C Accuracy.

The Sea YF-S201 water flow sensor presented in Fig. 2 (d) is used to measure the rate of flow of water and calculate the amount of water followed through the pipe. It is characterized by a water pressure inferior to 1.95, Working Flow Rate from 1 to 30 L/min, and is connected to the pipe of the submersible water pump illustrated in figure 5.

4. Overview of the proposed approach

Multi-agent systems are today an emerging technology for the simulation, understanding, and resolution of complex problems through the design and implementation of open and distributed systems that can integrate human and/or artificial agents.

In this context, we are interested in the modeling of gravity irrigation systems, by applying agent technology. Nowadays, this technology has found its place in production systems (workshop scheduling, management of industrial processes, multi-sensor systems, etc.), control tasks (road traffic control, air traffic, distribution energy, ...)[21], telecommunications, transport systems [22], networks and many other applications. Furthermore, MAS integrates various intelligent techniques to optimize system performance[23].

As aforementioned, the objective of our work is the modeling of an intelligent irrigation management system by a multi-agent system that represents the different actors and defines the negotiation interactions between them and this before the start of each agricultural campaign for the resolution of the water allocation for irrigation problems. These negotiations are based on climatic constraints, agricultural constraints, and the various decisions and behaviors of the actors.

The application of this proposed approach will be carried out in four phases as shown in the Fig. 3:

- The first phase of collecting the data necessary for decision-making during the negotiation process.
- The second phase of analysis concerns the capture of needs and the functional specification.
- The third phase is Agent modeling : Based on the analysis the agent model identifies which agent class is tasked to play specific roles and how many instances of each class have to be instantiated.
- A fourth phase for the implementation and development of agents under the AnyLogic platform.

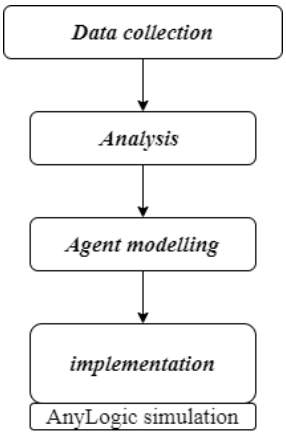


Fig. 3. The development process model

The irrigation system is functionally distributed. Commonly, agents are perceived as analyzing levels with abstraction upper than components and objects, which make MAS suitable with complex and distributed problems. In our proposed approach each farm is viewed as an isolated node monitored by a set of sensors. All nodes, as well as actuation units, are controlled by a community of autonomous, cooperative, and intelligent agents, this agent community is divided into subsystems tasked to achieve specific objectives.

4.1. Data Collection

The first step for the implementation of an intelligent irrigation system for agricultural automation is to place the wireless sensor network, each node is interconnected by a Wi-Fi module and deposits data on a common server, this server can continue to query the data and then send a commands signal for the required operation. Figure. 1. represents an overview topology of various sensor nodes, however, the actual network topology depends on the demographics of the region. The first step is to collect data via various sensors connected to the farms. The MAS will act as the gateway node responsible for communicating with all other sensor nodes. Each farm node consists of an advanced soil moisture sensor, Wi-Fi module, temperature and humidity, temperature sensor, water level indicators, alarm, clock module, battery, relay module. Ultrasonic sensor for the detection of intruders. Each of the nodes will relay the information to the Gateway which is the MAS which will be responsible for storing the data, analyzing it, and presenting it to the end-user through an application layer. The connection of various sensors to the microcontroller is based on the fundamental concept of receiver, transmitter. The first phase of the suggested system is complete after establishing the network topology and collecting data. Collecting data through various sensors is the prerequisite and the first step in data processing.

4.2. Analysis

All the data that has been collected by the sensors must be analyzed and processed so that the subsequent signal can be sent to the actuators as well as alerts can be sent to the end-user in case of manual interaction is required. The MAS will check various conditions from the data received from the nodes as well as from the web. Irrigation would be based on soil type and specific crop to automate the systems. Thus, after processing all the parameters, a control signal from the gateway node to the actuators will start the water pump. A continuous and recursive survey of soil moisture will be done at a fixed time interval and after a certain humidity level, irrigation will be stopped.

4.3. Agent Modeling

In this stage, we will establish the MAS infrastructure capturing the distributed aspect of the system, the interactions, and relationships that can take place between the different constituents of the system. For this, we based on the classification of the intelligent irrigation system on different tasks. This classification would make it possible to group all the tasks that are strongly linked in a subsystem and those that have little or no relationship to each other in different subsystems [22]. Thus, all the tasks relating to the preparation of the agricultural campaign, the negotiation of the annual water allocation in a first sub-system, the tasks relating to the execution of irrigation, and the allocation of water resources are negotiated in the second sub-system. A third subsystem is tasked to monitor the good progress of the agricultural campaign. Therefore, the intelligent irrigation system will be composed of the following three subsystems (see Fig. 4):

- Supervision and planning subsystem: responsible for the supervision of the whole system, the management of negotiations, the management of irrigation, the programming of the water tower, and the planning of the opening and closing operations of the irrigation valves. This subsystem serves a set of farmers, it must optimize the irrigation process (by deploying certain heuristics) to minimize the duration of the tour. Water and flow variations at the secondary canals. This subsystem will be modeled by two agents: Supervisor Agent, Scheduling Agent, and Water need Agent.
- Execution sub-system: responsible for monitoring the smooth running of the agricultural campaign through the execution of the water tower program established by the scheduling agent. This subsystem will be modeled by two agents: Agents Farm, Agent Source, Tank Agent, Pump Agent.
- Users' subsystem: made up of all the farmers who form the Water User Association (WUA), responsible for monitoring the progress of the irrigations and the negotiation of the water supply. This subsystem will be modeled by a WUA Agent.

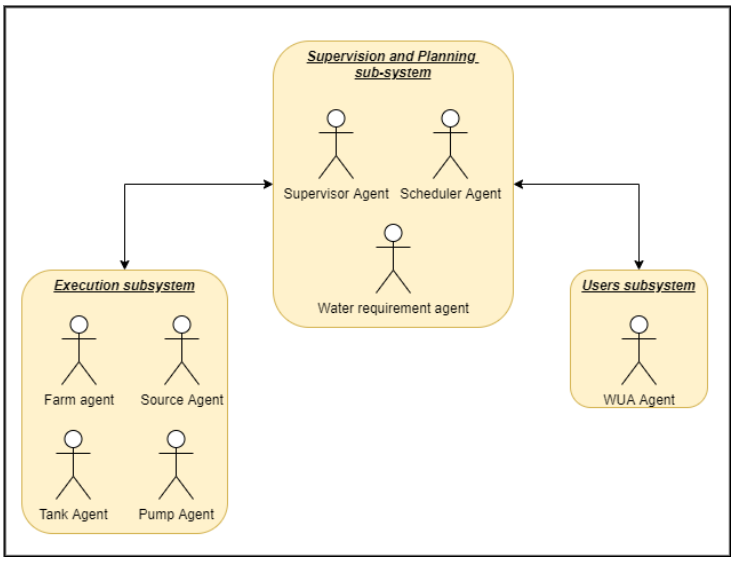


Fig. 4. The organization of the multi-agent system

Supervision and planning sub-system

The prescribed agent-based model defines specific agents in correspondence with the types of tasks in the precision subsystem. Each agent achieves a specific role, through cooperation and interaction with other agents in the same subsystem and the other subsystems. Various types of agents are conceived in the structuring process, dedicated to achieving different roles:

Supervisor Agent: The Irrigation Supervisor will operate and perform the supervision of all the irrigation processes and will assist the scheduler agent in the scheduling process. The Supervisor Agent represents a Development Center for the irrigation of the plots, it receives the needs of each crop and defines the allocations according to the availability of an annual water allocation from the Hydraulic Basin Agency (HBA), the annual water allocation will be subject to negotiation with the WUA agents in order to reach a compromise for its allocation. The Supervisor Agent will therefore be responsible for the following tasks:

- Inform the Scheduling Agent of the water allocation granted and of the list of farmers entitled to irrigate their plots.
- Negotiate the water supply with the farm agents.
- Send the final water allocation to the Scheduling Agent.
- Transmit the program of the water tower received by the Scheduling Agent to each farm Agent.
- Receive the authorization to start irrigation from the Source Agent and inform the Farm Agent.

- Manage the no satisfaction received from the farm Agent and manage the incidents during the irrigation planning.
- Establish reports on each farm.

Scheduling Agent: The Scheduling Agent is responsible for scheduling the tour by providing an optimal timing diagram for the filling of the channels of the network which will be sent to the Supervisor Agent.

Water requirement agent: calculate water requirements for each farm based on environment variables and water demand prediction process.

To ensure potential crop yields, the water requirements of the zone of a crop must be met. These needs are generally defined by the evapotranspiration of crops and represented by ETc. Evapotranspiration combines evaporation from the soil surface or wet surfaces of plants, and transpiration from leaves. Water needs could be met through precipitation, groundwater, or irrigation. Irrigation is therefore necessary when the crop water requirement (ETc) exceeds both stored water and rainfall. Since ETc is dependent on crop stage and weather changes, the amount and timing of irrigation are critical. The water balance approach allows easy estimation of water requirements for irrigation planning. This method is based on several factors, in particular, the initial soil water content in the root zone, ETc, rainfall, and soil capacity. The Daily crop evapotranspiration (ETc) can be accessed as follows:

$$ETc = ETr \times Kc \times Ks. \quad (1)$$

Where *ETr* is the evapotranspiration rate of the reference crop, *Kc* is the crop coefficient which depends on the stage of development (0 to 1) and *Ks* is the water stress coefficient (0 to 1). At each stage of the growing season, the Kc of each crop is essentially estimated as the ratio of its *ETc* to the *ETr* of the reference crop [24].

ETr (reference evapotranspiration) is estimated from climatological variables and expresses the effect of meteorological conditions on the net water requirements of crops, while *Kc* marks the characteristics of the crop and its effect on the requirements of plants. Water (type of crop, development, phonological stage, etc.). Third, the quality of the irrigation water, the uniformity coefficient of the irrigation system, the size of the field, etc., determine the actual amount of irrigation. This value gives a rough idea of the volume of water needed to meet the crop's water needs. This value is balanced with the water state of the soil to obtain the irrigation volume for the contribution to the harvest stage.

Execution sub-system

Farm Agent: each farm is assigned to an agent farm; its main task is to monitor the execution of the irrigation schedule transmitted by the Supervisor Agent. It is responsible for the irrigation of crops belonging to the zone for which he is responsible, in the form: "*distribute the quantities* Q_1, Q_2 ..., Q_n *respectively on the crops* C_1, C_2, ..., C_n".

Agent Source: The Source Agent represents the rural commune; its role is to allocate the flow requested by the fam agent by opening the flow control valves which are installed at the level of the tank and which are under his responsibility. This agent also reads the tank level information from the sensors and makes sure this level is in the convenient degree.

Tank Agent: This agent charges to control the tank**.**

Pump Agent: this agent monitors the initial condition requiring the tank to be full before water circulation is started.

Users' subsystem

WUA Agent: The WUA Agent represents a Water User Association which is made up of a representative and the farmers belonging to it, its role is to negotiate the annual water allocation with the Supervisory Agent before the agricultural campaign. This agent also has the role of monitoring the progress of irrigations and informing the agent farm or the supervisor Agent of incidents in the irrigation canals. This agent plays the role of the user interface. It represents the human-machine interface and focuses on displaying the irrigation state to the final users.

Each agent acts according to a set of rules, whose format depends on its variables.

Fig 5 represents the architecture of the intelligent irrigation system and interactions between agents.

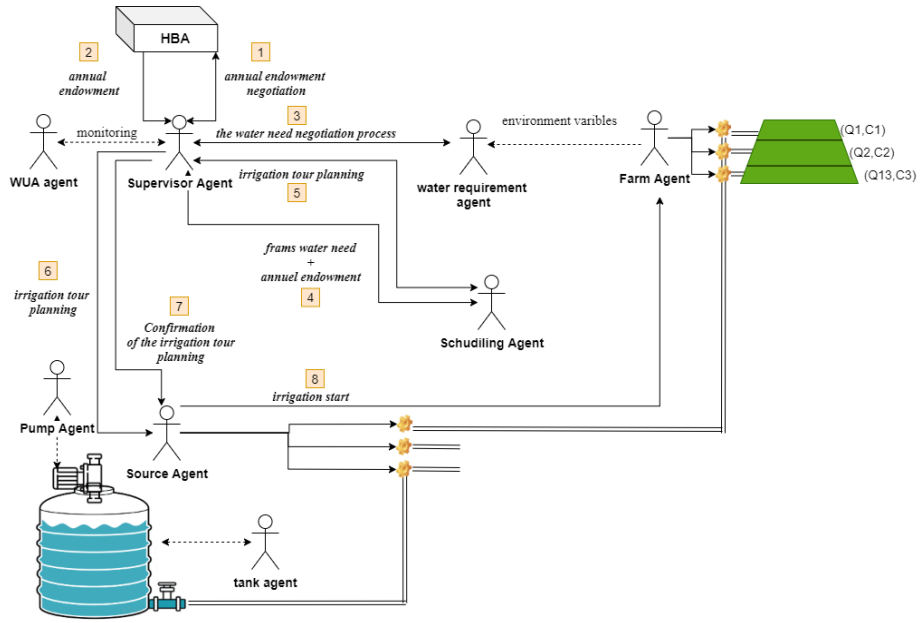


Fig. 5. The architecture of the MAS smart irrigation system

5. Experimental Results and Performance Analysis

5.1. Case Study

Our study area is jointly managed by three farmers' associations and the local center of the Haouz Regional Agricultural Development Office, especially in terms of irrigation. At the start of the season, they decide on the number of water towers and the quantities allocated for irrigating cereal crops. This quantity depends on the annual allocation granted and negotiated with the Hydraulic Basin Agency, which is the main supplier of irrigation water. At each turn, farmers receive a volume of water according to the size of the farms, regardless of the type of crop and its water needs, even if some areas are unexploited or in biological comfort time.

5.2. Workflow of The System

At the start, the water pump is disabled, then if the sensed values go beyond the threshold values set in the program (soil moisture< soil saturation = 360 = 64.80% or temperature > 30°C), it turns ON and irrigates the soil of the farm. Whenever the pump is turned ON, the flow sensor restarts measuring the irrigation flow, the horologe sleeps for a predetermined time, then the measures are recorded again until the soil moisture outperforms the soil saturation. This time control allows an optimized irrigation time which optimizes the amount of water flowing to the farm by minimizing the water loss that may be related to a long period of irrigation time before restarting the sensing and the threshold verification. In the case where the soil moisture exceeds the configured soil saturation level, the pump turns off, and the horologe sleeps for a second predetermined time before repeating the iteration. This process can be summarized as follows:

Step 1. Start.

Step 2. The system's parameters can be initialized.

Step 3. The clock date is initialized on the Arduino UNO board, and the water pump is set to OFF.

Step 4. The farm agent checks the soil parameters level constantly.

Step 5. The microcontroller updates the current date and checks for the threshold condition. If the water content level is inferior to the soil saturation level (360) of the farm, which means that the soil becomes dry or the temperature surpasses (30°C), then the relay that is connected to the Arduino UNO will turn ON the motor to irrigate the field. The delay is set to (5 seconds) to repeat step 3. Otherwise, if the threshold condition isn't satisfied, the motor will be turned OFF, and the delay will be set to (1 hour) before repeating step 4.

The performance of the proposed system is validated and designed in the ANYLOGIC simulator, which is used to handle the MAS. Based on the JAVA language, AnyLogic is a programing and simulation platform used to model hybrid systems, it allows to handle agent modeling, graphical model editor and code generator.

5.3. Results and Analysis

Environment parameters are generated and adjusted to simulate different scenarios. The measures are recorded after a sleep time equal to (1 hour). (Fig. 6 and 7) show the temperature fluctuations and experimental results.

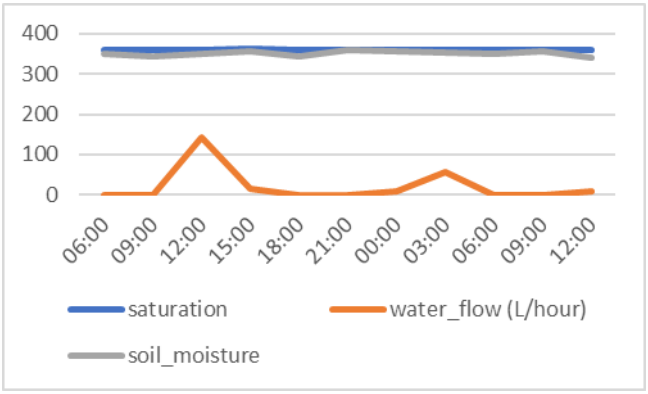


Fig. 6. Soil moisture changes and water flow fluctuations over time

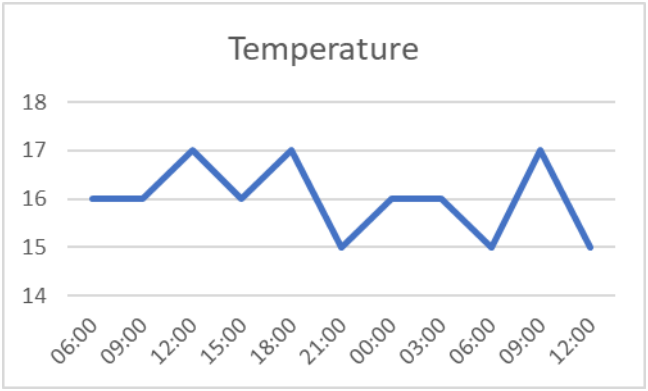


Fig. 7. Temperature fluctuations over time

6. Conclusion

In this paper, we propose a smart collective irrigation system using an agent and the internet of things technologies. The development of the proposed system is based on a specific methodology to design an intelligent irrigation system that determines the water requirement of each farm according to the water loss due to the process of

evapotranspiration. The water requirement is calculated from data collected from a series of sensors installed in the plantation farm. This project focuses on smart irrigation based on IoT that is effective and can be used by farmer associations whose endowments and irrigation planning are defined according to the need and quantity of water available in the rural municipality.

The designed system can operate successfully for irrigation purposes with equity in calculating needs and allocating endowments in proportion to those needs. It frees farmers from the tedious task of irrigation while optimizing considerable quantities of water as well as preventing rural communities from managing water allocation and distribution. The system provides a suitable working device and allows the farmer to monitor his fields from a distance. Excessive irrigation can be avoided with this system, which can damage crops. The system can reduce the problems encountered in traditional agriculture and collective irrigation.

Our solution is very generic and adaptable to different contexts and collective irrigation issues. However, applying the solution in certain situations requires more functional and technical specifications. Specific modules that can be integrated into the management systems will have to be developed separately. In addition, we have implemented prototype applications, the implementation of such systems in practice requires the contribution of several contributors and experts in the field, as well as the implementation of various equipment and important material resources.

In the future, the proposed system will be tested and instantiated in an adequate simulation platform. The experimental result and performance analysis can provide information about the proposed system's effectiveness and performance.

References

1. B. Fatima, S. I. Siddiqui, R. Ahmad, N. T. T. Linh, and V. N. Thai, “CuO-ZnO-CdWO4: a sustainable and environmentally benign photocatalytic system for water cleansing,” Environmental Science and Pollution Research, vol. 28, no. 38, pp. 53793–53803, 2021, doi: 10.1007/s11356-021-14543-9.
2. M. M. Maha, S. Bhuiyan, and M. Masuduzzaman, “Smart board for precision farming using wireless sensor network,” 1st International Conference on Robotics, Electrical and Signal Processing Techniques, ICREST 2019, pp. 445–450, 2019, doi: 10.1109/ICREST.2019.8644215.
3. [R. Ramya, C. Sandhya, and R. Shwetha, “Smart farming systems using sensors,” in Proceedings - 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development, TIAR 2017, 2018, vol. 2018-Janua, doi: 10.1109/TIAR.2017.8273719.
4. J. Gutierrez, J. F. Villa-Medina, A. Nieto-Garibay, and M. A. Porta-Gandara, “Automated irrigation system using a wireless sensor network and GPRS module,” IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 1, 2014, doi: 10.1109/TIM.2013.2276487.
5. O. K. A, “A Mobile Phone Controllable Smart Irrigation System,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 1, pp. 279–284, Feb. 2020, doi: 10.30534/ijatcse/2020/42912020.
6. M. Waleed, T. W. Um, T. Kamal, and S. M. Usman, “Classification of agriculture farm machinery using machine learning and internet of things,” Symmetry, vol. 13, no. 3, pp. 1–16, 2021, doi: 10.3390/sym13030403.
7. N. S. Pezol, R. Adnan, and M. Tajjudin, “Design of an Internet of Things (Iot) Based Smart Irrigation and Fertilization System Using Fuzzy Logic for Chili Plant,” 2020 IEEE

International Conference on Automatic Control and Intelligent Systems, I2CACIS 2020 - Proceedings, no. June, pp. 69–73, 2020, doi: 10.1109/I2CACIS49202.2020.9140199.
8. M. E. Pérez-pons, R. S. Alonso, O. García, G. Marreiros, and J. M. Corchado, “Deep Q-Learning and Preference Based Multi-Agent System for Sustainable Agricultural Market,” pp. 1–16, 2021.
9. F. Almomani, “Prediction of biogas production from chemically treated co-digested agricultural waste using artificial neural network,” Fuel, vol. 280, no. April, p. 118573, 2020, doi: 10.1016/j.fuel.2020.118573.
10. Y. W. Ma, J. Q. Shi, J. L. Chen, C. C. Hsu, and C. H. Chuang, “Integration Agricultural Knowledge and Internet of Things for Multi-Agent Deficit Irrigation Control,” International Conference on Advanced Communication Technology, ICACT, vol. 2019-Febru, pp. 299–304, 2019, doi: 10.23919/ICACT.2019.8702012.
11. A. Ikidid, E. F. Abdelaziz, and M. Sadgal, “Multi-Agent and Fuzzy Inference-Based Framework for Traffic Light Optimization,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. InPress, no. InPress, p. 1, 2021, doi: 10.9781/ijimai.2021.12.002.
12. A. Ikidid, A. El Fazziki, and M. Sadgal, “A Multi-Agent Framework for Dynamic Traffic Management Considering Priority Link,” International Journal of Communication Networks and Information Security, vol. 13, no. 2, pp. 324–330, 2021, doi: 10.54039/ijcnis.v13i2.4977.
13. D. Alfer’ev, “Artificial intelligence in agriculture,” Agricultural and Lifestock Technology / АгроЗооТехника, no. 4 (4), 2018, doi: 10.15838/alt.2018.1.4.5.
14. A. González-Briones, Y. Mezquita, J. A. Castellanos-Garzón, J. Prieto, and J. M. Corchado, “Intelligent multi-agent system for water reduction in automotive irrigation processes,” Procedia Computer Science, vol. 151, no. 2018, pp. 971–976, 2019, doi: 10.1016/j.procs.2019.04.136.
15. T. Wanyama and B. Far, “Multi-Agent System for Irrigation Using Fuzzy Logic Algorithm and Open Platform Communication Data Access,” vol. 11, no. 6, pp. 703–708, 2017.
16. K. Chiewchan, P. Anthony, K. C. Birendra, and S. Samarasinghe, Improving Water Allocation Using Multi-agent Negotiation Mechanisms, vol. 148. Springer Singapore, 2020.
17. M. Smith, R. Allen, and L. Pereira, “Revised FAO methodology for crop-water requirements,” International Atomic Energy Agency (IAEA), 1998.
18. S. R. Prathibha, A. Hongal, and M. P. Jyothi, “IOT Based Monitoring System in Smart Agriculture,” in Proceedings - 2017 International Conference on Recent Advances in Electronics and Communication Technology, ICRAECT 2017, 2017, doi: 10.1109/ICRAECT.2017.52.
19. P. C. Sentelhas, T. J. Gillespie, and E. A. Santos, “Evaluation of FAO Penman-Monteith and alternative methods for estimating reference evapotranspiration with missing data in Southern Ontario, Canada,” Agricultural Water Management, vol. 97, no. 5, 2010, doi: 10.1016/j.agwat.2009.12.001.
20. A. Ikidid, A. El Fazziki, and M. Sadgal, “Smart collective irrigation: Agent and internet of things based system,” ACM International Conference Proceeding Series, pp. 100–106, Nov. 2021, doi: 10.1145/3444757.3485113.
21. A. Ikidid and E. F. Abdelaziz, “Multi-Agent and Fuzzy Inference Based Framework for Urban Traffic Simulation,” in Proceedings - 2019 4th International Conference on Systems of Collaboration, Big Data, Internet of Things and Security, SysCoBIoTS 2019, 2019, doi: 10.1109/SysCoBIoTS48768.2019.9028016.
22. A. Ikidid and A. El Fazziki, “Multi-agent based traffic light management for privileged lane,” 8th International Workshop on Simulation for Energy, Sustainable Development and Environment, SESDE 2020, pp. 1–6, 2020, doi: 10.46354/i3m.2020.sesde.001.

23. A. Ikidid, A. El Fazziki, and M. Sadgal, “A Fuzzy Logic Supported Multi-Agent System for Urban Traffic and Priority Link Control,” JUCS - Journal of Universal Computer Science, vol. 27, no. 10, pp. 2987–3006, 2021, doi: 10.3897/jucs.69750.
24. A. A. Andales, J. L. Chávez, and T. A. Bauder, “Irrigation Scheduling: The Water Balance Approach,” Colorado State University Extension, no. 4, pp. 1–6, 2011.

Abdelouafi Ikidid received the Ph.D. degree in computer science from Cadi Ayyad University in 2022. He received his Master’s degree in Information Systems Engineering from the same university in 2016. His research interests are in software engineering, focusing on multi-agent systems and artificial Intelligence.

Abdelaziz El Fazziki received the M.S. degree from the University of Nancy, France, in 1985, and the Ph.D. degree in computer science from the Cadi Ayyad University in 2002. He is a professor of computer science at Cadi Ayyad University, where he has been since 1985. He is the author of over 50 papers on software engineering. His research interests are in software engineering and focusing on information system development

Mohammed Sadgal received the Ph.D. degree in computer science from the University of Lyon in 1989, and the Ph.D. degree in computer science from Cadi Ayyad University in 2005. From 1985 to 1987, he was an Associate Researcher with Lyon I, France. He is currently a Professor with Cadi Ayyad University, Marrakesh, Morocco. His research interests include computer vision, artificial intelligence, and multi-agent systems.

Received: February 27, 2022; Accepted: August 22, 2022.

Computer Science and Information Systems 20(1):423–457 https://doi.org/10.2298/CSIS220228063T

Combining Offline and On-the-fly Disambiguation to Perform Semantic-aware XML Querying

Joe Tekli[1], Gilbert Tekli[2], and Richard Chbeir[3]

[1] School of Engineering, ECE dept., Lebanese American University,
36 Byblos, Lebanon
joe.tekli@lau.edu.lb
[2] Faculty of Technology, Mechatronics dept., University of Balamand,
100 Tripoli, Lebanon
gilbert.Tekli@balamand.edu.lb
[3] LIUPPA Lab., IUT de Bayonne, University of Pau and Pays Adour
64000 Anglet, France
richard.chbeir@univ-pau.fr

Abstract. Many efforts have been deployed by the IR community to extend free-text query processing toward semi-structured XML search. Most methods rely on the concept of Lowest Comment Ancestor (LCA) between two or multiple structural nodes to identify the most specific XML elements containing query keywords posted by the user. Yet, few of the existing approaches consider XML semantics, and the methods that process semantics generally rely on computationally expensive word sense disambiguation (WSD) techniques, or apply semantic analysis in one stage only: performing *query relaxation/refinement* over the *bag of words* retrieval model, to reduce processing time. In this paper, we describe a new approach for XML keyword search aiming to solve the limitations mentioned above. Our solution first transforms the XML document collection (offline) and the keyword query (on-the-fly) into meaningful semantic representations using context-based and global disambiguation methods, specially designed to allow almost linear computation efficiency. We use a semantic-aware inverted index to allow semantic-aware search, result selection, and result ranking functionality. The semantically augmented XML data tree is processed for structural node clustering, based on semantic query concepts (i.e., key-concepts), in order to identify and rank candidate answer sub-trees containing related occurrences of query key-concepts. Dedicated weighting functions and various search algorithms have been developed for that purpose and will be presented here. Experimental results highlight the quality and potential of our approach.

Keywords: Semi-structured data, XML, Semantic Analysis, Semantic Disambiguation, Keyword Search, Query Processing.

1. Introduction

Various methods have been proposed for XML ranked retrieval. While most approaches consider content-and-structure features in specifying XML query constraints, few approaches have targeted semantic XML search based on simple keyword queries. Most approaches in this category exploit the concept of LCA (Lowest Common Ancestor)

between two or multiple structural nodes to identify the most specific XML elements containing query keywords posted by the user. Yet LCA-based methods underline various limitations: i) each result candidate must contain all query keywords, which is not always intuitive since a candidate result (element or sub-tree) containing most (and not necessarily all) keywords might be deemed relevant by the user; ii) some meaningful results might be missed: as XML trees underline different nesting hierarchies, restricting results to the LCA encompassing all keywords might miss some more general, and yet relevant results; iii) few of the proposed approaches consider semantics: for instance, when submitting sample keyword query "Universities in Sao Paulo", the user is probably interested in information concerning universities, academies and colleges in Sao Paulo, and cities in its vicinity such as Campinas, Sao Carlos, etc. Hence, semantic analysis becomes essential in such a context in order to improve search results; iv) the few existing methods that do target XML semantics generally rely on word sense disambiguation (WSD) and are computationally expensive, or v) apply semantic analysis in one stage only, performing query relaxation/refinement over the bag of words retrieval model, to reduce processing time.

In this paper, we introduce *XSemSearch*, a semantic-aware XML keyword search solution aiming to solve the limitations mentioned above. We propose to integrate semantic analysis and structural clustering in formulating an efficient solution to the problem. Our solution first transforms the XML document collection (offline) and the keyword query (on-the-fly) into meaningful semantic representations using context-based and global disambiguation methods, specially designed to allow almost linear computation efficiency. The semantically augmented XML data tree is processed for structural node clustering, based on semantic query concepts (i.e., key-concepts), in order to identify and rank candidate answer sub-trees containing related occurrences of query key-concepts. The overall architecture of our approach is depicted in Fig. 1

An initial description of *XSemSearch*'s architecture is given in [84]. This paper adds: i) a dedicated inverted index structure to handle semantically augmented XML data, ii) an dedicated query formalism to allow structure-and-content queries with only partial knowledge of the data collection structure and semantics, iii) two alternative query processing algorithms including *Query As You Type Search* and *Parallel Semantic Search*, and iv) an extended empirical study to evaluate query processing time and quality.

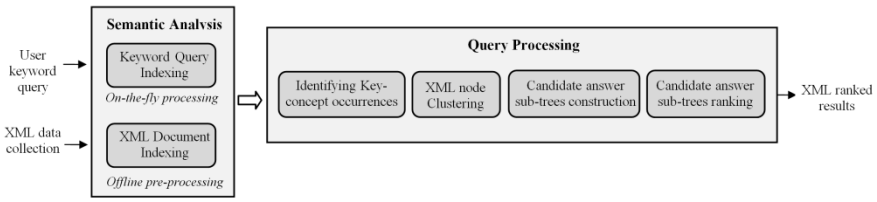


Fig. 1. Overall architecture of our XML semantic-aware search approach

The remainder of the paper is organized as follows. Section 2 reviews the background in XML and query semantic analysis. Section 3 provides an overview of our solution framework. Sections 4 and 5 respectively describe the XML semantic analysis and keyword query semantic analysis components. Section 6 describes the

query processing component. Section 7 provides experimental results, before concluding in Section 8.

2. Background

In this section, we review the background in semantic information retrieval, while focusing on XML and keyword query semantic analysis and disambiguation.

2.1. Semantic Information Retrieval

The retrieval model for an information retrieval system specifies how documents and queries are represented, and how these representations are compared to produce relevant result estimates [7]. A core problem in this context is lexical ambiguity: a word may have multiple meanings (homonymy), a word maybe implied by other related words (metonymy), and/or several words can have the same meaning (synonymy) [42].

The lexical ambiguity problem becomes even more acute on the Web, with the latter's heterogeneous and unstructured nature which makes it even more difficult to query and retrieve meaningful information. Semantic IR is part of the Semantic Web vision [76] that promises to solve the retrieval ambiguity problem, by i) associating terms in Web pages and queries with explicit semantics (i.e., word senses or concepts), and then ii) performing search functions based on document/query concepts rather than plain terms [55]. A core challenge in this context is word sense disambiguation (WSD): how to resolve the semantic ambiguities and identify the intended meanings of document terms and query keywords [11]. Various methods have been proposed for WSD in the literature [42, 53, 75]. They fall in two main categories: *corpus-based* WSD and *knowledge-based* WSD. The corpus-based approach is data-driven, as it involves information about words previously disambiguated and requires supervised learning from sense-tagged corpora to enable predictions for new words. Knowledge-based methods are knowledge-driven, as they handle a sense inventory and/or a repository of information about words that can be exploited to distinguish their meanings in the text. Machine-readable knowledge bases (e.g., dictionaries or semantic networks: thesauri, taxonomies, or ontologies) provide ready-made sources of information about word senses to be exploited in knowledge-based WSD. While corpus-based methods have been popular in recent years [6, 38], they are generally data hungry and require extensive training, huge textual corpora, and/or a considerable amount of manual effort to produce a relevant sense-annotated corpus, which are not always available or feasible in practice. Therefore, knowledge-based methods have been receiving more attention [16, 42]. In the remainder of our study, we focus on knowledge-based WSD and semantic analysis.

2.2. XML Semantic Analysis and Disambiguation

While a considerable amount of research has been undertaken around (knowledge-based) WSD in flat textual data [53], yet few approaches have been developed in the

context of XML and semi-structured information [75]. The main difference resides in the notion of XML (structural) contextualization. The context of a keyword, in traditional textual data, consists of the set of terms in the keyword's vicinity (i.e., terms occurring to the left and right of the considered keyword, within a certain predefined distance from the keyword [11]). However, there is no clear definition regarding the context of a node in an XML tree. The authors in [72, 73] consider the context of an XML data element to be efficiently determined by its parent element, and thus process a parent node and its children data elements as one unified (canonical) entity, using context-driven search techniques for determining the relationships between the different unified entities, so as to identify related semantic labels.

In [70, 71], the authors extend the notion of XML node context to include the whole XML root path, i.e., path consisting of the sequence of nodes connecting a given node with the root of the XML document (or document collection). They consequently perform per-path sense disambiguation, comparing every node label in each path with all possible senses of node labels occurring in the same path (using a gloss-based WordNet similarity measure [8]) in order to select the most appropriate sense for the label at hand. Different from the notions of parent context and path context, the authors in [85] consider the set of XML tag names contained in the sub-tree rooted at a given element node, i.e., the set of labels corresponding to the node at hand and all its subordinates, to describe the node's XML context. The authors apply a similar paradigm to identify to contexts of all possible node label senses in WordNet. Consequently, they perform label sense disambiguation by comparing the XML label context to all candidate sense contexts in WordNet, identifying the sense (semantic concept) with the highest similarity.

In [49], the authors combine the notions of parent context and descendent (sub-tree) context in disambiguating generic structured data (e.g., XML, web directories, and ontologies). The authors consider that a node's context definition depends on the nature of the data and the application domain at hand. They propose various edge-weighting heuristics (namely a Gaussian decay function) to identify *crossable* edges, i.e., nodes reachable from a given node through any *crossable* edge belong to the target node's context. Consequently, structure disambiguation is undertaken by comparing the target node label with each candidate sense (semantic concept) corresponding to the labels in the target node's context (using an edge-based semantic similarity measure [43], following the hypernymy/hyponymy relations in WordNet) in order to identify the highest matching semantic concept.

Another concern in XML-based WSD is how to effectively process the context of an XML node taking into account the structural dispositions of XML data. In fact, most existing WSD methods developed for flat textual data [42, 53], and those developed for XML-based data [70-73], follow the bag-of-words paradigm where the context is processed as a plain set of words surrounding the term/label (node) to disambiguate. In other words, all context nodes are treated the same, despite their structural positions in the XML tree. We encountered an approach in [49] which extends the traditional bag-or-words paradigm with additional information considering distance weights separating the context and target nodes (identified as *relational information model* [49]). The authors employ a heuristic Gaussian distance decay function estimating edge weights such that the closer a node (following a user-specified direction, e.g., ancestor, descendent, or both), the more it influences the target node's disambiguation [49]. The

semantic contribution of each context node is weighted by its position in the context graph of the target node.

2.3. Query Semantic Analysis and Disambiguation

Semantic query analysis in information retrieval usually involves two steps: i) WSD to identify the user's intended meaning of query terms, and ii) semantic query representation/expansion in order to alter the query so that it achieves better (precision and recall) results [67]. As described in the previous section, traditional semantic analysis and disambiguation techniques usually rely on the notion of context such that terms (e.g., node labels in the context of XML) that appear together in the same context have related meanings [11]. While context-based solutions are applicable with classic IR queries which are rather lengthy (e.g., 15 terms on average for short queries [91], reaching up to 50-85 terms for long queries [12]), nonetheless, keyword queries on the Web are usually 2-3 words long [15] which is generally insufficient in identifying a meaningful context [42, 49]. In fact, lexical ambiguity with Web search is often the consequence of the low number of query words entered on average by Web users [40]. Therefore, some sort of user interaction is usually required to counter the lack of contextualization, and more accurately identify the intended senses of Web query terms [24, 89].

Various methods for interactive keyword querying have been proposed in the literature, e.g., [41, 66] [30, 74]. Most existing approaches are *corpus-based* in that they expand user queries by adding words that co-occur with the query terms in a given corpora, i.e. words that, on a probabilistic ground, are believed to describe the same *semantic concept* (e.g. *car* and *driver*). Here, expansion terms are usually identified from i) user feedback: extracting frequent terms occurring in previous results deemed relevant by the user [41, 66], and/or ii) query logs: identifying frequent terms in the document collection based on the associations between past queries and the documents downloaded by the user [30, 74]. Yet, the extensive training and huge corpora requirements of *corpus-based* methods makes them less practical in the context of Web search applications, which has led to a growing interest in *knowledge-based* solutions [35, 61]. The latter family of methods investigates the use of ontological information to assist the user in formulating and/or expanding keyword queries by: i) allowing user interaction to identify the intended senses of query-terms, and then ii) expanding/modifying query keywords via their most related semantic concepts in the reference semantic source (e.g., WordNet) [67].

Following [14], a keyword query is first processed for lexical normalization, and then presented to the user as a set of lexical tokens, where each token is associated with a set of possible semantic meanings (identified using WordNet and/or domain specific ontologies). Consequently, the user is asked to select the most relevant sense for each lexical token. The system then exploits the selected user senses to reformulate the query using dedicated heuristics (e.g., replacing actual keywords via their synonyms with highest frequency of usage in WordNet, identifying negative keywords, i.e., the terms corresponding to the highest frequency synset remaining beside the one selected by the user, etc.), thus obtaining a semantically augmented keyword query. A similar approach is adopted in [45] with a special emphasis on failed-query reformulation. The authors in [45] assume that the reformulation of a failed query without help from the system can

be frustrating to the user, and thus suggest to assist the later by proposing semantically meaningful keywords selected from WordNet (using heuristics similar to those adopted in [14]). The method in [45] is developed in the context of the NALIX project for building an interactive natural language interface for querying XML [44].

A fully automated approach to *knowledge-based* query disambiguation is introduced in [54], where the authors exploit structural pattern recognition [25] in mapping query keyword senses. The proposed method creates a local semantic network for each keyword-sense in the query, including most semantic relations utilized in WordNet [28] (hypernymy, hyponymy, meronymy, etc.). Then, for each possible configuration of senses, the system identifies the intersections between corresponding pair-wise local semantic networks using an adapted structure pattern recognition algorithm. Common nodes are those that can be reached through both semantic networks being compared. The configuration with the highest intersection score (i.e., highest number of intersecting nodes) is selected as the one encompassing the most relevant keyword senses. In a subsequent step, the authors propose various heuristics to expand the query using synset, hyponymy and/or gloss information. Experimental results in [54] show a 26.85% improvement in retrieval precision over the plain query words.

Note that most existing studies targeting *knowledge-based* query semantic analysis, e.g., [35, 61] [14, 45, 56], do not evaluate the complexity (or execution time) levels of their proposed methods. Nonetheless, time complexity is critical for on-the-fly execution on the Web (in comparison with document semantic analysis which could be performed offline). The time complexity of query semantic analysis might even prove to be problematic in the case of the pattern recognition-based methods [19, 54], since traditional structure pattern recognition problems are usually of exponential complexity [25, 58].

3. Proposal Overview

Semantic similarity evaluation between two terms usually consists in looking up each term's lexical concept in a reference knowledge base (e.g., a semantic network such as WordNet), and consequently comparing the underlying concepts. Nonetheless, semantic similarity evaluation has been proven to be an expensive task: comparing two semantic concepts following one of the most prominent semantic similarity measures in the literature, i.e., [47], requires $O(|SN| \times Depth(SN))$ time where $|SN|$ is the size (i.e., cardinality in number of concepts) of the reference semantic network SN, and $Depth(SN)$ its maximum depth. Evaluating the semantic similarity between query keywords and each label/term in the XML document collection becomes extremely complex, and practically unfeasible.

A way of getting round the complexity problem would be to perform semantic analysis of the XML document collection, offline, and prior to the retrieval phase. This consists in transforming the XML documents into weighted semantic trees (graphs), and transforming and expanding the keyword query into a set of weighed semantic concepts. Consequently, an adapted XML IR engine (cf. Section 6) processes the semantically indexed documents and queries, so as produce more meaningful results. Our semantic analysis processes are depicted in Fig. 2.

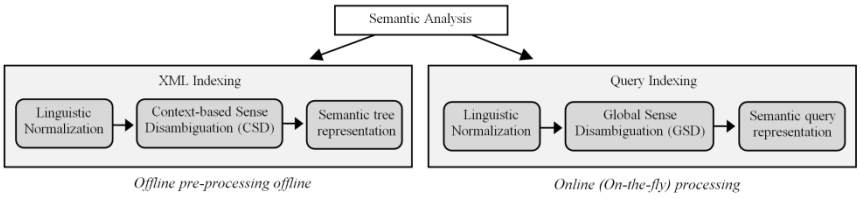


Fig. 2. Semantic analysis of XML document and keyword query

Note that while semantic query indexing is performed online, XML document indexing is performed offline, and does not affect the complexity of the approach. As shown in Fig. 2, semantic indexing consists of three main phases: i) Linguistic Normalization, ii) Sense Disambiguation, and iii) Semantic Representation. While the first phase (Linguistic Normalization, including *tokenization*, *expansion*, *stop word removal*, and *stemming*) is similar for both document labels and query keywords, yet, we design the latter two (sense disambiguation, and semantic representation) differently following the data models and requirements at hand. Sense disambiguation usually relies on the notion of context, where terms that appear together in the same context have related meanings [11]. While context information is available for XML document nodes (e.g., the context of a node could be its parent node, its root path, the whole document tree containing the node, etc.), yet, keyword queries on the Web are usually two-to-three words long [15] which is generally insufficient in identifying a meaningful context [42, 49]. Hence, we introduce two different methods for document and query sense disambiguation: i) Context-based Sense Disambiguation (CSD) for XML documents, ii) Global Sense Disambiguation (GSD) for keyword queries.

In the following, Sections 4 and 5 present the XML document semantic analysis and the keyword query semantic analysis processes respectively

4. Semantic XML Document Analysis

Our XML document semantic analysis process consists in: i) disambiguating each label following its context, to associate each label with the proper semantic concept in the reference knowledge base (e.g., WordNet), and ii) producing a semantically indexed XML tree, with the corresponding index structures and pair-wise concept semantic weights, to be consequently utilized in the query processing task. We describe the latter in the following sub-sections.

4.1. XML Data Model

XML documents represent hierarchically structured information and are generally modeled as ordered labeled trees (cf. Fig. 3). In a traditional DOM (Document Object Model) ordered labeled tree [87], nodes represent XML elements, and are labeled with corresponding element tag names, ordered following their order of appearance in the document. Attributes usually appear as children of their encompassing element nodes, sorted by attribute name, and appearing before all sub-element siblings [57, 93]. Other

types of nodes, such as entities, comments and notations, are commonly disregarded in most XML comparison approaches, e.g., [22, 36], since they underline complementary information and are not part of the core XML data.

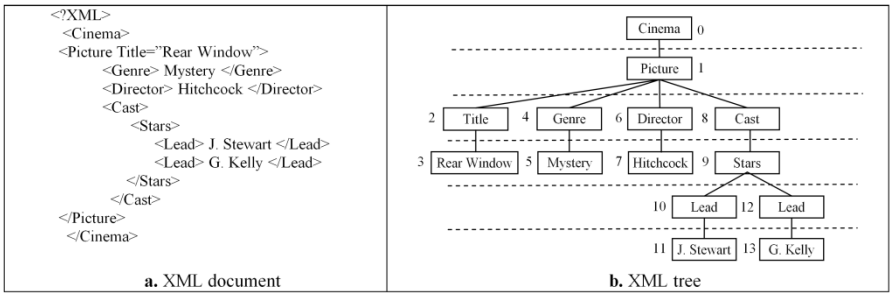


Fig. 3. A sample XML document with corresponding tree

In general, element/attribute values are disregarded when evaluating the structural properties of heterogeneous XML documents (originating from different data-sources and not conforming to the same grammar), so as to perform XML structural classification/clustering [22, 36, 57, 59] or structural querying (i.e., querying the structure of documents, disregarding content [10, 63]). Nonetheless, values are usually taken into account with methods dedicated to XML change management [18, 20], data integration [32, 46], and XML structure-and-content querying applications [64, 65], which is the main application in our current study. More formally:

Definition 1 - XML Tree: We represent an XML document as a rooted ordered labeled tree $T = (N_T, E_T, L_T, \Delta_T, g_T)$ where N_T is the set of nodes in T, $E_T \subseteq N_T \times N_T$ is the set of edges (element/attribute containment relations), L_T is the set of labels corresponding to the nodes of T ($L_T = El_T \cup Ev_T \cup Al_T \cup Av_T$ such as El_T (Al_T) and Ev_T (Av_T) designate respectively the labels and values of the elements and attributes of T), Δ_T is the set of data-types associated to the elements and attribute nodes of T (Δ_T= {*Concept*} $\cup$ $E\Delta \cup A\Delta$, having $E\Delta = A\Delta$ = {*Text*, *Number*, *Date*}), and g_T is a function $g_T : N_T \rightarrow L_T, \Delta_T$ that associates a label $l \in L_T$ and a data-type $t \in \Delta_T$ to each node $n \in N_T$. We denote by *root*(T) the root node of T, and by $T' \blacktriangleright T$ a sub-tree of T ●

Value data-types in the XML tree model are extracted from the corresponding XML schema. In other words, during XML tree construction time, the XML document and corresponding schema are assessed simultaneously so as to build the XML tree. Textual values are treated for stemming and stop word removal, and are mapped to leaf nodes of type *Text* in the XML tree. Numerical and date values are mapped to leaf nodes of types *Number* and *Date* respectively. As for the disambiguated element/attribute nodes, they are assigned the data-type *Concept*, their labels corresponding to the semantic concepts defined through the reference knowledge base. To model the XML data repository, we connect all XML trees to a single root node, with a unique label (e.g., *'Root'*).

4.2. Semantic Knowledge Representation and Indexing

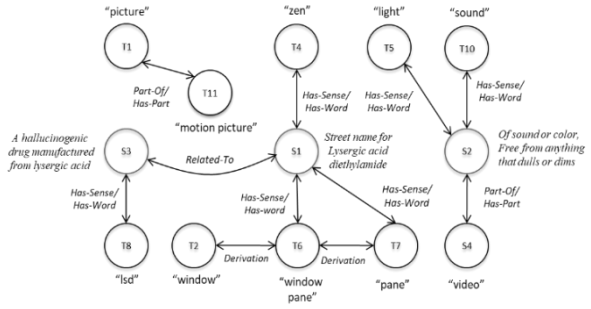
Semantic knowledge bases (i.e., thesauri, taxonomies, and/or Ontologies such as WordNet [51], Roget's thesaurus [90], and Yago [37]) provide a framework for organizing words/expressions into a semantic space [13]. A knowledge base is usually modeled as a semantic network made of a set of entities representing semantic concepts (or groups of words/expressions), and a set of links between the entities, representing semantic relationships (*synonymy*, *hyponymy*, etc.). We adopt a graph-based structure to model a semantic network from, where entities are represented as vertices, and the semantic relationships between entities are represented as directed edges. Formally:

Definition 1 — Semantic Network: A semantic network is represented as a graph SN (V, E, L, f_V, f_E) where:

- V is a set of vertices (nodes), designating entities in the semantic network. V includes both: i) *sense* nodes, representing semantic senses (*synsets*) with glosses, and ii) *term* nodes, representing literal words/expressions.
- E is a set of directed edges, an edge consisting of an ordered pair of vertices in V.
- L is a set of edge labels denoting semantic/lexical relationships. For WordNet, L includes: semantic relationships between concepts (e.g., *hyponymy*, *hypernymy*, *meronymy*), semantic relationships between concepts and terms (e.g., *has-sense* and *has-term*), and lexical relationships between terms (e.g., *derivation*).
- f_V is a function defined on V, designating the string value of each node in V. For WordNet, string values include: i) glosses/definitions, when dealing with *sense* nodes, and ii) and literal words/expressions,
- f_E is a function defined on E, assigning a label from L to each edge in E. Multiple edges may exist between the same pair of vertices when dealing with *term* nodes, which makes SN a multi-graph •

An extract from the WordNet lexical ontology is shown in Fig. 4, where S_1, S_2 and S_3 represent senses (i.e., synsets), and their string values (i.e., the synsets' glosses/definitions), and T_1, T_2, …, T_{11} represent terms, and their string values (i.e., literal words/expressions) shown along aside the nodes. Given that most semantic/lexical relationships are symmetrical (*hyponymy*/*hypernymy*, *meronymy*/*holonymy*, *has-sense/has-term*, etc.), and given that a relationship cannot exist without its symmetrical counterpart, we simplify our graph model by representing each couple of symmetrical relationships between senses and/or terms with one edge having opposite directions (instead of two edges), labeled with the names of the symmetrical relationships.

A simple inverted index *InvIndex*(*SN*) can be subsequently built for the textual tokens of each *SN* entity (i.e., string values of *term* nodes and *sense* nodes, cf. Fig. 4b) to speed up term/sense lookup when creating and then querying the XML structure.



a. Sample graph *SN* representing an extract from WordNet

Term	Sense IDs[]
"acid"	⟨S1, S3⟩
"clean"	⟨S2⟩
"light"	⟨S2⟩
"lsd"	⟨S3⟩
"lysergic"	⟨S1, S3⟩
"window pane"	⟨S1⟩
…	…

b. Extract of inverted index *InvIndex*(*SN*) connecting terms in *SN* with corresponding senses (to speed up term/synset lookup)

Fig. 4. Extract from the semantic graph of WordNet, with the corresponding index

4.3. XML Context-based Sense Disambiguation

Our XML sense disambiguation approach was introduced in [17, 78]. Here, we only provide an overview of the approach describing the constructs and methods required in our current study.

Different from previous approaches which limit XML context to the parent node [72, 73], to the root node path [70, 71], to the node sub-tree [85], or to nodes reachable through heuristically identified crossable edges [49], we introduce the notion of XML *Sphere-Ring* context, inspired from the sphere-search paradigm in XML IR [31], to consider the whole structural surrounding of an XML node, including its ancestors, descendants, and siblings, tuned to better describe the node's context. An XML *ring* w.r.t. to a given node consists of the set of nodes situated at a specific distance from the center node. An XML sphere encompasses all rings contained at distances lesser or equal to the size (diameter) of the sphere. The size of the XML sphere is tuned following the nature of the XML data at hand (e.g., certain XML trees might underline specialized and domain-specific data, and thus would only require small contexts so as to achieve relevant WSD results, whereas more heterogeneous and generic XML data might require larger contexts to better describe the intended meaning of each node label).

In addition, we extend the traditional bag-of-words WSD paradigm, adopting a relational information approach, i.e., considering the interconnections among XML nodes in computing disambiguation scores (in contrast with the classic bag-of-words approach [70-73], where all context nodes are treated as a homogeneous set of words regardless of their proximity/relations with the target node). We consider the structural distance separating the center node and each of its context nodes, following the intuition that the farther the context node from the sphere center, the lesser should be its impact in determining the semantic meaning of the center node label. Formally, consider $R_d(n)$ to be the ring corresponding to the center node n at distance d, i.e. the set of all nodes whose distance from n is d. Hence, the context sphere $S_D(n)$ of node n, with size D, consists of all the rings contained in $S_D(n)$, such that $S_D(n) = \{\text{all } R_d(n) \,/\, d \leq D\}$. Following our *Sphere-Ring* context model, node scores can be weighted following the sizes of the sphere rings to which they correspond, such that the larger the sphere ring

radius, the lesser the node weight. Hence, we can represent the context of a node *n* as a weighed vector, whose dimensions correspond to the all distinct nodes in its sphere context, weighted following their distances from the center node. In short, our approach:

- Integrates all notions of XML context, including ancestor, decedent, and sibling structural relations, which were considered separately in existing studies [70-73, 85],
- Allows the user/system administrator to manually and/or automatically tune the size of the XML context window following the nature and properties of the XML data at hand, in comparison with most existing static methods [70-73, 85],
- Extends the traditional bag-of-words WSD paradigm, adopting a relational information approach so as to consider the interconnections among XML nodes in computing disambiguation scores, in contrast with most existing methods using the traditional bag-of-words approach [70-73].

Once the contexts of all XML nodes have been determined, we process each target node label and its context node labels for WSD. Here, we evaluate the semantic similarity/relatedness between the target node label and each of its context node labels, by comparing the node's context with the context of each of its potential senses, extracted from the reference semantic source (a similar paradigm is utilized in [85] for XML node annotation). The idea is to first identify all possible senses of the target word node label in the reference semantic network. Consequently, we exploit the same notion of *Sphere-Ring*, which we adopted for XML trees (graphs), to identify the context of each potential sense in the reference semantic network (e.g., WordNet). Having computed the weighted context for the XML target node in the XML document tree (graph), and each of its possible senses in the semantic network, we compute the similarity between the node vector and each of its sense vectors. The sense vector yielding the highest similarity would underline the most meaningful sense describing the XML node label. This approach requires polynomial complexity: $O(|senses(x.\lambda)|\times(|S_D(n)|+|S_D(s_p)|))$, where $|S_D(s_p)|$ designates the maximum context sphere cardinality for any sense (concept) in the semantic network.

Note that to our knowledge, existing approaches have seldom provide a complexity and time performance analysis of their WSD methods. Despite being performed offline, nonetheless, WSD time performance remains potent w.r.t. practicability, when indexing documents published on Web. The proposed approach has to be: i) effective in identifying the correct senses, but also ii) reasonably efficient in order to be practically applied to the large corpora of XML documents published online. Here, the complexity of our combined XML sense disambiguation approach is polynomial and simplified to $O(|X|\times|senses(x.\lambda)|\times(|S_D(n)|+|S_D(s_p)|))$, where $|X|$ represents the number of nodes to be disambiguated in the target XML document.

4.4. XML Document Semantic Indexing

Having disambiguated all XML labels, the latter are replaced with their corresponding semantic concepts extracted from the reference semantic network (e.g., WordNet). Dedicated index structures (Concept-Doc and Concept-SN indexes [79-81], cf. Fig. 5) are utilized to handle the mapping between XML document labels and semantic network concepts. The output of the semantic document indexing process is a conceptual XML tree, i.e., an XML tree which labels consist of concepts with explicit

semantic definitions (which is at the core of the vision of the Semantic Web: Extending the WWW by giving information well defined meaning [76]).

Given an data collection *C*, an inverted index (also referred to as a posting file, or inverted list) built upon *C*, is (in its most basic form) a sorted list of index terms associated each with a set of object identifiers from *C*, disregarding structural information. In this study, we extend the basic inverted index to handle semi-structured data elements, introducing an *element-attribute* (*OA*) index:

Definition 2 - Element-Attribute (EA) Inverted Index: Given a XML data collection *C*, an *EA* inverted index built on *C*, denoted as $InvIndex_{EA}(C)$, is a structure of the form (*dom*(*A*), *EAs*, *f*) where:

- *dom*(*A*) designates the set of values within the domains of all attributes $\forall A_j \in C.A$. Considering text-only domains, values come down to textual tokens, i.e., *terms* (words/expressions),
- *EAs* designates the set of element (identifier)-attribute doublets, i.e., $EAs = \{(id(E_i), A_j)\} \ \forall E_i \in C$ and $\forall E_j \in C.A \ / \ \exists E_i.a_j \neq \emptyset$, where A_j is an attribute for which object E_i has a non-null value,
- *f* is a function mapping each $term \in dom(A)$ with a list of element-attribute doublets *EAs*[] designating the term's occurrence locations in *C*, i.e., $EAs[] = \langle (id(E_i), A_j) \rangle \ / \ term \in E_i.a_j$

A term used as textual token in the inverted index is referred to as index term, whereas the list of element-attribute doublets, i.e., *EAs*[], mapping to each index term is referred to as the term's posting list •

Consequently, we compute the semantic relatedness between each pair of node concepts in the XML tree. The idea is to produce a semantically weighted XML tree to be consequently exploited in keyword query processing (cf. Section 6). Here, various semantic similarity measures can be used (as briefly mentioned in the previous section): i) edge-based measures (computing semantic similarity based on the distance separating the concepts in the semantic network) [88], ii) node-based (computing semantic similarity based on the information content of each concept in the semantic network, w.r.t. a given text corpus) [47], and iii) gloss-based (comparing the glosses associated with each concept definition in the semantic network) [8]. Gloss-based approaches are particularly interesting in the context of WSD since they allow 'semantic relatedness' evaluation, which is a more general notion than '*semantic similarity*', including the latter as well as any kind of functional relation between terms [39] (e.g., *penguin* and *Antarctica* are not necessarily similar, but they are semantic related due to their *natural_habitat* connection), particularly *antonymy* (e.g., *hot* and *cold* are semantically dissimilar since they have opposite meanings, but they are semantically related).

A simple example depicting the semantic indexing of a sample XML tree is shown in Fig. 5. The sample XML document describes the movie *Rear Window*, one of *Alfred Hitchcock*'s masterpieces. While the XML labels seem meaningful and straightforward for a human user, nonetheless, they are highly ambiguous for a computer system. Most labels can be associated with more than 2 or 3 semantic senses (concepts) in WordNet reference. For instance, the label *Stewart* is associated with 2 semantic concepts: i) *James Stewart* (the leading actor who starred in *Rear Window*), and ii) *Dugald Stewart* (an 18th century Scottish philosopher). Likewise for most remaining labels in the input tree (e.g., *Kelly* underlines 3 semantic concepts, among which is *Grace Kelly*, the co-

star of *Stewart* in *Rear Window*; *plot* underlines 4 different senses, among which *movie plot*, etc.).

Recall that semantic XML document indexing is performed offline, as a pre-processing step prior to query evaluation, and does not affect the online computational complexity of the approach.

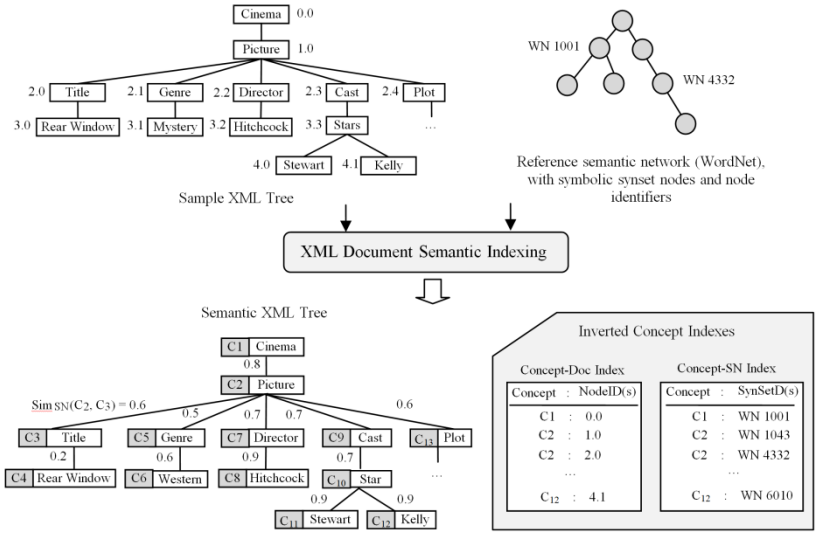


Fig. 5. Semantic analysis of XML document

5. Semantic Query Analysis

While semantic XML document analysis relies on the notion of XML context (e.g., the surroundings of a given node) in identifying the meanings of XML labels, nonetheless, semantic keyword query analysis differs in the lack of sufficient contextualization (keyword queries on the Web are usually 2-3 words long [15], which might not be sufficient in identifying a meaningful context [42, 49], cf. background in Section 2). To get round the lack of keyword contextualization in identifying meaningful query keyword senses, we introduce a method to *global query sense disambiguation*. Our proposal is based on the following assumption: *A keyword query on the Web usually conveys a certain global semantic meaning, reflecting a certain global information need*. Hence, rather than analyzing the individual senses of each query-term separately, considering each term's context information (similarly to most existing approaches, e.g., [35, 61]), we evaluate the aggregate semantic meaning of the query as a whole such that: the higher the semantic homogeneity of the query, the higher the consistency of the unified global semantic meaning conveyed by the query, and thus the more likely the query reflects the user's need. This is in accordance with the traditional assumption in WSD: *the most plausible assignment of senses to multiple co-occurring words is the one that maximizes the relatedness of meaning among the chosen senses* [52].

In short, we disambiguate the query as a whole, by i) pinpointing all possible configurations of query-term senses, and ii) consequently estimating a global semantic relatedness score (given a reference information source, e.g., WordNet) for all senses combined in each configuration. The configuration with the highest score would underline the most semantically meaningful query. Global query sense ranking can also be performed to identify the top most meaningful query sense configurations.

A major problem with the above approach is its computational complexity. In fact, computing semantic similarity/relatedness for all possible sense configurations for a set of lexical terms was shown to be intractable [52] due to its best case exponential complexity (i.e., $O(senses(k)^N)$ where N is the number of query keywords, and $senses(k)$ is the maximum number of senses per keyword). A few approximation methods have been proposed, such as computing pair-wise keyword similarities [52], and evaluating the similarity between each keyword sense and all remaining node senses [9]. Nonetheless, in contrast with existing approximation solutions, e.g., [9, 60], we introduce a sense disambiguation method to solve the computational complexity described above, producing optimal results similarly to the initial (exponential complexity) approach, while confining to polynomial complexity. We do so by transforming the problem of identifying all possible sense configurations, into that of identifying the shortest (semantic) path in a (semantically) weighted graph, using an adaptation of Dijkstra's shortest path algorithm [21]. In short, we capitalize on Dijkstra's polynomial computation approach to eliminate all unnecessary similarity computations, while still considering all possible query sense configurations.

Our query semantic analysis approach is described in the following sub-sections. Sub-section 5.1 presents our global query sense disambiguation approach, while Sub-section 5.2 describes our semantic query representation method. Recall that linguistic normalization (including *tokenization*, *expansion*, *stop word removal*, and *stemming*) is similar for both XML documents labels and query keywords, and will not be discussed hereunder

5.1. Structure-and-Content Query Model

In addition to the keyword query model, we put forward a structure-and-content query model to allow a higher level of expressiveness in querying semi-structured XML data. We suggest a simple model consisting of an XML tree variant with special leaf nodes to represent query predicates. A query with an *Or* logical operator is decomposed into a *disjunctive normal form* [64], and is thus represented as a set of XML trees, corresponding to the set of conjunctive queries.

Definition 3 – Structure-and-Content Query: It is expressed as an XML tree, $Q = (N_Q, E_Q, L_Q, T_Q, g_Q, n_d)$ encompassing a *distinguished* node n_d underlining the matches in the data tree that are required as answers to the query (i.e., the query's return clause). The query's root node $R(Q)$ designates its search scope/context. Its set T_Q encompasses the node type for distinguishing disambiguated XML nodes, and predicate types P_t_i corresponding to every value data-type t_i considered in the data model (e.g., $T_Q = \{Concept\} \cup \{P\_Text, P\_Number, P\_Date\}$) ●

Definition 4 - Query Node: It is a XML tree node with additional properties to represent predicates. With $n.t = P\_t_i$ (predicate corresponding to data-type t_i), the node's

label *n.l* underlines a composite content made of the predicate operator *n.l.op* and value *n.l.val* (e.g., leaf node $Q_1[2]$ of query Q_1 in Fig. 6 is of $Q_1[2].l.op$ = *'<'* and $Q_1[2].l.val$ = *'1965'*, having $Q_1[2].t$ = *P_Date*, which underlines that the predicate value *'1965'* is of type *Date*) ●

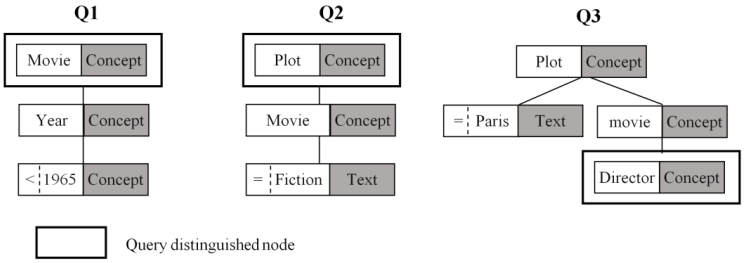


Fig. 6. Sample XML query trees

Note that each data-type has its own set of operators (e.g., {=, <, ≤, >, ≥, ≠} for numbers and dates, and {=, *like*} for text). Sample query trees are depicted in Fig. 6. Recall that query trees can be constructed via a dedicated GUI, which would suggest, on-the-fly, the list of possible query nodes following the context of the query at hand.

Definition 5 – Predicate Satisfaction: Given a predicate XML query node q_i, and a data node s_j, s_j satisfies q_i ($s_j \models q_i$) if:

- The data node type corresponds to that of the query ($s_i.t \approx q_i.t$, i.e., $\forall\, t_r \in$ {Text, Number, Date}, $q_i.t = P\_t_r \wedge s_j.t = t_r$),
- The data node label $s_j.l$ verifies the logical condition defined by $q_i.l$ ●

For instance, leaf node $T[13]$ of data tree T_2 in Fig. 3, having $T_2[13].l$ = *'1954'* and $T_2[13].t$ = *'date'*, satisfies predicate node $Q_1[2]$ of query Q_1 in Fig. 6, with $Q_1[2].l$ = *'<1965'* and $Q_1[2].t$ = *'date'*.

Definition 6 - Query Scope: Given a structure-and-content query *Q*, the scope of *Q* is identified by its root node *R*(*Q*), and corresponds to the XML sub-trees, in the data collection, having identical or semantically similar root nodes as that of the query ●

We assume that the user defines, with the query, the kind of XML data she is looking for, i.e. the scope/context of her query. If for instance the root of the query is labeled *University*, then XML data in the context of XML data entity *University*, or semantically similar entities such as *College*, *Academy*, etc., would naturally interest the user.

Definition 7 - *Template* and *Minimal constraint* querying: An structure-and-content query *Q* could be either evaluated as a i) *template* of the XML data the user is searching for, ii) or could represent the *minimal constraints* the data should meet to belong to the query answer set. In the former case, all query and data nodes would be considered in query/data similarity evaluation. Following the latter strategy, only elements required by the query tree are taken into account in query/data similarity evaluation, additional elements in the data tree being disregarded in the evaluation process ●

Note that XML queries most likely follow the *minimal constraint* style, the user usually specifying her information needs in the simplest form possible (cf. queries Q_1,

Q_2 and Q_3 in Fig. 6). Nonetheless, *template* querying could be particularly useful in *search-by-document* and *search-by-image* systems for instance, where the query could be a whole document or an SVG image [62] the user is searching for in the XML repository. A *template* style query could be any of the sub-trees in the XML tree of Fig. 3.

5.2. Global Query Sense Disambiguation

As mentioned previously, we assume that a query on the Web conveys a certain global semantic information request. The main objective is to associate each query-term with the appropriate semantic sense (concept) maximizing global query sense homogeneity. To do so, we proceed as follows:

Step 1 – Identifying Keyword Senses: The first step consists in identifying the set of possible senses corresponding to each individual query-term (keyword). Formally, for each keyword k_r, we obtain a set of senses $S_r = \{s^r_1, s^r_2, \ldots, s^r_{|Sr|}\}$ where s^r_i underlines the ith possible sense of keyword k_r extracted from the reference semantic network (e.g., WordNet), and $|senses(k_r)|$ the maximum number of possible senses corresponding to k_r. This first step is similar to most existing semantic based approaches, and the process is applied to structure-and-content queries.

Step 2 – Building the Semantic Query Graph: Having identified all possible senses for each query-term, we construct a semantic graph where each node represents of a possible keyword sense. The graph is structured in different layers, such that:

i. Each layer corresponds to a query-term, and consists of nodes representing all possible semantic senses for that query-term,
ii. The layers are ordered following the order of appearance of the query-terms in the keyword query,
iii. Nodes within the same layer (i.e., representing possible senses for the same term) are not connected to each other. In fact, same layer nodes underline senses of the same query-term and thus should not appear simultaneously in the same path (i.e., same query sense configuration),
iv. Each pair of nodes corresponding to two consecutive layers (i.e., describing the possible meanings of two consecutive query-terms), are connected together via a weighted edge, underlining the semantic distance (as an inverse function of semantic similarity/relatedness) between node senses,
v. Two virtual *start* and *end* nodes are added to the graph, connected to the nodes of the first/last graph layers respectively, via edges of null distances. These are introduced to guide the execution process of our adapted shortest path discovery algorithm (described hereunder),
vi. With content-and-structure queries, the query tree structure is considered when ordering the query nodes.

Step 3 - Identifying the Shortest Semantic Path: Consequently, the problem of identifying the most homogeneous configuration of query-term senses, simplifies to that of identifying the shortest semantic path in the semantic query graph. Here, we introduce an adaptation of Dijkstra's famous shortest path algorithm [21]. Our approach can be summarized as follows:

i. Initialize node distance scores such that: the *start node* score is set to zero, and all other node scores are set to infinity,
ii. Mark all nodes as *unvisited*, and set the *start node* as *current node*,

iii. For current node n_c, calculate the semantic distance with each of its connected nodes n_j in the consecutive layer, and preserve minimum distance scores, i.e., for each n_j, $Dist(n_j) = Min\{ Dist(n_c) + Weight(Egde(n_c, n_j)), Dist(n_j) \}$,
iv. When scores for all nodes connected to the current node n_c have been computed, n_c is marked as *visited*. A visited node would have a minimal and final distance score,
v. Select the *unvisited* node with the smallest distance score (from the initial node, considering all nodes in the graph) as the *current node* and continue from step 3,
vi. Terminate the algorithm when *end node* is deemed *visited*.

Consider keyword query *'Stewart Mystery Films'* (a similar process is applied to structure-and-content queries). The corresponding semantic query graph, built based on query-term semantic senses extracted from WordNet [50], is depicted in Fig. 7. Each graph layer corresponds to a query-term, and each node in a given layer underlines a semantic sense (concept) corresponding to the term at hand. The weight of an edge underlines the semantic distance between the connected nodes. Semantic distance can be computed as an inverse function of semantic similarity/relatedness, e.g., $Dist_{Sem} = 1 - Sim_{Sem}$. Recall that we adopt an aggregate semantic similarity/relatedness function combining *edge-based* methods [88], *node-based* methods [47], and *gloss-based* methods [8], w.r.t. WordNet. For ease of presentation, Fig. 7 shows sample semantic weight values for some (and not all) of the graph edges (e.g., $weight(edge(n_1, n_4)) = 0.3$ indicating that semantic concepts *James Stewart* and *Mystery story* are more similar than *James Stewart* and *Enigma*, having $weight(edge(n_1, n_3)) = 0.5$).

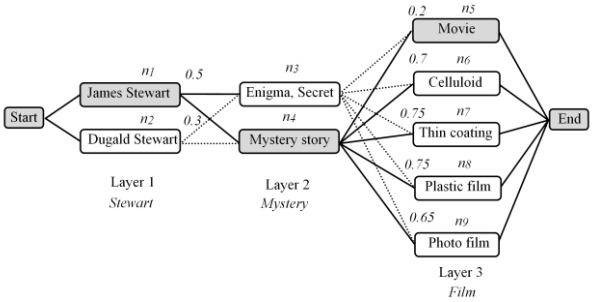


Fig. 7. Semantic analysis of keyword query

The result of applying our adapted shortest path algorithm to the semantic query graph in Fig. 7 is highlighted in the graph, and consists of nodes: n_1, n_4 and n_5. These underline the (WordNet) semantic concepts maximizing global query sense homogeneity: *James Stewart*, *Mystery story*, and *Movie*.

5.3. Semantic Query Representation

Having identified the best (i.e., most homogeneous) query sense configuration, we represent the query as a set of weighted semantic concepts (i.e., key-concepts), allowing the user to semantically expand the query, including additional concepts related to those originally conveyed by the query, in order to improve search result precision/recall.

Formally, a user keyword query Q consisting of a sequence of lexical keywords k_r, $Q = \{k_1, k_2, \ldots k_N\}$ is transformed into a semantic query representation Q_{Sem} consisting of a set of weighted concepts, $Q_{Sem}(D)= \{(c_1, w_1), (c_2, w_2), \ldots, (c_M, w_M)\}$ where c_i is a key-concept, w_i is the weight of c_i, and D is the query semantic depth parameter. The number of resulting key-concepts $M \geq N$ since additional key-concepts can be added following the user-chosen D expansion parameter as explained in the following. Semantic query expansion is performed using our *Sphere-Ring* model (cf. Section 4.1) to consider the semantic context of each query key-concept in the reference semantic network (e.g., WordNet). Note that the semantic contexts of query concepts can be determined, since the latter have already been disambiguated (as opposed to the pre-disambiguation keyword query where the semantic meanings of query-terms were undefined). The idea is to expand the query with additional concepts within the semantic vicinity of the original query key-concepts. Following our *Sphere-Ring* model, a semantic *ring* $R_d(c)$ w.r.t. to a given concept c consists of the set of concept nodes, in the reference semantic network, situated at a specific distance d from the target concept node c. The semantic context sphere SD(ci) encompasses all semantic rings contained at distances lesser or equal to the size (diameter D) of the sphere, such that SD(c) = {all Rd(c) / d ≤ D}. The sphere context size is specified by the user as a query semantic depth parameter:

- For $D = 0$, the query is represented with it original key-concepts, associated maximum (unit, =1) weights,
- For $D > 0$, the query is expanded with concepts situated within each original key-concept's semantic sphere (in the reference semantic network). Expanded query concepts are weighted such that concepts farther away from the semantic sphere center have a larger semantic distance w.r.t. the sphere's center, and hence should have a lesser impact on the query's semantic meaning. Following our *Sphere-Ring* context model, concept weights can be computed following the sizes of the sphere rings to which they correspond, such that the larger the sphere ring radius, the lesser the concept weight (e.g., a given weight decay function could be computed as $weight(c_i) = w_i = \frac{1}{1+d} \in [0, 1]$ having $c_i \in R_d(c) \subset S_D(c)$). Note that parameter D can be normalized in the $[0, 1]$ interval, following the maximum depth of the reference semantic network SN at hand (e.g., $\frac{D}{Depth(SN)}$), to simplify the user's task in specifying the expansion threshold.

Consider for instance the sample keyword query Q = *'Stewart Mystery Films'*:

- For D = 0, $Q_{Sem}(0)$ = {(James Stewart, 1), (Mystery story, 1), (Movie, 1)},
- For $D = 1$, the resulting query representation includes all semantic concepts appearing in the unit ($D=1$) semantic context spheres of each original key-concept. Here, following the WordNet extracts in Fig. 8, the semantic context of concept *James Stewart* includes concept *Actor* (cf. Fig. 8.a). Likewise, the semantic context of concept *Mystery movie* includes *Story*, *Detective story* and *Murder story* (Fig. 8.b). The semantic context of concept *Movie* includes *Show*, and *17* children (hyponym) concepts including *Telefilm*, *Feature film*, *Final cut*, *Home movie*, etc., (the remaining child concepts are omitted here for ease of presentation, cf. Fig. 8.c). The weights of all expanded concepts are equal to $\frac{1}{1+d} = \frac{1}{1+D}$ = 0.5, following our adopted decay function. Hence, the semantic query becomes:

$Q_{Sem}(1)$={ (James Stewart, 1), (Actor, 0.5), (Mystery story, 1),
(*Story*, 0.5), (*Detective story*, 0.5),
(*Murder story*, 0.5), (*Movie*, 1), (*Show*, 0.5),
(*Telefilm*, 0.5), (*Final cut*, 0.5), (*Home movie*, 0.5) }

The time complexity of our global query disambiguation approach comes down to that of the shortest path computation process, which comes down to almost linear $O(N\times\log(N))$ time where $N=|S_D(c)|\times|Q|\times|senses(k_r)|$. The latter simplifies to $N=|S_D(c)|\times|senses(k_r)|$ since $|Q|$ is usually limited to 2-3 keywords [15] and can be omitted as a fixed parameter.

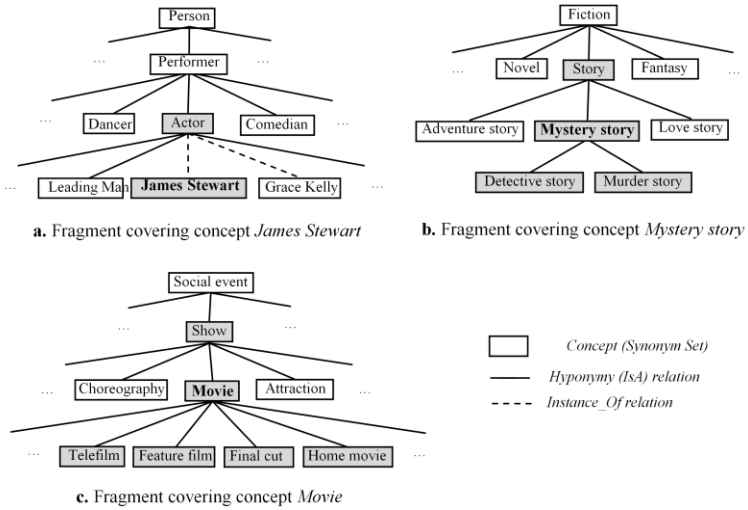


Fig. 8. Taxonomy fragments extracted from WordNet, covering the key-concepts in our example

6. Semantic Query Processing

6.1. Candidate Answer Tree

The first step in assessing a query is to identify its search scope. Following the traditional IR logic, whole physical files are considered as candidate answers. Nonetheless, XML documents differ in their structural organization and granularity: some documents may contain information about *movies*, while others include information about *actors* acting in many *movies*. Hence, it is not relevant to retrieve the entire *movie* when the user is searching for certain *actors*. Hence, the XML query search scope should be identified dynamically, considering the query at hand.

Following our XML data and query models, the query scope can be identified as the set of XML data sub-trees (which we identify as Candidate Answer Trees, *CAT*s), in the data repository, having identical, or semantically similar enough, root nodes as that of the query (i.e., same/similar label, with the same data-type). Consider for instance query Q_1, searching for *movies* that have certain characteristics. When considering root

node identity, query Q_1's *CAT*s would be all data sub-trees having root node *movie*. When taking into account semantic similarity, Q_1's *CAT*s would also encompass subtree T_1 of root node *picture* from Fig. 5.

Definition 8. Candidate Answer Tree: Given an XML node similarity measure $Sim_{Semantic}$, and reference semantic network *SN* for evaluating the semantic similarity between XML concept and node labels, and a semantic similarity threshold α, the set of candidate answer trees Q_{CAT}, for a given query *Q*, in an XML data collection *C*, $Q_{CAT} = \{S / S \blacktriangleleft C \wedge ((R(Q) = R(S) \text{ if } \alpha = 1) \vee Sim_{Semantic}(R(Q), R(S), SN) \geq \alpha \text{ otherwise})\}$ ●

The semantic similarity threshold also serves as a structural/semantic similarity parameter, underlying the extent of structural/semantic similarity considered while identifying candidate answers. It allows the user to assign more importance to the structural or semantic characteristics of XML data in answering the query at hand:

- For $\alpha = 1$, only *CAT*s with root nodes identical to that of the query are the only ones considered. This corresponds to purely structural querying.
- For $0 < \alpha < 1$, *CAT*s with root nodes of semantic similarity higher than α are considered. As α decreases, the size of the answer set Q_{CAT} will increase, following the semantic similarities between query and *CAT* root nodes.
- For $\alpha = 0$, all data sub-trees in the XML data collection are considered as *CAT*s.

As for the semantic similarity measure $Sim_{Semantic}$, it is evaluated w.r.t. the nodes' constituents, i.e. their concepts and tag labels, where existing semantic similarity measures (e.g. Lin [47], Wu and Palmer [88]) can be exploited (cf. background in Section 2), taking into account the concerned reference semantic network. In our approach, our measure consists of a linear combination of Lin [47], and Wu and Palmer [88]), assigning equal weights to both measures. Other measures can be used according to the admin user's preferences.

6.2. Relevance Weight Function

We introduce a set of weighting functions to assign weight scores to XML nodes and edges, allowing to weight and rank the candidate answer trees. Considering an XML node n_i in the semantic XML tree, the weight of n_i is computed according to the below formula where we consider "Fan-in" to be the number of nodes connected with the target XML node:

$$W_{XMLNode}(n_i) = \frac{Fan-in(n_i)}{\underset{\forall v_j \in V_{index}}{Max}\ (Fan-in(n_j))} \in [0,1] \quad (1)$$

The rational is that an XML node is more important if it shares more links from other XML nodes. Given an XML edge e_i^j connecting XML nodes n_i and n_j in the XML tree, we define the weight of e_i^j as follows:

$$W_{XMLEdge}(e_i^j) = \frac{1}{Fan-out_{Label}(n_i)} \in]0,1] \quad (2)$$

The weight of an XML edge is inversely proportional to the number of links from a certain node to another, taking into account the semantic relation type of the link at hand (e.g., parent-child, element-attribute, element-value). The rationale here is that an XML edge designates a stronger connection between two XML nodes when it carries most of

the descriptive power from the source node to the destination node, such that the source node has few other out-going connections.

The scores of XML nodes/edges returned as query answers are computed using typical *Diskstra*-style shortest distance computations. Yet, instead of identifying the shortest (smallest) distance, we identify as answers XML sub-tree root nodes having the maximum similarity (similarity being the inverse function of distance) w.r.t. the starting nodes (mapping to keyword queries). In other words, given the sample CAT T in 0, with root node $n_d = root(T)$ and leaf nodes $n_{i...j}$, we define the relevance score of n_d w.r.t. $n_{i...j}$ as follows:

$$score(n_d, n_{i...j}) = \frac{\sum_{i...j} \frac{W_{XMLNode}(n_d) \times W_{XMLEdge}(e_p^d) \times \frac{1}{d(n_p, n_d)} + ... + W_{XMLNode}(n_x) \times W_{XMLEdge}(e_i^x) \times \frac{1}{d(n_i, n_d)}}{d(n_i, n_d)}}{|n_{i...j}|} \in [0,1] \quad (7)$$

where $d(n_i, n_d)$ is the distance in number of edges between two nodes, and $|n_{i...j}|$ is the number of leaf nodes in the CAT T rooted at n_d. In other words, in the following example, $d(n_p, n_d) = 1$, $d(n_j, n_d) = 2$, and $d(n_i, n_d) = 3$.

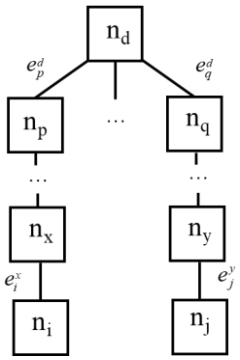


Fig. 9. Sample node linkage in an XML candidate answer tree

6.3. Semantic Query Processing

Having transformed the XML document collection and the keyword query into meaningful semantic representations, XML semantic search comes down to identifying and ranking the most relevant semantic XML sub-structures encompassing the semantic key-concepts in the query. Our extended framework includes three query processing algorithms: i) the core semantic search algorithm and two other variants designed to improve: ii) user involvement, and iii) query efficiency:

i. Core algorithm: titled *Semantic Search* and originally described in [84], it performs semantic-aware search using shortest path navigation in the *SemIndex* graph,
ii. User involvement: *Query-As-You-Type Search*, allows users to manually choose the meanings of query keywords before performing semantic search, aiming to involve the user in improving search result quality,

iii. Query Efficiency: *Parallel Semantic Search* is a parallel processing (multithreading) version of *SI_SS*, aiming to reduce query execution time.

6.4. Semantic Search

Our main querying method is based on a structural clustering technique to group together key-concept occurrences, in the XML data collection, which are structurally close. Our objective is to identify and rank the most prominent candidate answer sub-trees, in the XML data set, containing related occurrences of query key-concepts. Our semantic search algorithm is shown in Fig. 10 and is described below:

Step 1 - Identifying concept occurrences: The first step consists in pinpointing the XML nodes, in the data collection, containing occurrences of the query key-concepts.

Step 2 - Performing XML node clustering: Having identified the XML nodes encompassing key-concept occurrences, we perform structural clustering [55] to group together the XML nodes which are closest in the XML tree. The algorithm is applied on the weighted distances separating concept occurrences (cf. Section 6.2).

Step 3 – Constructing Answer Trees: We construct candidate answer trees based on the XML node clusters. An answer tree consists of the sub-tree rooted at the lowest common ancestor of all concept occurrences in the corresponding cluster.

Step 4 – Ranking Answer Trees: Having identified the candidate answer sub-trees, we rank them following their relevance to the query. Here, we utilize an integrated function combining various ranking criteria including i) weights of semantic concepts; ii) answer tree size (compactness), iii) common usage of senses (e.g., WordNet estimates the average usage frequency of word meanings in the English language, following the Brown corpus [29]), where the most commonly used senses are deemed more relevant in ranking results [49]. Other weighting functions can be used.

```
Algorithm SemanticSearch
Input:  T        // Semantic XML tree
        K        // Set of query selection terms
        D        // Sphere diameter designating query context size

Ouput: N_Out     // List of ranked trees from T designating query answers

Begin                                                                    1
N_Out = ϕ                                                                2
Step 0:  S = getSemanticQuerySenses(K)   // Global disambiguation        3
For each term s_i ∈ S                    // For each keyword sense
{                                                                        4
    Step 1: n_In = getNodeID(s_i, T)    // Identify concept occurrences
    Step 2: SP = PerformClustering(n_In, D, T)                           5
    Step 3: N_init = constructAnswerTree(SP, T)                          6
    Step 4: N_Out = rankAnswerTree(N_init, T)                            7
}                                                                        8
  Return N_Out                                                           9
                                                                        10
End
```

Fig. 10. Pseudo-code of *Semantic Search* algorithm

Note that the complexity of the semantic search algorithm comes down to the complexity of the structure clustering algorithm in Step 2. We utilize Lloyd's heuristic algorithm [48] to bound clustering complexity to $O(N \times C \times I)$ where N is the number of XML nodes to be clustered, C the number of produced clusters, and I the number of iterations to reach convergence.

6.5. Query As You Type Search

This algorihtm allows the user to choose the proper meaning for every query keyword, by allowing her to choose the intended sense from the set of all possible senses provided by WordNet. Once the senses have been chosen, the algorithm pinpoints in semantic network graph the indexing nodes corresponding to the chosen senses, and then runs typical shortest patth search starting from the chosen nodes. The pseudo-code of is basically the same as that of *Semantic Search*, except for adding a *step 0*: n_{In} = *manual*(*K*, *SN*), i.e., allowing the user to manually choose the proper meaning of every query term, among the list of possible meanings presented to the user through the system's GUI (cf. Fig. 11). Then, *Semantic Search* resumes by identifying and only processing the starting nodes corresponding to the term senses (synsets) chosen by the user. The algorithm's main steps can be described as follows:

i. Allow the user to choose the sense of each term in the query according to WordNet,
ii. Identify in the semantic XML tree the nodes corresponding to the chosen senses,
iii. Run the resulting query, starting from the identified index nodes, as a typical semantic keyword query search.

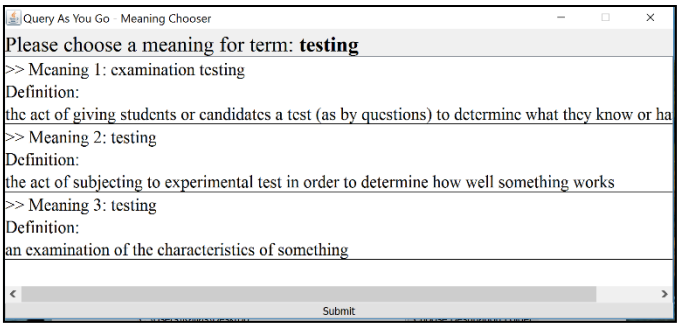


Fig. 11. *Query-As-You-Type* sub-interface

6.6. Parallel Semantic Search

We have also introduced a parallelized version of algorithm *Semantic Search* (cf. Fig. 12), which preserves (more or less) the same workflow of the original algorithm except

that it processes query terms and starting XML nodes using multiple threads running in parallel. The algorithm's main steps are described as follows:

i. Every query term is assigned a dedicated thread, and is thus processed independently from other threads (lines 1-2),
ii. After identifying the starting nodes for a query term (line 4), every starting node is then assigned its own dedicated thread (line 5), allowing to: compute the shortest paths from the starting node to data nodes in the XML tree (line 7), and then identify the reached data nodes designating potential query answers (i.e., CATs, line 8),
iii. Results are gradually merged (line 9) as they are produced by each thread, to rank and select (lines 10-12) query answers.

The implementation of the algorithm is configured to run as many threads as there are terms in the user query, where thread scheduling and parallel execution is left to the operating system.

```
Algorithm ParallelSemanticSearch

Input:  T        // Semantic XML tree
        K        // Set of query selection terms
        D        // Sphere diameter designating query context size

Ouput: N_Out     // List of ranked trees from T designating query answers

Begin                                                                    1

N_Out = ϕ                                                                2

Step 0:  S = getSemanticQuerySenses(K)   // Global disambiguation         3

Create Thread for each term s_i ∈ S       // For each keyword sense       4
{
    Step 1: n_In = getNodeID(s_i, T)   // Identify concept occurrences    5
    Create Thread for each n_i ∈ n_In                                     6
    {                                                                    7
        Step 2: SP = PerformClustering(n_i, D, T)                        8
        Step 3: N_init = constructAnswerTree(SP, T)                      9
        Step 4: N_Out = rankAnswerTree(N_init, T)                        10
    }                                                                    11
}                                                                        12
Return N_Out                                                             13

End
```

Fig. 12. Pseudo-code of *Parallel Semantic Search* algorithm

7. Experimental Evaluation

7.1. Experimental Scenario

We conducted a battery of experiments to test and evaluate our approach. We used a collection of 80 test documents gathered from several data sources having different properties[1]. Target XML nodes were first subject to manual disambiguation (12-to-13 nodes were randomly selected per document, yielding a total of 1000 target nodes, allowing human testers to annotate each node by choosing appropriate senses from WordNet) followed by automatic disambiguation. We formulated different with varying numbers of keywords, e.g., from 1 (single term query) to 5, where each query expands its predecessor by adding an additional selection term to the latter cf. sample queries in Table 1). We then compared user and system generated senses to compute *precision* (PR), *recall* (R), *f-value*, and *mean average precision* (MAP) scores.

Table 1. Sample test queries used in our experiments

Query *Q1*	
ID	Terms
Q1_1	"music"
Q1_2	"music", romance"
Q1_3	"music", romance", "dinner"
Q1_4	"music", romance", "dinner", "trip"
Q1_5	"music", romance", "dinner", "trip", "Paris"

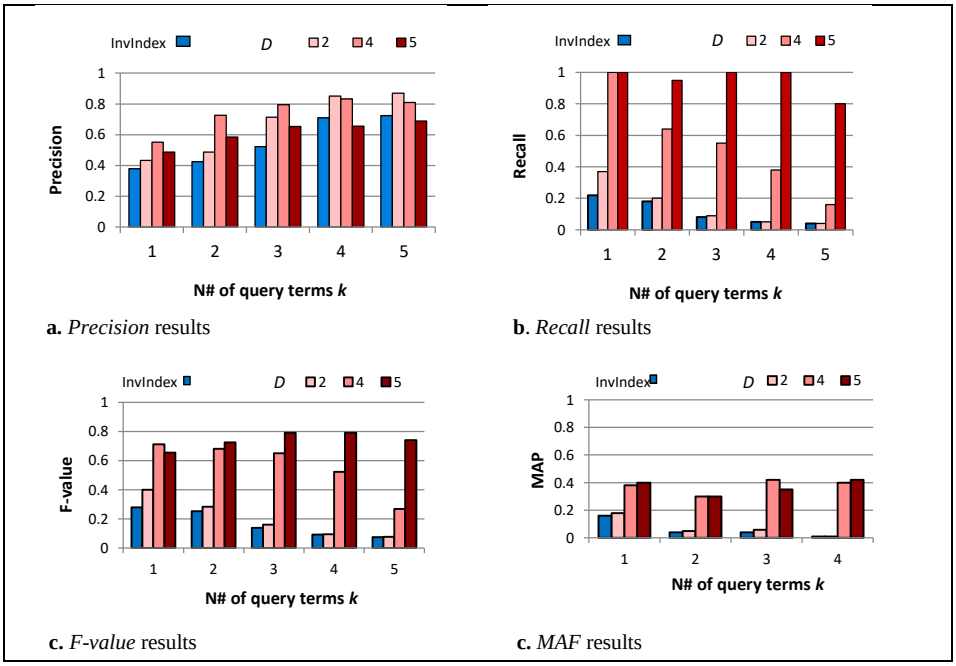
Query *Q2*	
ID	Terms
Q2_1	"play"
Q2_2	"play", "theater"
Q2_3	"play", "theater", "scene"
Q2_4	"play", "theater", "scene", "hero"
Q2_5	"play", "theater", "scene", "hero", "climax"

7.2. Query Result Quality

We first tested the effectiveness of our approach considering its different features and configurations: i) the properties of XML data (w.r.t. ambiguity and structure), and ii) context size (sphere neighborhood radius). Results in Fig. 13 show that precision levels increase with the number of query terms *k*. This is due to the human testers' expectations: given that queries are expanded versions of one another, result quality is evaluated based on the user's intent: which is expressed with the most expanded (i.e., most expressive) query (e.g., *Q1_5* and *Q2_5*). One can realize that using fewer query terms produces lower precision levels, which is due to the system returning more results which are (semantically related to the query terms but which are) not necessary related to the user's intent. As for recall, one can realize that levels steadily increase with concept depth *D*, where the number of correct (i.e., user expected) results returned by the system increases as more semantically related terms are covered in the querying process. F-value results increase with the increase of context depth *D*, and they slightly decrease with the increase of the number of query keywords *k*. This confirms the precision and recall results, where the determining factor affecting retrieval quality

[1] Shakespeare collection http://metalab.unc.edu/bosak/xml/eg/shaks200.zip, Amazon product files simply-amazon.com/content/XML.html,SIGMOD Record, http://www.acm.org/sigmod/xml, Niagara collection http://www.cs.wisc.edu/niagara/

remains context depth *D*. An increase in the number of keywords *k* tends to reduce system recall with higher values of *k* (queries becoming very selective, thus missing some relevant results). F-value levels are significantly higher than those obtained with the legacy inverted index, highlighting a clear improvement over syntactic retrieval quality. Also, mean average precision levels seem to concur with those of *f-value*, such that the ranking of relevant results compared with non-relevant ones in the queries' result lists seems to increase with the increase of *D* and fluctuate (based on the values of *D*) with the increase of *K*. In other words, increasing *D* not allowed retrieving more relevant results and improved the ranking of relevant results w.r.t. non-relevant ones in the query result list.



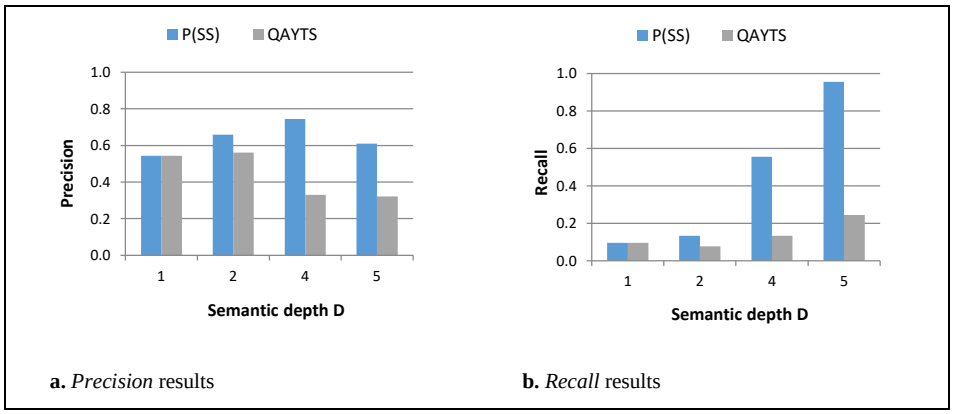
a. *Precision* results

b. *Recall* results

c. *F-value* results

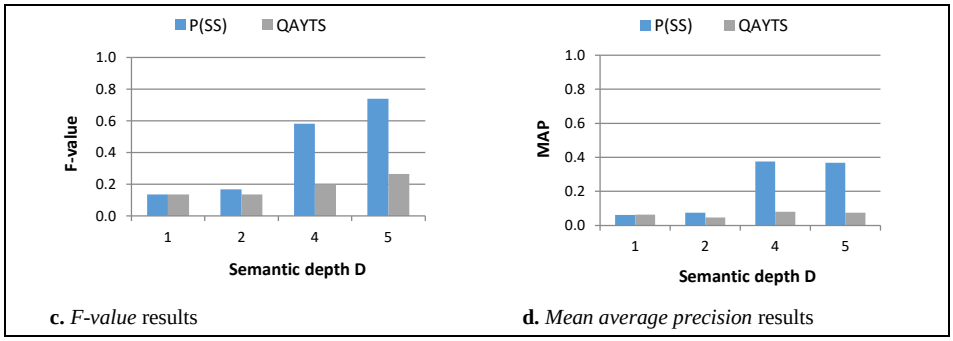
c. *MAF* results

Fig. 13. Comparing *Semantic Search* average precision (PR), recall (R), f-value, and mean average precision results with legacy inverted index syntactic search



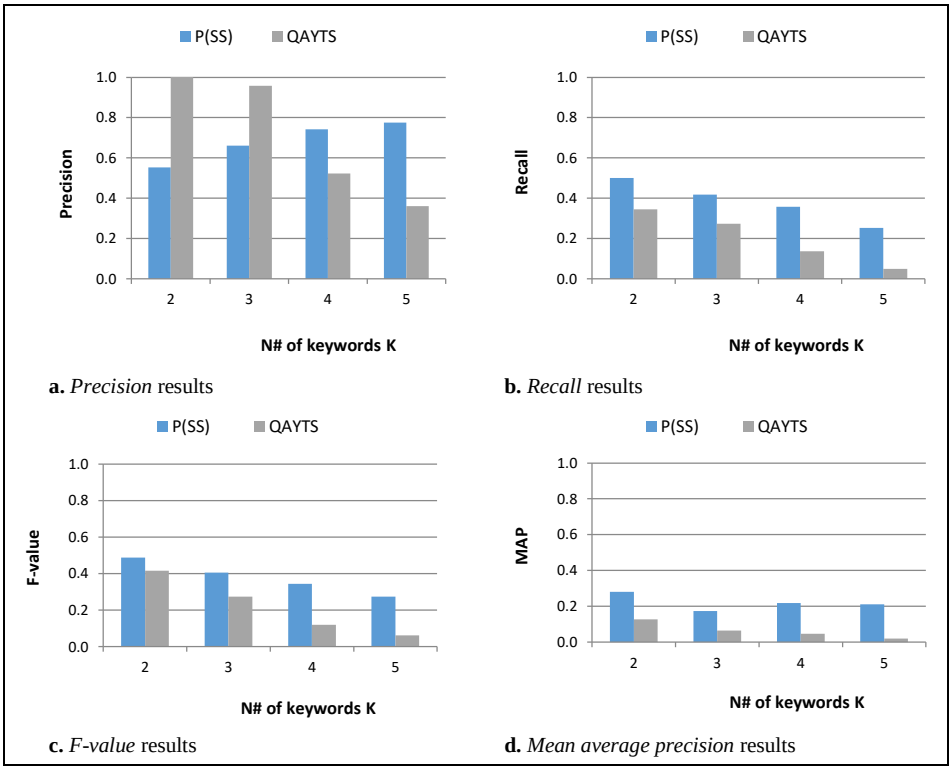
a. *Precision* results

b. *Recall* results



c. *F-value* results **d.** *Mean average precision* results

Fig. 14 Comparing *Semantic Search* (*SS*), *Query As You type Search* (*QAYTS*), and *Parallel semantic Search* (*PSS*) average precision (PR), recall (R), f-value, and mean average precision (MAP) results when varying semantic depth *D*



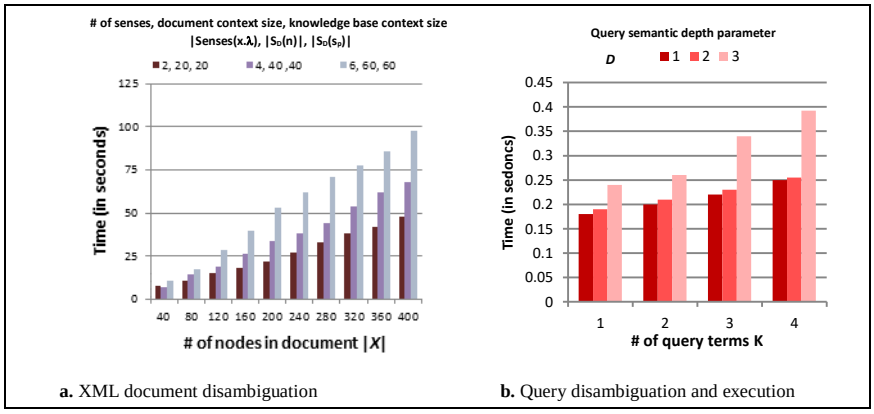
a. *Precision* results **b.** *Recall* results

c. *F-value* results **d.** *Mean average precision* results

Fig. 15. Comparing *Semantic Search* (*SS*), *Query As You type Search* (*QAYTS*), and *Parallel semantic Search* (*PSS*) average precision (PR), recall (R), f-value, and mean average precision (MAP) results when varying semantic depth *D*

7.3. Query Processing Time

We evaluated our solution's almost linear efficiency. Results in Fig. 16 highlight the polynomial (almost linear) complexities of both our (offline) XML document disambiguation and (online) global query disambiguation approaches, considering different parameter configurations for both processes. Results in Fig. 16.b show total query execution time including online disambiguation, by varying both the number of keywords and query semantic depth *D* (i.e., semantic context size).

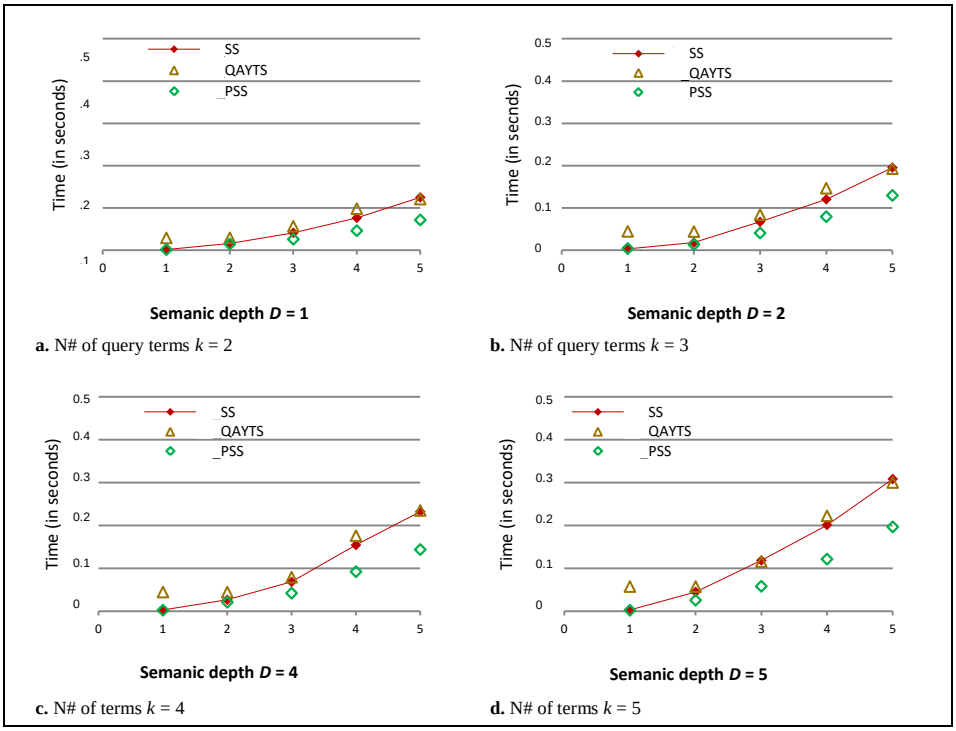


a. XML document disambiguation **b.** Query disambiguation and execution

Fig. 16. XML document disambiguation time (a) and query disambiguation and execution time (b)

We ran the same queries through the three querying algorithms: *SS*, *QAYTS*, and *PSS*. Fig. 17 provides average processing time results for all queries, plotted by varying the number of query terms *K* and link distance threshold *D*. First, results of all three algorithms show that query execution time increases almost linearly with the number of query terms *K* (when fixing link distance *D*), and increases linearly with *D* (when fixing *K*), highlighting the algorithms quadratic complexity levels. Second, results show that all three algorithms have very close query time levels when both *K* and *D* are small (=1 and 2), such that time difference increases as both *K* and *D* increase. This is due to the fact increasing either *K* or *D* means increasing the number of nodes to be navigated in the semantic XML tree: increasing *k* means navigating the XML tree starting from a larger number of initial nodes, and increasing *D* means reaching deeper into the tree structure to identify more semantically relevant results. Third, algorithms *SS* and *QAYTS* produced almost identical time levels (disregarding the manual effort required in *QAYTS*[2]), whereas the parallel processing *PSS* algorithm is clearly the most efficient of its counterparts, requiring almost 33.34% less time than *SS* and *QAYTS* with maximum *k*=5 and *D*=5.

[2] *QAYTS*'s time shown in Fig. 16 does not encompass the time it took the testers to manually choose the meanings of query terms (which we did not consider to be part of the algorithm itself), but only considers actual algorithm (CPU and SQL) execution time.



a. N# of query terms $k = 2$ **b.** N# of query terms $k = 3$

c. N# of terms $k = 4$ **d.** N# of terms $k = 5$

Fig. 17. Comparing average query execution time of *Semantic Search* (*SS*), *Query As You type Search* (*QAYTS*), and *Parallel semantic Search* (*PSS*), while varying semantic depth *D* and the number of query terms *K*

8. Conclusion

In this paper, we describe *XSemSearch*, a solution for XML keyword search allowing to transform both XML documents and keyword queries into semantic representations, using semantic concepts in a reference knowledge base. We describe two approaches for i) offline context-based XML document disambiguation and ii) online global keyword query disambiguation, both designed to run in almost linear time. Our solution is: i) fully automated, compared with existing interactive solutions which require user input to manually identify the intended query senses e.g., [35, 61], and ii) tractable (of almost linear time) and thus reasonably applicable on the Web, compared with polynomial or exponential solutions, e.g., [23, 58]. Our solution also provides iii) a dedicated index structure to handle semantic XML trees, iv) a dedicated query formalism to allow structure-and-content queries with only partial knowledge of the data collection structure and semantics, and iv) three alternative query processing algorithms to evaluate query processing time and quality.

We are currently investigating the integration of semantic-aware indexing capabilities [79-81] and different clustering algorithms to form XML answer trees [33, 77]. This would provide more opportunities toward both speed-ups and semantic-based

filtering. We are also investigating the use of alternative knowledge sources such as Google [1], Wikipedia [86], and FOAF [4] to acquire a wider word sense coverage, and explore our approach in practical applications, namely semantic-aware document and schema matching [82, 83], RSS news feed merging [68, 69], affective blog analysis [26, 27], social event detection [3, 5], and semantic relations' identification from social media data [2]. On the long run, we aim to investigate word embeddings and learning statistical distributions in a corpus [34, 92], to infer semantics without the need for predefined knowledge bases.

References

1. Abdulhayoglu M. and Thijs B., *Use of ResearchGate and Google CSE for author name disambiguation.* Scientometrics 2017. 111(3): 1965-1985.
2. Abebe M., et al., *Generic Metadata Representation Framework for Social-based Event Detection, Description, and Linkage.* Knowledge Based Systems 2020. 188.
3. Abebe M. A., et al., *Overview of Event-Based Collective Knowledge Management in Multimedia Digital Ecosystems.* International Conference of Signal Image Technology and Internet-based Systems (SITIS'17), 2017. pp. 40-49.
4. Amith M., Fujimoto K., Mauldin R.,, and Tao C., *Friend of a Friend with Benefits ontology (FOAF+): extending a social network ontology for public health.* BMC Medical Informatics & Decision Making - Supplement, 2020. 20-S(10): 269.
5. Ashagrie M., et al., *A General Multimedia Representation Space Model toward Event-based Collective Knowledge Management.* Submitted to 19th IEEE International Conference on Computational Science and Engineering (CSE 2016), 2016. Paris, France.
6. Azzini A., et al., *A Neuro-Evolutionary Corpus-based Method for Word Sense Disambiguation.* IEEE Intelligent Systems, 2012. 27(6): 26-35.
7. Baeza-Yates R. and Ribeiro-Neto B., *Modern Information Retrieval: The Concepts and Technology behind Search.* ACM Press Books, Addison-Wesley Professional, 2nd Ed., 2011. p. 944.
8. Banerjee S. and Pedersen T., *Extended Gloss Overlaps as a Measure of Semantic Relatedness.* International Joint Conference on Artificial Intelligence (IJCAI'03), 2003. p. 805-810.
9. Baziz M.; Boughanem M. and Traboulsi S., *A concept-based approach for indexing documents in IR.* INFORSID 2005, 2005. pp. 489-504, Grenoble, France.
10. Bertino E.; Guerrini G.; and Mesiti, M., *A Matching Algorithm for Measuring the Structural Similarity between an XML Documents and a DTD and its Applications.* Elsevier Information Systems, 2004. (29):23-46.
11. Bobed C. and Mena E., *QueryGen: Semantic Interpretation of Keyword Queries over Heterogeneous Information Systems.* Information Sciences, 2016. 329: 412-433.
12. Bonab H., et al., *Incorporating Hierarchical Domain Information to Disambiguate Very Short Queries.* International Conference on the Theory of Information Retrieval (ICTIR'19), 2019. pp. 51-54.
13. Budanitsky A. and Hirst G., *Evaluating WordNet-based Measures of Lexical Semantic Relatedness.* Computational Linguistics, 2006. 32(1): 13-47.
14. Burton-Jones A.; Storey V.C.; Sugumaran V. and Purao S., *A Heuristic-Based Methodology for Semantic Augmentation of User Queries on the Web.* In Proceedings ot the International Conference on Conceptual Modeling (ER'03), 2003. pp. 476–489.
15. Calì A., Martinenghi D., and Torlone R., *Keyword Queries over the Deep Web.* International Conference on Conceptual Modeling (ER'16), 2016. pp. 260-268.

16. Chaplot D. and Salakhutdinov R., *Knowledge-based Word Sense Disambiguation using Topic Models.* AAAI Conference on Artificial Intelligence (AAAI'18), 2018. pp. 5062-5069.
17. Charbel N., et al., *Resolving XML Semantic Ambiguity.* International Conference on Extending Database Technology (EDBT'15), 2015. Brussels, Belgium, pp 277-288.
18. Chawathe S.; Rajaraman A.; Garcia-Molina H.; and Widom J., *Change Detection in Hierarchically Structured Information.* Proceedings of the ACM International Conference on Management of Data (SIGMOD), 1996. pp. 26-37. Montreal.
19. Che D., Ling T., and Hou W., *Holistic Boolean-Twig Pattern Matching for Efficient XML Query Processing.* IEEE Transactions on Knowledge and Data Engineering, 2012. 24(11): 2008-2024.
20. Cobéna G.; Abiteboul S.; and Marian A., *Detecting Changes in XML Documents.* Proceedings of the IEEE International Conference on Data Engineering (ICDE), 2002. pp. 41-52.
21. Cormen T. H.; Leiserson C. E.; Rivest R. L. and Stein C., *Introduction to Algorithms (Second ed.) - Section 24.3: Dijkstra's Algorithm.* MIT Press and McGraw-Hill, 2001. pp. 595–601.
22. Dalamagas T.; Cheng T.; Winkel K.; and Sellis T., *A Methodology for Clustering XML Documents by Structure.* Information Systems, 2006. 31(3):187-228.
23. de Campos L., et al., *XML Search Personalization Strategies using Query Expansion, Reranking and a Search Engine Modification.* ACM Symposium on Applied Computing (SAC'13) 2013. pp. 872-877.
24. Demidova E., ZhouIrina X., and Nejdl O., *Evaluating Evidences for Keyword Query Disambiguation in Entity Centric Database Search.* International Conference on Database and Expert Systems Applications (DEXA'10), 2010. pp. 240-247.
25. Di Iorio A., et al., *A First Approach to the Automatic Recognition of Structural Patterns in XML Documents* ACM Symposium on Document Engineering, 2012. pp. 85-94.
26. Fares M., et al., *Difficulties and Improvements to Graph-based Lexical Sentiment Analysis using LISA* IEEE International Conference on Cognitive Computing (ICCC'19), 2019.
27. Fares M., et al., *Unsupervised Word-level Affect Analysis and Propagation in a Lexical Knowledge Graph.* Elsevier Knowledge-Based Systems, 2019. 165: 432-459.
28. Fragos K., *Modeling WordNet Glosses to Perform Word Sense Disambiguation.* International Journal of Artificial Intelligence Tools, 2013. 22(2).
29. Francis W. N. and Kucera H., *Frequency Analysis of English Usage.* Houghton Mifflin, Boston, 1982.
30. Gao J., et al., *Learning Lexicon Models from Search Logs for Query Expansion.* Conference on Empirical Methods in Natural Language Processing (EMNLP'12), 2012. pp. 666-676.
31. Graupmann J.; Schenkel R. and Weikum G., *The SphereSearch Engine for Unified Ranked Retrieval of Heterogeneous XML and Web Documents.* Proceedings of the International Conference on Very Large Databases (VLDB), 2005. pp. 529-540.
32. Guha S.; Jagadish H.V.; Koudas N.; Srivastava D.; and Yu T., *Approximate XML Joins.* Proceedings of ACM International Conference on Managemenet of Data (SIGMOD), 2002. pp. 287-298.
33. Haraty R., Dimishkieh M., and Masud M., *An Enhanced k-Means Clustering Algorithm for Pattern Discovery in Healthcare Data.* Intelligent Journal on Distributed Sensor Networks, 2015. 11: 615740:1-615740:11.
34. Haraty R. and Nasrallah R., *Indexing Arabic Texts using Association Rule Data Mining.* Library Hi Tech, 2019. 37(1): 101-117.
35. Harman D., *Towards Interactive Query Expansion.* SIGIR Forum 2017. 51(2): 79-89.
36. Helmer S., *Measuring the Structural Similarity of Semistructured Documents Using Entropy* Proceedings of the International Conference on Very Large Databases (VLDB), 2007. pp. 1022-1032.
37. Hoffart J., et al., *YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia.* Artif. Intell., 2013. 194: 28-61.

38. Holub M., et al., *Tailored Feature Extraction for Lexical Disambiguation of English Verbs Based on Corpus Pattern Analysis.* International Conference on Computational Linguistics (COLING'12), 2012. pp. 1195-1210.
39. Iranzo P. and Sáenz-Pérez F., *Implementing WordNet Measures of Lexical Semantic Similarity in a Fuzzy Logic Programming System.* Theory and Practice of Logic Programming, 2021. 21(2): 264-282.
40. Kamvar M. and Baluja S., *A Large Scale Study of Wireless Search Behavior: Google Mobile Search.* In Proceedings of the SIGCHI Conference on Computer Human Interaction, 2006. pp. 701–709, Montreal, Canada.
41. Kumar R., Guggilla B., and Pamula R., *Book search using social information, user profiles and query expansion with Pseudo Relevance Feedback.* Applied Intelligence, 2019. 49(6): 2178-2200.
42. Kwon S., Oh D., and Ko Y., *Word Sense Disambiguation based on Context Selection using Knowledge-based Word Similarity.* Information Processing and Management, 2021. 58(4): 102551.
43. Leacock C. and Chodorow M., *Combining Local Context and WordNet Similarity for Word Sense Identification.* FellBaum C. editor, WordNet: An Electronic Lexical Database, Chapter 11, The MIT Press, Cambridge, 1998. pp. 265-283.
44. Li Y.; Yang H. and Jagadish H.V., *NaLIX: an interactive natural language interface for querying XML.* Proceedings of the International ACM Conference on Management of Data (SIGMOD), 2005. pp. 900-902.
45. Li Y.; Yang H. and Jagadish H.V., *Term Disambiguation in Natural Language Query for XML.* In Proceedings of the International Conference on Flexible Query Answering Systems (FQAS), 2006. LNAI 4027, pp. 133–146.
46. Liang W.; and Yokota H., *LAX: An Efficient Approximate XML Join Based on Clustered Leaf Nodes for XML Data Integration.* Proceedings of the British National Conference on Databases (BNCOD), 2005. pp. 82-97.
47. Lin D., *An Information-Theoretic Definition of Similarity.* Proceedings of the International Conference on Machine Learning (ICML), 1998. pp. 296-304. Morgan Kaufmann Pub. Inc.
48. Lloyd S., *Least Squares quantization in PCM.* IEEE Transactions on Information Theory, 1982. 28(2):129-137.
49. Mandreoli F. and Martoglia R., *Knowledge-based sense disambiguation (almost) for all structures.* Information Systems, 2011. 36(2): 406-430.
50. Miller G., *WordNet: An On-Line Lexical Database.* International Journal of Lexicography, 1990. 3(4).
51. Miller G.A. and Fellbaum C., *WordNet Then and Now.* Language Resources and Evaluation, 2007. 41(2): 209-214.
52. Mohammad S., Hirst G., and Resnik P., *Tor, TorMd: Distributional Profiles of Concepts for Unsupervised Word Sense Disambiguation.* SemEval@ACL 2007, 2007. pp. 326-333.
53. Navigli R., *Word Sense Disambiguation: a Survey.* ACM Computing Surveys, 2009. 41(2):1–69.
54. Navigli R. and Velardi P., *Structural Semantic Interconnections: A knowledge-based Approach to Word Sense Disambiguation* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005. 27(7):1075–1086.
55. Navigli R. and Crisafulli G., *Inducing Word Senses to Improve Web Search Result Clustering.* In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, 2010. pp. 116–126, MIT, USA.
56. Navigli R. and Velardi P., *An Analysis of Ontology-based Query Expansion Strategies.* In proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'03), 2003. pp. 42-49.
57. Nierman A. and Jagadish H. V., *Evaluating structural similarity in XML documents.* Proceedings of the ACM SIGMOD International Workshop on the Web and Databases (WebDB), 2002. pp. 61-66.

58. Qtaish A. and Alshammari M., *A Narrative Review of Storing and Querying XML Documents using Relational Database.* Journal of Information & Knowledge Management, 2019. 18(4): 1950048:1-1950048:28.
59. Rafiei D.; Moise D.; and Sun D., *Finding Syntactic Similarities between XML Documents.* Proceedings of the International Conference on Database and Expert Systems Applications (DEXA), 2006. pp. 512-516.
60. Resnik P., *Disambiguating Noun Groupings with Respect to WordNet Senses.* In Proceedings of the 3rd Workshop on Large Corpora, 1995. pp. 54-68.
61. Russell-Rose T., Gooch P., and Kruschwitz U., *Interactive Query Expansion for Professional Search Applications.* CoRR abs/2106.13528, 2021.
62. Salameh K., Tekli J., and Chbeir R., *SVG-to-RDF Image Semantization.* 7th International SISAP Conference, 2014. pp. 214-228.
63. Sanz I.; Mesiti M.; Guerrini G.; Berlanga La R.; and Berlanga Lavori R., *Approximate Subtree Identification in Heterogeneous XML Documents Collections.* XML Symposium, 2005. pp. 192-206.
64. Schlieder T., *Similarity Search in XML Data Using Cost-based Query Transformations.* Proceedings of the ACM SIGMOD International Workshop on the Web and Databases (WebDB), 2001. pp. 19-24.
65. Schlieder T. and Meuss H., *Querying and Ranking XML Documents.* Journal of the American Society for Information Science, Special Topic XML/IR, 2002. 53(6):489-503.
66. Singh S., Murthy H., and Gonsalves T., *Dynamic Query Expansion based on User's Real Time Implicit Feedback.* Conference on Knowledge Discovery and Information Retrieval (KDIR'10) 2010. pp. 112-121.
67. Soudani N., Bounhas I., and Ben Babis S., *Ambiguity Aware Arabic Document Indexing and Query Expansion: A Morphological Knowledge Learning-Based Approach.* The Florida AI Research Society Conference (FLAIRS'18 Conference), 2018. pp. 230-235.
68. Taddesse F.G., et al., *Semantic-based Merging of RSS Items.* World Wide Web Journal: Internet and Web Information Systems Journal Special Issue: Human-Centered Web Science., 2010. 13(1-2): 169-207, Springer Netherlands.
69. Taddesse F.G., et al., *Relating RSS News/Items.* Proceedings of the 9th International Conference on Web Engineering (ICWE'09), LNCS, 2009. pp. 44-452, San Sebastian, Spain.
70. Tagarelli A. and Greco S., *Semantic Clustering of XML Documents.* ACM Transactions on Information Systems, 2010. 28(1):3.
71. Tagarelli A.; Longo M. and Greco S., *Word Sense Disambiguation for XML Structure Feature Generation.* European Semantic Web Conference, 2009. LNCS 5554, pp. 143–157.
72. Taha K. and Elmasri R., *CXLEngine: A Comprehensive XML Loosely Structured Search Engine.* Proceedings of the EDBT workshop on Database Technologies for Handling XML Information on the Web (DataX'08), 2008. pp. 37-42, Nantes, France.
73. Taha K. and Elmasri R., *XCDSearch: An XML Context-Driven Search Engine.* IEEE Transactions on Knowledge and Data Engineering, 2010. 22(12):1781-1796.
74. Tannebaum W. and Rauber A., *Using Query Logs of USPTO Patent Examiners for Automatic Query Expansion in Patent Searching.* Information Retrieval, 2014. 17(5-6): 452-470.
75. Tekli J., *An Overview on XML Semantic Disambiguation from Unstructured Text to Semi-Structured Data: Background, Applications, and Ongoing Challenges.* IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), 2016. 28(6): 1383-1407.
76. Tekli J., et al., *Semantic to intelligent web era: building blocks, applications, and current trends.* . International Conference on Managment of Emergent Digital EcoSystems (MEDES), 2013. pp. 159-168.
77. Tekli J., et al., *(k, l)-Clustering for Transactional Data Streams Anonymization.* Information Security Practice and Experience, 2018. pp. 544-556.
78. Tekli J., Charbel N., and Chbeir R., *Building Semantic Trees from XML Documents.* Elsevier Journal of Web Semantics (JWS), 2016. 37–38:1–24.

79. Tekli J., et al., *SemIndex: Semantic-Aware Inverted Index.* Symposium on Advances in Databases and Information Systems (ADBIS), 2015. pp. 290-307.
80. Tekli J., et al., *SemIndex+: A Semantic Indexing Scheme for Structured, Unstructured, and Partly Structured Data.* Elsevier Knowledge-Based Systems, 2019. 164: 378-403.
81. Tekli J., et al., *Full-fledged Semantic Indexing and Querying Model Designed for Seamless Integration in Legacy RDBMS.* Data and Knowledge Engineering, 2018. 117: 133-173.
82. Tekli J., Chbeir R., and Yétongnon K., *A Fine-grained XML Structural Comparison Approach.* 26th International Conference on Conceptual Modeling (ER), 2007. LNCS 4801, pp. 582-598.
83. Tekli J., Chbeir R., and Yétongnon K., *Structural Similarity Evaluation between XML Documents and DTDs.* Proceedings of the 8th International Conference on Web Information Systems Engineering (WISE), 2007. pp. 196-211.
84. Tekli J., Tekli G., and Chbeir R., *Almost Linear Semantic XML Keyword Search.* Inter. ACM Conf. on Management of Emergent Digital EcoSystems (MEDES'21), 2021. pp. 129-138.
85. Theobald M.; Schenkel R. and Weikum G., *Exploiting Structure, Annotation, and Ontological Knowledge for Automatic Classification of XML Data.* In Proceedings of the ACM SIGMOD International Workshop on Databases (WebDB), 2003. pp. 1-6, San Diego, California.
86. Tu H., et al., *Word Sense Disambiguation Using Wikipedia Link Graph.* IEEE BigData 2019, 2019. pp. 6235-6236.
87. World Wide Web Consortium. *The Document Object Model*. http://www.w3.org/DOM, [Accessed Feb. 2022].
88. Wu Z. and Palmer M., *Verb Semantics and Lexical Selection.* Proceedings of the 32nd Annual Meeting of the Associations of Computational Linguistics, 1994. pp. 133-138.
89. Yang D., et al., *Query Intent Disambiguation of Keyword-Based Semantic Entity Search in Dataspaces.* Journal of Computer Science and Technology, 2013. 28:382–393.
90. Yaworsky D., *Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora.* Proceedings of the International Conference on Computational Linguistics (Coling), 1992. Vol 2, pp. 454-460. Nantes.
91. Yi J., Maghoul F., and Pedersen J., *Deciphering Mobile Search Patterns: a Study of Yahoo! Mobile Search Queries.* The Web Conference (WWW'08), 2008. pp. 257-266.
92. Zhang H. et al., *Learning from collective intelligence: Feature learning using social images and tags.* ACM transactions on multimedia computing, communications, and applications (TOMM), 2017. 13(1):1.
93. Zhang Z.; Li R.; Cao S.; and Zhu Y., *Similarity Metric in XML Documents.* Knowledge Management and Experience Management Workshop, 2003.

Joe Tekli is an Associate Professor in Computer Engineering in the Lebanese American University (LAU). He obtained his Ph.D. from the University of Bourgogne, LE2I-CNRS (France_2009). He completed various post-docs/research missions: University of Michigan (USA_2018), University of Pau (France 2017), University of Sao Paulo (Brazil_2011), University of Shizuoka (Japan_2010), University of Milan (Italy 2009). He was awarded various fellowships: Fulbright (USA 2018), FAPESP (Brazil 2011), JSPS (Japan 2010), Cariplo Foundation (Italy 2009), French Ministry of Education (France 2006-09), and AUF (France 2005). He has coordinated/participated in various projects: FAPESP (Brazil 2016-20), LAU-NCSR-L (Lebanon 2018-20), NCSR-L (Lebanon 2017-18), STICAmSud (France 2013-14), and CEDRE (France 2012-13). His research covers semi-structured, semantic, and multimedia data processing, and has more than 50 peer-reviewed publications. He is Vice Chair of ACM SIGAPP French Chapter (2018-) and founding member of UN-ESCWA Knowledge Hub

Gibert Tekli is an Associate Professor in the Mechatronics Engineering Technology Dept., the associate dean of the Issam Fares Faculty of Technology and an R&D engineering specialist in full stack agile cloud-based development, soft robotics and artificial intelligence. He holds a PhD in Computer Engineering from Telecom Saint Etienne, University of Lyon, France. He thrives on challenges rising from merging both worlds, Industrial R&D and Academia. He has successfully secured funds, lead, developed and consulted on international R&D projects (such as H2020 EU projects) while ensuring proper technology transfer from universities to the industry and vice versa.

Richard Chbeir received his PhD in Computer Science from the University of INSA-de-Lyon, France, in 2001. The author became a member of IEEE since 1999. He is currently a Full Professor in the Computer Science Department of the University of Pau and Pays de l'Adour (UPPA), Anglet, France. His is also Director of the UPPA Computer Science research laboratory (LIUPPA). His research interests are in the areas of distributed multimedia database management, XML similarity and rewriting, spatio-temporal applications, indexing methods, multimedia access control models, security and watermarking. He has published (more than 180 peer-reviewed publications) in international journals, books, and conferences, and has served on the program committees of several international conferences. He has been organizing many international conferences and workshops (ICDIM, CSTST, SITIS, MEDES, etc.). He is currently the Chair of the French Chapter ACM SIGAPP and the vice-chair of ACM SIGAPP.

Received: February 28, 2022; Accepted: August 22, 2022.

Computer Science and Information Systems 20(1):459–489 https://doi.org/10.2298/CSIS220301064B

Data-centric UML Profile for Agroecology Applications: Agricultural Autonomous Robots Monitoring Case Study

Sandro Bimonte[1], Hassan Badir[6], Pietro Battistoni[2], Houssam Bazza[6], Amina Belhassena[1], Christophe Cariou[1], Gerard Chalhoub[3], Juan Carlos Corrales[4], Adrian Couvent[1], Jean Laneurit[1], Rim Moussa[5], Julian Eduardo Plazas[4], Monica Sebillo[2], and Nicolas Tricot[1]

[1] Université Clermont Auvergne, TSCF, INRAE, France
{name.surname@inrae.fr
[2] University Salerno, Italy
{pbattistoni,msebillo}@unisa.it
[3] Université Clermont Auvergne, LIMOS-CNRS, France
gerard.chalhoub@uca.fr
[4] Universidad del Cauca, Colombia
{jcorral, jeplazas}@unicauca.edu.co
[5] University of Carthage, Tunisia
rim.moussa@enicarthage.rnu.tn
[6] IDS Team, Abdelmalek Essaadi University, Morocco
{houssam.bazza@etu.uae.ac.ma, badir.hassan@uae.ac.ma}

Abstract. The conceptual design of information systems is mandatory in several application domains. The advent of the Internet of Things (IoT) technologies pushes conceptual design tools and methodologies to consider the complexity of IoT data, architectures, and communication networks. In agroecology applications, the usage of IoT is quite promising, but it raises several methodological and technical issues. These issues are related to the complexity and heterogeneity of data (social, economic, environmental, and agricultural) needed by agroecology practices. Motivated by the lack of a conceptual model for IoT data, in this work, we present a UML profile taking into account different kinds of data (e.g., sensors, stream, or transactional) and non-functional Requirements. We show how the UML profile integrates with classical UML diagrams to support the design of complex systems. Moreover, We prove the feasibility of our conceptual framework through a theoretical quality assessment and its implementation in the agroecology case study concerning the monitoring of autonomous agricultural robots.

Keywords: Data Analytics, Internet of Things, Conceptual Modeling, UML Profile.

1. Introduction

In recent years, the Internet of Things (IoT) [29] has received much attention in multiple application domains, such as smart buildings and living, transport and mobility, healthcare, environment, energy, manufacturing, and *agriculture* and *agroecology* [4]. IoT represents a set of physical devices connected to the Internet that can generate, compute, store, and send data in real-time through different media (e.g., ZigBee, Wi-Fi, LoRaWAN) [1]. Moreover, the volume, heterogeneity and speed at which IoT can generate data classify them as a source of Big Data [29].

Technologies used for the implementation of IoT architectures have reached maturity. However, to the best of our knowledge, data-modeling methods for such architectures have not been well researched so far [29]. The **challenge in modeling IoT data** is caused by the fact that IoT data are: (1) *distributed and communicated over complex network architectures* (such as edge-fog-cloud) and (2) generated by a complex system. Such a system is typically composed of relational databases, NoSQL servers, and data stream management systems (DSMSs) besides the IoT. These components are implemented with various technologies supporting different programming languages and run on heterogeneous hardware (e.g., IoT devices, personal computers, and cloud servers). We refer to data generated in such a system as *polyglot data*.

IoT data are not only persistent but also transient. IoT data arrive into a system in the form of streams and are processed in real-time by streaming analytics applications [46] and Complex Event Processing (CEP) applications [53]. These applications monitor and discover trends and detect anomalies by means of continuous queries. Next, these data are typically stored into repositories such as *data warehouses* [52], *data lakes* [44] or *lakehouses* [54] to analyze them offline through OLAP (On-Line Analytical Processing) applications. Indeed, IoT real-time data are often combined with offline data to provide more advanced analysis [33].

Moreover, IoT applications are characterized by a geographically distributed deployment of devices and a network communication continuum over different layers (from the edge to the cloud) [38]. Therefore, Quality of Service features (QoS) plays a significant role in IoT data architectures, especially in the agricultural field of application, which is usually characterized by low quality communication networks. QoS can reflect some functional requirements, such as latency, which leads to a particular placement of data and computation over the different layers. For example, in the context of hard real-time applications, data and computation can be deployed at the edge level to improve performance.

Conceptual design of Information Systems (IS) has several advantages [43]. *First*, it allows to keep away implementation details and allows decision-makers and IS to exclusively focus application content and functionalities. *Second*, it provides a formal and non-ambiguous support used by decision-makers to validate their requirements. *Third*, it streamlines the implementation phase providing some technical guidelines (and sometimes also an automatic implementation). Although the conceptual design of data for IoT applications is crucial for their successful implementations, this topic has not been intensively researched yet [41].

Indeed, existing conceptual models do not allow to represent different data types issued from IoT in the same design framework. In addition, they do not support any QoS at the conceptual level. Therefore, the software engineering process for IoT-based applications is based on different conceptual models for each data type (stream, data warehouse, etc.), and QoS are taken into account at the implementation time. This implies that the merging phase of these different implementations is difficult or sometimes unfeasible.

This lack of design methodologies for IoT applications is evident in several domains, such as urban vehicles management (i.e., smart scheduling of traffic), health (i.e., real-time monitoring of physical and biological behavior of patients), logistic (i.e., smart affectation of human resources), tourism (i.e., enhance and optimize paths and stay) and also agroecology, which is the focus of this paper.

Nowadays, every organization, enterprise, and country must recognize the importance of new agricultural paradigms considering environmental, animal, and human health for sustainable development. In this line, agroecology is the main pattern to achieve this mandatory goal of humanity. The Food and Agriculture Organization of the United Nations defines agroecology as "*An integrated approach that simultaneously applies ecological and social concepts and principles to the design and management of food and agricultural systems. It seeks to optimize the interactions between plants, animals, humans, and the environment while taking into consideration the social aspects that need to be addressed for a sustainable and fair food system*" [7]. Agroecology evolves precision agriculture concepts, which mainly analyze crop-related data at detailed granularities, to consider more complex and global agronomic, social, economic, and environmental contexts [17]. Therefore, agroecological systems need a comprehensive approach, where versatile data types can be integrated and analyzed in multiple spatio-temporal dimensions.

Motivated by this lack of comprehensive solutions and by the formal support for data conceptual models provided by UML profiles, in [7] we proposed a UML profile for the data-centric design of agroecology IoT applications that considers some QoS network features, relevant at the conceptual level for end-users. Our UML profile, which is based on class diagrams, allows an *easy understanding and a formal representation* of these different types of data within a unique framework, which allows at the same time a coherent and global representation of all relationships between data. It is a crucial factor for the data-centric design of IoT applications. The different types of data have interactions among them in terms of data associations and network communications.

In this paper, we extend [7] in the following ways:

1. We present a design and implementation methodology centered on our UML profile. The main idea is to use the UML profile to define all data and non-functional requirements at a conceptual level in the same UML class diagram. Then, use these classes to define dynamic aspects of the application by means of other standard UML diagrams. In this way, our UML profile can be transparently adopted in all existing UML based software engineering development methodologies. To validate this feature, we show how classical UML-based software engineering design methodologies (such as [31]) can be used using our Class diagram UML profile. In particular, we leverage our case study with autonomous agricultural robots as an example. Besides, we detail how the UML Use Case and Activity diagrams can be used to derive and further detail the implementation of our Class diagram.
2. We provide a theoretical assessment of our UML profile quality, evaluating five quantitative metrics: *Reusability*, *Understandability*, *Well-structuredness*, *Functionality* and *Extendibility* according to [6,28].
3. We detail the technical implementation of each kind of data supported by our UML profile.

The paper is organized in the following way: Section 2 presents a real-world application in agroecology using IoT and autonomous robots. Section 3 presents our UML profile [7]. Section 4.1 presents the theoretical quality assessment. Section 5 shows the details of the implementation of the different supported data types (and underlying systems) using

[7] https://www.fao.org/3/i9037en/i9037en.pdf

our case study. Related work is shown in Section 6. Finally, Section 7 concludes the paper and proposes future work.

2. Motivation: Agricultural robots monitoring and scheduling case study

This section extends [7], presenting the motivation of our work by means of a case study, which will be also used in this paper to describe our proposal. In particular, the case study outlines the set of functional and non-functional requirements that must be supported.

The case study is based on the French I-SITE CAP2025 *Superob* project. The overall goal of the project is to develop and deploy an architecture for scheduling and monitoring field works of autonomous mobile robots used in agroecology practices. Autonomous agricultural robots represent an innovative solution for agroecology since they allow precise technical tasks and reduce environmental impacts.

With the advent of IoT, smart farming becomes a reality in the context of the agriculture domain. Farms are more frequently equipped with physical sensors [4] to acquire meteorological data such as rain, temperature, or soil moisture from the fields. Furthermore, autonomous robots are applied to handle technical operations, such as plowing [49].

As a business-like example of our *Superob* project, let us consider a scenario where a farmer needs to supervise the activities of some robots in a field. To this end, real-time data monitoring is necessary. Therefore, such a system must handle different types of data. In particular, we distinguish three basic data categories: stream, historical, and standard.

In this scenario, *Real-time streaming data* include, among others:

- *Trajectories of robots*, necessary to verify if a robot follows a scheduled trajectory, track the work in progress, and reschedule future tasks when necessary.
- *Meteorological data* (e.g., rain and wind data), necessary to check whether a given robot task can be done, e.g., some tasks such as spraying cannot be run when the wind is too strong. These data can be provided by some external weather services or by meteorological stations installed in the field. The choice depends on the needs and economic possibilities of the farmer. Using sensors provides more precise spatial scale data, but it is more expensive than free or commercial external meteorological services. In any case, these two different information sources must provide a minimal subset of equal attributes, such as air temperature and humidity.
- *Odometry robot data* (i.e., mechanical robot data), necessary to determine whether robots are experiencing any mechanical problems.
- *Scheduling data* (i.e., demands from farmers), necessary to define the organization of robots' tasks.

Historical data are crucial for decision-making. Analytical queries analyze such data. For example, historical data corresponding to the same robot in the same field and its technical operations allow comparing the current work to the past ones, to decide if the robot has abnormal behavior.

Finally, *standard data* are needed to complement real-time and historical data. Examples of standard data include: lists of plots (with their geometries and basic data), as well

as robots characteristics. These data represent the contextual information associated with other data and play the role of dictionary data.

Decision-support applications usually provide *computations* over data to calculate new indicators. In our scenario, we compute continuous queries on real-time data to create meteorological, robots-fault, and delay alerts. These alerts must also be stored since, as described above, they are useful for the decision-making process.

Data type requirement: The aforementioned scenario allows us to state that *agroecology IoT applications must cope with: (1) complex spatio-temporal data (such as robots trajectory), (2) stream data (such as weather information and robots data generated in real-time data), and (3) historical data.*

Non-functional requirements refer to systems and constraints, such as time computation or quality. In our context, since decision-makers must supervise their agricultural practices in real-time, they must agree to a set of non-functional requirements that must be supported by the system. In some cases, such as the computation time for robot faults, they must be involved in the definition step of these constraints. Therefore, we consider that taking into account non-functional requirements about data at the design step is a mandatory issue for agroecology IoT-based applications.

Moreover, the discussed data are deployed over the network architecture presented in Figure 1, which is typical for rural agricultural areas. As most farms are located in rural areas, the cellular network coverage might not be enough to ensure the QoS, which can be considered as network non-functional requirements, required by the application. Also, the use of cellular networks induces an additional cost for every node (sensors and robots) associated with the network. Consequently, they communicate and send their data through a standard local Wi-Fi connection in this architecture. They could also use other wireless technologies like *Zigbee* or *LoRaWAN*. Data from robots and sensors are thus sent to a workstation deployed on the farm. The workstation has a standard internet connection, through which data are sent to the cloud containing complex decision-making applications.

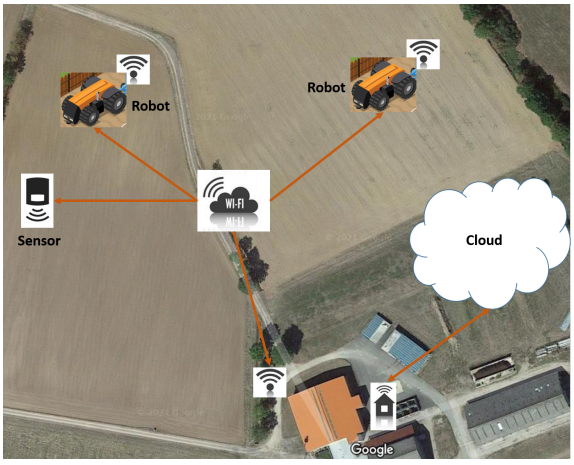


Fig. 1. Network communication example in a field

The network must ensure a particular QoS to meet the requirements of decision-makers. For example, Internet communication (e.g., an ADSL network) available in the farm must be fast enough to provide real-time data exchange with a cloud (e.g., online adjustment of scheduled tasks). The control of faulty robots using odometry data requires very low latency for real-time communications between a farmer and a robot. This latency cannot be satisfied on the end-to-end link with cloud servers, and thus it should be locally established. Hence, the local wireless network (Wi-Fi network) should be designed to answer these performance needs.

Non-functional requirements requirement: For the previously mentioned reason, *we argue that network performance indicators as well as data non-functional requirements must be integrated into the design of data-centric agroecology IoT applications*.

Depending on the network capacities of an Internet connection and a local Wi-Fi network, the distribution, storage, and processing of data would be modeled differently. For example, tasks requiring a high data rate and low latency (like remote control of faulty robots) should be executed on the farm and not on the cloud in case that the Internet connection does not support such a QoS.

Finally, let us note that data produced and consumed by the architecture components are strongly related to each other, which must be reflected in a data model. For example, a robot during its work must be associated with the plot where it is working, and it needs access to meteorological sensors' data of the plot.

Network communication also plays a crucial role in disseminating the information obtained through data analysis once they have been analyzed, either on a farm workstation or in the cloud. The information must reach its consumers, i.e., decision-makers of various roles who may be geographically distributed. To this end, visualization tools must be able to present the information asynchronously from multiple sources, producing data at different rates. Furthermore, some of the information must be communicated in real-time, e.g., rescheduling a robot that lost its trajectory or is malfunctioning.

From the above described example, we can conclude that the different actors involved in the design and development of agroecology applications are:

- *Agroecology stakeholders*, who define the application's requirements.
- *Information systems experts*, who are in charge of the implementation of the different systems to store and provide standard, historical and stream data.
- *IoT experts*, who provide the implementation of different IoT devices (e.g., sensors, robots, etc.).
- *Network experts*, who set up and configure the communication networks.

High quality model requirement: Since different actors are involved in the design step, the formalism used must be effective and grant important quality issues, such as understandability and reusability.

The usage of classical design and implementation methodologies is depicted in Figure 2. For each kind of data, a conceptual design step, and then its implementation, are **separately** applied. Then, the communication network configuration is provided. Finally, these different systems are coupled together to finalize the application deployment. Usually, this *merge* step raises several problems due to:

- *At design step*: Requirements are not well and exhaustively defined. Indeed, agroecology stakeholders must exchange separately with other actors. Therefore, it does not prevent them from having a global and unique vision of the defined requirements.

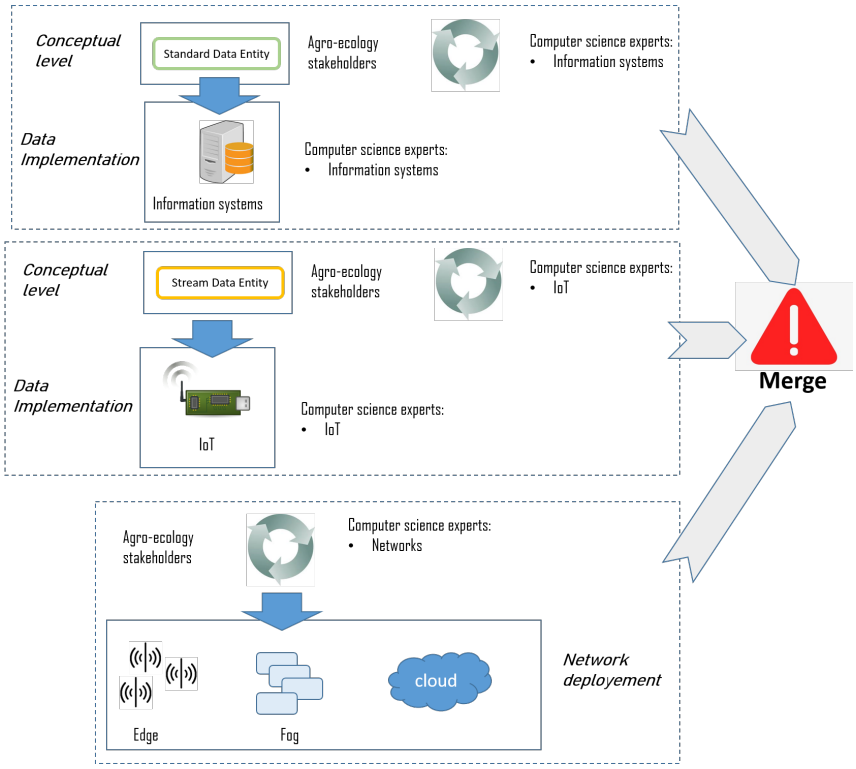


Fig. 2. Existing methodologies

– *At implementation step*: the risks related to (i) Possible incompatibility of the different systems implementation and the communication network communication constraints, and (ii) Manually generation of code representing associations among the different data, which is usually the source of technical and conceptual translation errors.

Integrated design and implementation requirement: The design step must be supported by a unique formal framework supporting different kinds of data and non-functional requirements and that makes transparent all implementation issues related to the usage of different technologies. Moreover, this formal framework could be used with other existing design methodologies to represent the dynamic aspects of the system.

Therefore, *there is the need for a unique data-centric conceptual model that allows all involved actors to exchange information about the requirements of the agroecology applications supporting different kinds of data and network communication issues*.

An example of the implemented web interface application is shown in Figure 3. The details of the implementation are shown in Section 5. Figure 3 clearly shows the different kinds of data involved in the application:

– meteorological data, which represent air humidity and temperature. These data are issued from the meteorological station.
– odometry data, which represent the robot speed. It is represented with a line chart.

- real time trajectory data. These data represent the real time position of the robot with a red line. The predefined trajectory is represented with a yellow line. A bullet point can be used to also visualize other odometry data in real time.
- background data. The visualized map is issued from google map or any other map server that could be used.
- video data collected by drone. This video is issued from a drone used in the experiment we have done.

3. UML Profile

In this section, which extends [7] with two new subsections,we present our UML profile for data-centric agroecology IoT applications. In Section 3.1, we present an overview of our UML profile, and then we detail the data and associations' representations.

3.1. Overview

Our UML profile provides a graphical and formal notation for functional requirements (in terms of data). A UML profile provides a generic extension mechanism for customizing UML models for particular domains and platforms. It is defined using stereotypes, tag definitions, and constraints applied to specific model elements, like Classes, Attributes, or Operations. We opt for an extension of UML elements of class diagrams since they are the de-facto standard to represent data.

Our UML profile allows designing all different kinds of data, and their associations, with the same UML Class diagram. Moreover, some network communication features can also be added inside this Class diagram. In this way, the design step of the agroecology application involves all the involved actors (agroecology stakeholders, information systems, IoT and network experts) at the same time. They share the same graphical formalism to exchange among them, which allows to avoid the *merge* step problems described in the Section 2. Moreover, the usage of Class diagrams allows using our UML profile with other tools provided by UML for the definition of functional and non-functional requirements, such as Use Case, Activity and Sequence diagrams (as shown in Section 4.2).

The design and implementation methodology for IoT applications based on our UML profile is depicted in Figure 4. The first step consists in the design of a conceptual model for all data involved in the applications and the associated QoS. This step concerns all the actors involved in the system (i.e. decision-makers - agroecology experts in our scenario, IoT, information system and network experts). This step can be iterative, and can include other UML diagrams to represent dynamic aspects of the application. Once an agreement about the conceptual model is found, the real implementation can be provided in different systems (sensors, database systems, etc.) for each data (by IoT and information system experts). Finally, the network communication is configured by network experts. In our approach, moving from the conceptual model to the implementation steps is feasible and do not require the intervention of decision-makers, since all the involved actors have reached an agreement about all data, IoT devices, and network configurations that will be used by the applications using the UML diagrams. This avoids the merge problems described in the previous section.

Fig. 3. Application user interface

Figure 5 shows the meta-model of our UML profile. In the next of this Section, we detail each element of the meta-model[8].

[8] A video describing the usage of the UML profile with Eclipse can be found here `www.youtube.com/watch?v=uTRewVj_eDs`

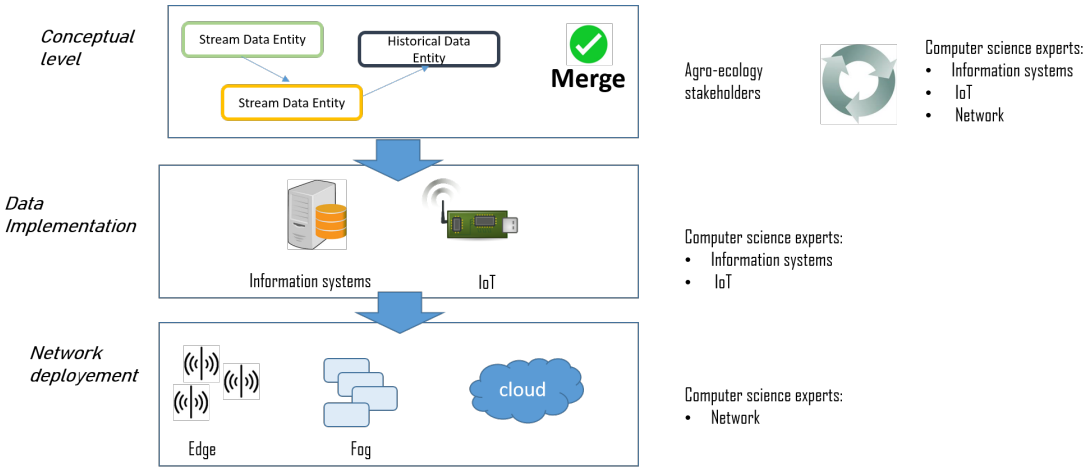


Fig. 4. Our approach

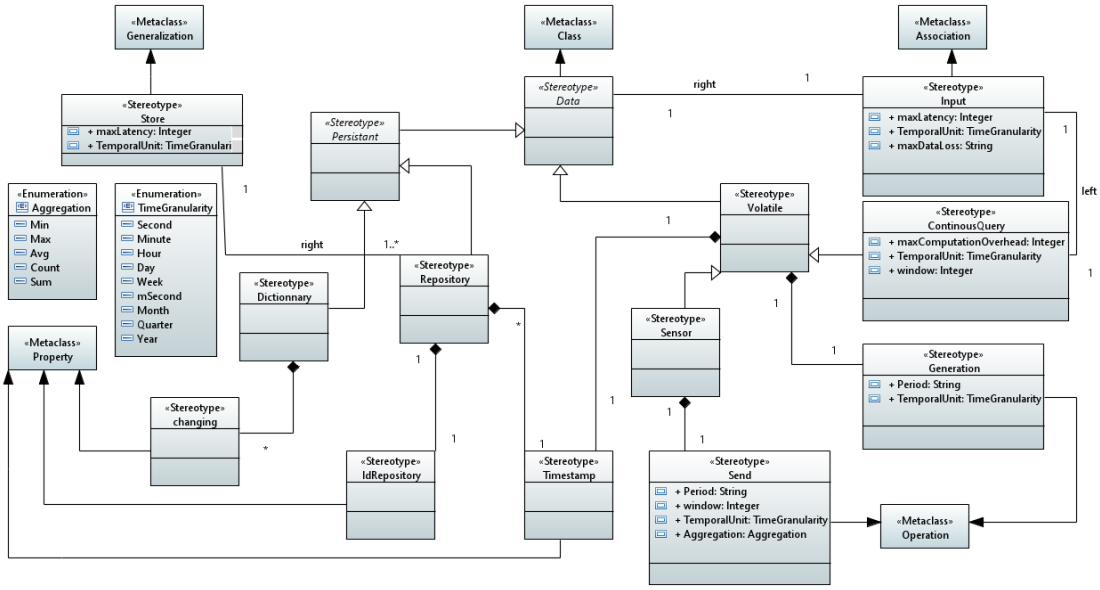


Fig. 5. The meta-model of our UML profile

3.2. Data

Data are classified into two main groups, namely, `Persistent` and `Volatile`, which are Class stereotypes. The `Persistent` stereotype is specialized in `Dictionary` and `Repository`.

`Dictionary` represents transactional data (i.e., standard data) that can be deleted, updated, and inserted in an On-Line Transaction Processing (OLTP) system. `Dictionary`

can include some attributes stereotyped as `Changing`. This stereotype means that attribute values can be updated, contrary to other attributes whose values do not change. The following associated Object Constraint Language (OCL) rule states that the attribute must not be changed (`isReadOnly=true`). Moreover, the `Dictionary` class must provide some attributes that uniquely identify its instances. This constraint is represented using the following OCL on such attributes: `isUnique=true`.

An example is shown in Figure 6, where the `Dictionary` stereotype is applied to *Robot*. This class presents (1) some standard attributes (e.g., *Name*, *SpeedMax*, *Weight*), and (2) some `Changing` attributes, like *Available*, which indicates when a robot is available for a particular task or is booked for another task within a given time slot.

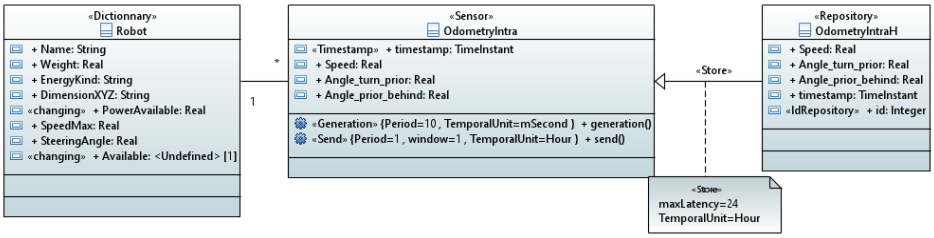


Fig. 6. Examples of *Dictionary*, *Sensor*, *Repository*, and *Store* association examples

`Repository` represents read-only historical data with the following characteristics:

- Attributes of the `Repository` class cannot be updated; only new values can be inserted. This constraint has been defined with OCL in the following way: `self.ownedAttribute->select(m|m.isReadOnly=false)->size()=0`.
- An instance of the `Repository` class cannot be deleted; it can only be inserted.

Moreover, `Repository` includes one attribute with stereotype `IdRepository` that uniquely identifies a datum in the collection of historical data (OCL: `ownedAttribute->select(m|m.oclIsTypeOf(IdRepository))->size()=1`). Finally, to model the temporality of the historical data represented by `Repository`, a `Timestamp` stereotype attribute is added, with an OCL constraint that forces it to have the `TimeInstant` type (OCL: `type.name='TimeInstant'`). Thus, `Repository` data represents historical data used for analytical purposes, such as OLAP or Machine Learning applications. An example is shown in Figure 6, where *OdometryIntraH* represents odometry historical data of robots.

`Volatile` represents data producers. These data are not permanently stored, and are characterized by a frequency generation represented by an operation with the `Generation` stereotype. `Generation` has two tagged values:

- `Period` that represents a temporal generation frequency, e.g., every second. In case of data generated on-demand, `Period` also accepts the *onDemand* value.
- `TemporalUnit` is the temporal granularity of `Period`. It takes values from enumeration `TimeGranularitiy`, e.g., second, minute, hour. This enumeration can be easily extended with other temporal types.

Moreover, `Volatile` also has one `Timestamp` attribute representing the time of the data generation.

The `Volatile` class is specialized into another class, called `Sensor`, which represents volatile data that are generated by physical sensors. `Sensor` extends `Volatile` with the `Send` operation, which represents the logic used for sending the data. It has the same tagged values of `Generation` (`Period`, `TemporalUnit`), and the following additional ones:

- `Window` represents a temporal window used to collect and process data before being aggregated and sent.
- `Aggregation` represents the aggregation function used on the collected data in the window before being sent. It takes values from enumeration `Aggregations`, e.g., Sum, Avg, Count (other aggregation functions can extend this enumeration).

It is crucial to specify these particular data sources at the design time since sensors must send data through a communication network, which can have substantial impacts on the system implementation.

An example of `Volatile` data are represented by the instances of class *Demand* -illustrated in Figure 7-B. This class model the activity requests of working tasks performed by a farmer. The instances of this class are generated on-demand. Consequently, tagged value `Period=onDemand`.

An example of sensor data is shown in Figure 7-A. It represents meteo data acquired by a sensor. Data (*wind, rain, temperature*, etc.) are collected each minute (`Period=1` and `TemporalUnit=minute`). Then, averages in a moving 10-minutes window are calculated.

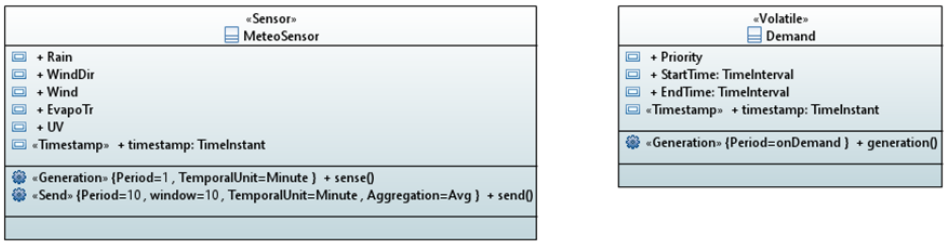


Fig. 7. Example instances of *Sensor* (**a**), and *Volatile* (**b**)

Commonly, a continuous query is executed over a data stream.

A continuous query is a query, which is re-computed continuously. For example, the query "Each minute, give me the average temperature of the last 10 minutes" will return different results depending on the current time.

In our UML profile, the stereotype `ContinuousQuery` represents a continuous query. It extends the `Volatile` stereotype with:

- `ComputationOverhead` tagged value, which represents the maximum time to compute the query.

– `Input` directed association, which represents the input data used for the query. `Input` has two tagged values: `maxLatency` and `maxDataLoss`. `maxLatency` represents the maximum tolerated time for input data to be transmitted into the system that implements `ContinuousQuery`. `maxDataLoss` represents the percentage of data that can be lost. These QoS constraints are issued from the application logic and come from the fact that data are generated in different points of a network, as described in the IoT architecture. Other network performance constraints exist; yet, they correspond to non-functional requirements (NFR), but not to the application logic. For instance, bandwidth is associated with a particular implementation of attribute data types (in terms of bytes used). Such NFR constraints should be represented at the Platform-Specific Model level following the Model-Driven Architecture, while our UML profile would correspond to the Platform Independent Model level.

The NFR are used to guide the implementation of the system. They impact the choice of the components of the system. For example, a low `ComputationOverhead` for the DSMS component implementing the query could necessitate a distributed DSMS, or a low `maxLatency` could lead to the use of a new communication network such as 5G instead of ADSL. If the NFR are not met temporarily then the multi-representation solution can be applicable. Multi-representation has been defined for classical data, and in particular for Geographic Information Systems [56], as different representations and computations of the same entity data according to different rules.

Figure 8 shows an example of `ContinuousQuery`. *AlertDelayQuery* computes in real-time the delay of a robot according to its predefined trajectory. It takes as inputs: *Point-Time*, which represents the real time position of the robot, and *TrajectoryRef*, which represents the planned trajectory. The tagged value of the *Input* association states that these GPS data must be received in real-time for the alert delay computation. Moreover, *AlertDelayQuery* is computed each minute using the last 5 minutes of received data, and 5% of GPS data can be lost, contrary to *TrajectoryRef* that cannot be affected by data loss (i.e. all data of the trajectory of reference must be present). End-users define the configuration of *AlertDelayQuery* parameters.

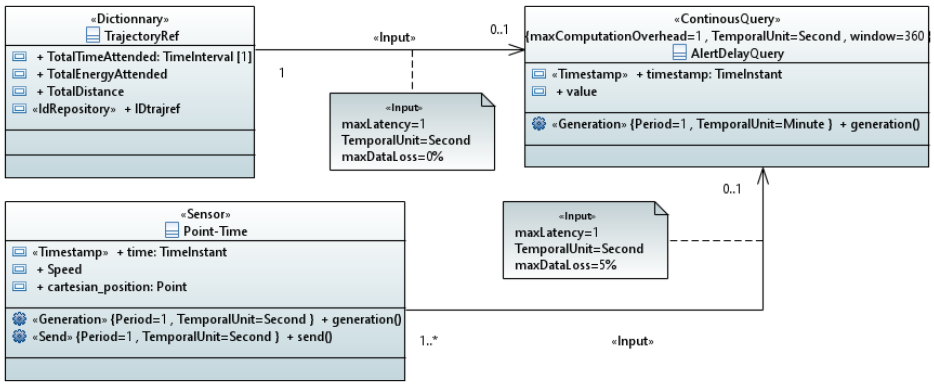


Fig. 8. An example of stereotype *ContinuousQuery*

Other kinds of queries are compatible with our approach. For example, it is possible to provide multidimensional queries (i.e. read only on-line queries over warehoused data - such as "*What is the average temperature per plot and month*?"), or transactional queries (i.e. update and write queries - such as "*Update the location of the sensors A*") [9].

3.3. Associations

This section describes how `Volatile` and `Persistent` data can be transparently associated to define a single coherent model.

From Figure 8, we can notice that any data can be associated with `ContinousQuery` via `Input` association. Moreover, `Volatile`, `Sensors`, and `ContinousQuery` can be associated with `Repository` data via the `Store` association. This association means that initially volatile data are made persistent by using the `Repository` class. As for the `Input` association, it has a `maxLatency` tagged value. This value represents a maximum time within which data must be stored in a repository. Since volatile data could be sent through a communication network they cannot be stored immediately, thus we use `maxLatency` value.

For example, Figure 6 shows that data collected by *OdometryIntra* into the robots (odometry data collected 100 times per second, and sent every hour) are stored into the `Repository` class *OdometryIntraH*. The `Store maxLatency` value is 24 hours since these data are stored in a data warehouse refreshed every 24h.

The association between `Store` and `Repository` is a generalization, because the `Repository` class must include in its structure all the attributes and associations of classes `Volatile`, `Sensor`, and `ContinuousQuery`. Moreover, `Repository` must not present methods of `Volatile` (OCL: `ownedOperation->size()=0`). Therefore, the `Store` association represents a total cloning operation of the `Volatile`, `Sensor`, and `ContinuousQuery` data in persistent storage.

Let us consider the example of Figure 6 again. If a persistent storage stored only the values of the odometry attributes, such data would be incomplete. Note that *OdometryIntra* is associated with `Robot`. Without the associated robot that generated these data, it would not be possible to identify the robot that has generated such odometry data.

To conclude, this data-centric representation of all kinds of data and queries allows us to associate all these data among them without considering if the data is classical data, or sensor data or stream data or data resulting from computations.

Therefore, our proposal satisfies the *Data types* and *non-functional requirements* described in Section 2.

4. Assessing proposed UML profile

In the above section, we have pointed out how our UML profile can be easily used to represent different kinds of data and non-functional requirements, as described in section 2. Therefore, in this section, we provide some theoretical and practical evaluation to show how the other defined requirements are supported by our proposal.

4.1. Assessing the quality of the proposed meta-model

In this section, we provide a quantitative validation of the quality of our UML profile following the framework proposed in [28], in order to show how our proposal satisfies the *High quality model requirement* identified in Section 2. The framework proposed in [28] allows measuring the quality of a metamodel using five metrics calculated from the metamodel with a three step process. First, the following metrics are computed:

- ANDM: the average number of direct associations between metaclasses.
- ANM: the average number of attributes.
- ANMC: the average number of direct association between a metaclass and other kinds (operation, property,...).
- ANR: average number of OCL constraints.
- NOH: the number of inheritance hierarchies
- ADI: the average depth of inheritance trees
- ANA: the average number of direct inheritance between metaclasses
- NAM: the number of abstract metaclasses
- NCM: the number of concrete metaclasses

Then, using the above presented metrics, some global metrics (such as modeling concepts size, abstract metaclass size, intension, coupling, ...) are computed. For instance, coupling that means the level of interdependence between the classes of a diagram, is computed as the sum of ANDM and ANA.

Finally, five ultimate metrics (calculated using the metrics of the second step provided.

- *Reusability*: it measures the ability of a metamodel's components to contribute to the definition of different metamodels (e.g., in other application domains).
- *Understandability*: it represents the degree of ease to understand and to use the content of a metamodel by end-users.
- *Functionality*: it measures the number of concrete metaclasses which reflect the strength of modeling ability of a metamodel.
- *Well-structuredness*: it represents how a meta-model is well-structured by measuring the structure quality of its architecture by means of its metaclasses.
- *Extendibility*: it measures the ease to add new modeling element to a metamodel. It is computed as **0.2 × Coupling + 0.3 × (Modeling concepts size + Abstract metaclass size))):**

Moreover, these quantitative measures enable quality comparisons against other metamodels. Indeed, [6] evaluates and compares more than 2500 UML metamodels from the literature using this framework ([28]). Similarly, we compare our UML profile with those works using the percentile rank. In this way, we can assess if our metamodel is outstanding, regular, or particularly bad on each quality measure.

Figure 9 shows the results of such comparisons. When compared to more than 2500 UML metamodels analyzed in [6], it is evident that:

- Our profile excels in Understandability and Well-structuredness, which are really important due to the diversity of actors involved in the design phase as described in Section 2.

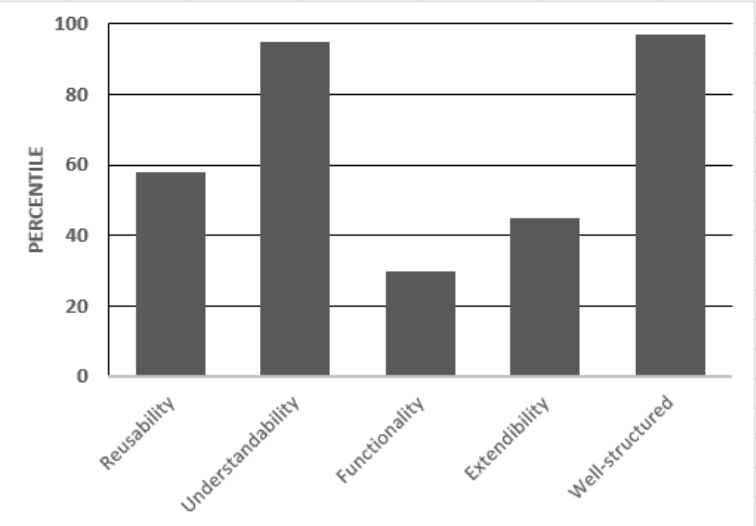


Fig. 9. Comparison our metamodel according to the framework of [28], in the context of all the meta-models considered in [6]

- Its Reusability and Extendibility are around the median. Therefore, it will not demand significant efforts to add new data elements, such as multimedia data.
- Only the functionality is below average, but it is not particularly bad. A low functionality usually means that the range of applications that our metamodel covers is narrow. However, we must consider that we use highly abstracted concepts in our profile, and thus each profile element relates to multiple implementation possibilities.

Considering the outcomes from this theoretical assessment, we infer that the quality of our UML profile is appropriate for the design of agroecology applications.

4.2. Assessing the integrated design

In the next, we describe how our UML profile can be used in a classical software UML based design methodology. For simplicity we exclusively focus on the class diagram represented in Figure 6. UML use case and activity diagrams are well recognized as effective tools for collecting and formalizing requirements [32]. These diagrams are then used to deduce Class diagrams. In our case study, one main functionality of the monitoring system is the analysis of historical odometry data, which are collected by robots in the field. The use case diagram showing this task is represented in Figure 10. It presents two actors, the robot that collects the odometry data in real time, and the repository system (i.e., database system) that stores all data collected by all robots during their work (i.e., historical odometry data). Therefore, the following classes can be deduced: *Robot*, *OdometryIntra* (for real time odometry data) and *OdometryIntraH* (i.e., for the Repository actor).

According to this use case, the associated activity diagram is shown in Figure 11. From this activity diagram, it is possible to deduce the need for (i) a generation method for *OdometryIntra*, (ii) a method that represents sending data over the network (from robot

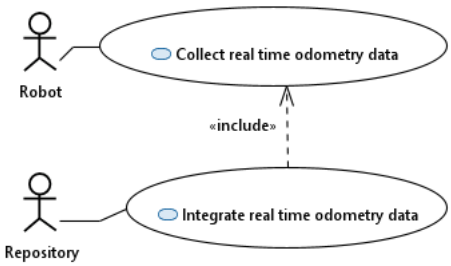


Fig. 10. Use case diagram

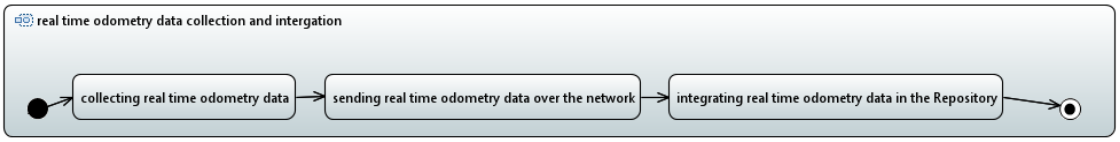


Fig. 11. Activity diagram

to the repository) (*send()*), and (iii) a store method to represent the integration of real time odometry data in the repository (i.e. *OdometryIntraH*).

Sequence diagrams can be used to express time constraints. Therefore, the need for the decision-makers to be able to analyze the last 24 hours collected odometry data could be expressed by means of a sequence diagram. In this work, in order to keep the UML models as simple as possible, instead of using a sequence diagram, we opt for using our UML profile since it is possible to represent this constraint using the `Store` association and its tagged value. Therefore, `Store` association represents the integration of real time data into historical data and the temporal constraint. At this point, using the use case and the activity diagrams we have easily obtained a skeleton of the three main classes and their associations of Figure 6. Finally, discussing with the agroecology stakeholders, the IoT, information systems and network experts can complete the class diagram to obtain the final one depicted in Figure 6. Indeed, using this skeleton of class diagram makes it more simple for agroecology stakeholders to define the details of each class, and therefore the choice of the right stereotype. At the end, by means of our UML profile, agroecology stakeholders are aware about the volatile (or not) character of data involved in the system and the fact that data are exchanged over a communication network.

From the above described example, we can conclude that our proposal supports the **Integrated design and implementation requirement** described in Section 2.

5. Implementation

This section presents the implementation in a commercial CASE tool, and how each type of data of our agricultural case study is implemented (Section 5.2), and we detail its corresponding IoT architecture (Section 5.3).

5.1. CASE tool implementation

In this section, we present the implementation of our UML profile by using Papyrus,an open-source software for UML modeling based on Eclipse. Papyrus supports the creation of UML profiles by specifying instances of the different UML meta-elements (e.g., stereotypes of properties, classes, operations, or OCL constraints). Papyrus allows checking OCL constraints at design time. For example, let us consider Figure 12. It shows how Papyrus checks that the constraint: (OCL:`ownedAttribute->select(m|m.oclIsTypeOf (IdRepository))->size()=1`)) for `Repository` (which indicates `Repository` must include one attribute with stereotype `IdRepository`).

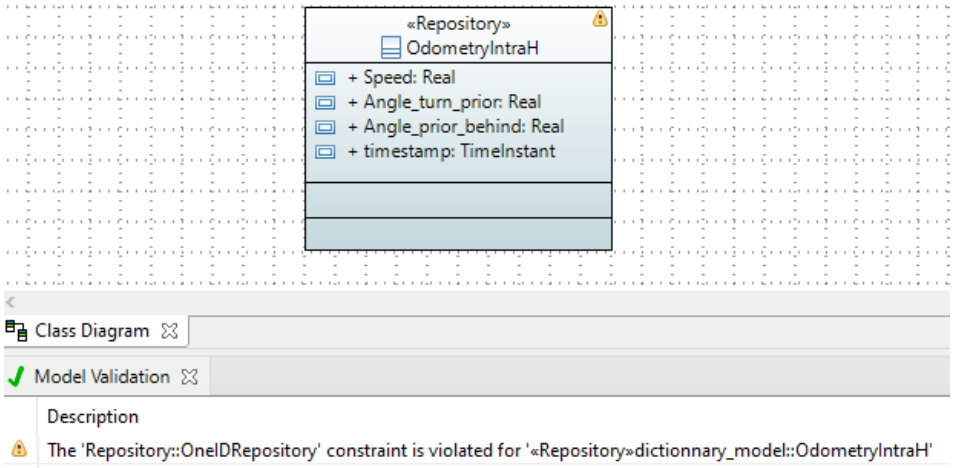


Fig. 12. Example of OCL constraint check with Papyrus

The UML profile implementation is available as open source project [9].

5.2. Data implementation

The implementation of our case study requires a complex digital ecosystem. Despite a uniform representation of data at the conceptual level, the different kinds of data must be generated and handled by diverse subsystems. For instance, `Sensor` and `ContinousQuery` data require an implementation using programming languages and Data Stream Management System. `Persistent` data either in a classical storage system (such as a relational database) or in a novel storage system (such as NoSQL systems when scalability is needed) can be deployed.

Consequently, in our case study, each component of the digital ecosystem have a particular implementation.

[9] https://www6.inrae.fr/tools4bi/Design/A-UML-Profile-for-Agroecology-data-centric-applications-design
We will provide it for download after the acceptance of the paper

Meteorological data (represented by the `Sensor` *MeteoSensor* in Figure 7-A) is implemented in an IoT device running *RIOT OS* [10]. RIOT is an open source operating system that supports several low-power IoT devices, micro-controller architectures (32-bit, 16-bit, 8-bit), and external devices. Applications for this OS are written in C and must specify the behaviour of each involved device [5]. Figure 13 shows two fragments of the code associated to *MeteoSensor* (Figure 7-A).

```
1.    /* Code for thread 'AirTemperature_avgThread' */
2.    /* Stack of memory for thread 'AirTemperature_avgThread' */
3.    static char AirTemperature_avgThread_memstack[THREAD_STACKSIZE_MAIN];
4.    /* Thread 'AirTemperature_avgThread' */
5.    /* Thread to sense and transform one variable */
6.    static void *AirTemperature_avgThread(void *arg)
7.    {
8.          (void)arg;
9.          while (1) {
10.                 AirTemperature_avg_dataStruct internalData;
11.                 //gather the variables
12.                 for (int16_t w=0; w< 10; w++) {
13.                           lpsxxx_read_temp(&lpsxxx, &internalData.avgTemp_src[w]);
14.                           internalData.avgTemp_src_lastplace = w;
15.                           xtimer_sleep(60);
16.                 }
17.                 //Apply average with order 0
18.                 // Aggregation: 'average'
19.                 int16_t sum = 0;
20.                 for (int8_t  i=0; i<=internalData.avgTemp_src_lastplace; i++){
21.                           sum += internalData.avgTemp_src[i];
22.                 }
23.                 int16_t average = sum / (internalData.avgTemp_src_lastplace + 1);
24.                 internalData.avgTemp_src[0] = average;
25.                 internalData.avgTemp_src_lastplace = 0;
26.                 // Save the internal data into the public struct:
27.                 mutex_lock(&public_AirTemperature_avg.lock);
28.                 public_AirTemperature_avg.data = internalData;
29.                 mutex_unlock(&public_AirTemperature_avg.lock);
30.         }
31.         return 0;
```

Fig. 13. Sensor implementation

The first code fragment (Figure 13) is the sensing and aggregation thread. To begin, this thread samples the plot temperature 10 times with a periodicity of one minute (i.e., during 10 minutes). Then, it calculates the average temperature of the plot of the last 10 minutes and saves it as a public variable. Finally, the thread process starts again to run indefinitely.

It is important to note that the data implementation strictly follows its conceptual definition. The IoT code (Figure 13) senses data every minute, and calculates the average and sends the data every 10 minutes as specified in *MeteoSensor* (Figure 7-A).

Odometry data (*OdometryIntra* in Figure 6) are implemented in Python in the *Fleet of Robots* using Robot Operating System (ROS). Besides, robots have tasks, trajectories, and timing constraints (e.g., indicated speed).

[10] https://www.riot-os.org/

`Persistent` data (e.g., *OdometryIntaH* in Figure 6) are stored in the relational spatial DBMS PostGIS. These data are further loaded into a *Data Warehouse* implemented in Mondrian and JRubik to analyze them.

The Data Warehouse storage is provided by PostGIS. It is important to note that the implementation of `Persistent` data needs some particular SQL statements. Indeed, attributes with the `Changing` stereotype are classical ones, contrary to the other ones that cannot be updated. This constraint is implemented in SQL with a trigger on the UPDATE SQL statement. An example for the *name* attribute of the *Robot* class is shown in Figure 14.

```
1. CREATE OR REPLACE FUNCTION not_changing()
RETURNS trigger AS
$BODY$
    BEGIN
        RAISE EXCEPTION 'no way';
    END;
$BODY$
  LANGUAGE plpgsql VOLATILE

2. BEFORE UPDATE OF Name
  ON Robot
  FOR EACH ROW
  EXECUTE PROCEDURE not_changing();
```

Fig. 14. SQL implementation example

The *AlertDelayQuery* continuous query (Figure 15) is implemented in Scala (the Sedona framework, which is a spatial extension of Apache Flink). This query joins GPS data coming from the robots (`Sensor` *Point-Time*) with data stored in PostGIS (`Dictionary` *TrajectoryREF*) (Line 1).

Then, the delay is computed (Line 2). Data is collected in a window of 1 minute (Line 3), and each 1 minute data is sent using the average aggregation function (Line 4).

```
1. val results=jdbcDF.as("a").join (TrajDf.as("c")).where("a.idDB
   = c.id") //Retrieve the table with the reference trajectory using the full join query

2. val delay=
   results.withColumn("DiffInSeconds",col("stampDB").cast(LongType
   ) - col("stamp").cast(LongType)) // compute the delay and add the result to
   the new column.

3. val windows1 = delay .groupBy(window($"stamp", "60 second", "60
   second"), $"id",$"lat",$"long", $"idDB",$"latDB",$"longDB",
   $"DiffInSeconds") //Group the trajectory data by window and delay.

4. val aggregatedDF1 = windows1.agg(avg("DiffInSeconds")) //aggregate
   the window results and calculate the average delay of each window group.
```

Fig. 15. Spark Streaming implementation example

Moreover, associations between `Dictionary` and `Repository` can be simply implemented using foreign keys in a relational DBMS or integrity constraints in a NoSQL DBMS. However, such mechanisms do not exist for `Volatile` data (`Sensor` and `ContinousQuery`). They require an ad-hoc method: the data structure sent by sensors must include their identifiers, such as previously described above for the meteorological sensor.

5.3. Multi-layer network architecture

This subsection is issued from [7]. Inspired by the Lambda reference architecture for IoT applications [48], we propose an architecture to host the technologies that handle the data required in our case study. It is composed of three main layers: *Field*, *Farm*, and *Cloud* (Figure 16). The *Field* and *Farm* layers are implemented in each farm. In contrast, the *Cloud* layer is implemented only once for any number of farms.

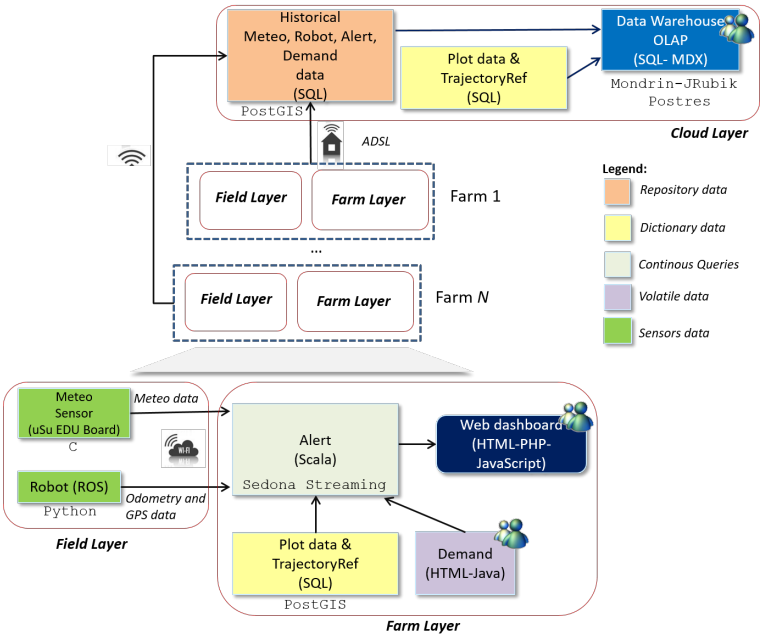


Fig. 16. Architecture implementation - data mapping

The *Field* layer represents different data sources deployed in the field (i.e., `Sensor` data). In our case study, it is composed of *MeteoSensor*s and *Robot*s, which provide data and execute specific tasks. These IoT devices do not have direct Internet access on the fields (there is no cellular network coverage). Thus, we deploy a Wi-Fi network to collect these data at the *Farm* layer, which connects to the Internet through ADSL.

However, ADSL may not guarantee the QoS required by the `ContinuousQuery` of our case study. For instance, the `Input` association between *AlertDelayQuery* and

TrajectoryREF (Figure 8) requires a maximum latency of 1 second with no data lost. Therefore, we implement these queries in the *Farm* layer. Moreover, we host a DBMS for *TrajectoryREF* in the same layer (*Farm*) since this allows for a local connection to the DBMS with less latency and data loss issues. Finally, the *Farm* layer only sends processed data streams to the *Cloud* layer.

The *Cloud* layer implements the storage of `Repository` and `Dictionary` data coming from all the farms. This layer hosts a data warehouse that analyzes the historical odometry data of the robots, providing an inter-farm analytical vision of all the available data.

This way, our case study shows that the proposed UML profile can effectively represent the multiple kinds of data present in a complex IoT-based application. The data was implemented in different languages, systems, and infrastructures that must successfully communicate and cooperate to provide useful decision support.

6. Related Work

In this section, which extends [7], we present some of the most relevant contributions related to our proposal comparing them according to the requirements defined in Section 2.

Regarding the conceptual design and implementation of IoT, [13] proposes a UML profile for IoT physical devices considering fog and cloud concepts for objects interoperability and reusability. In the same way, [34] provides an automatic implementation framework for modeling different IoT systems using simple drag-and-drop designs. However, these works focus strictly on abstracting and solving (highly relevant) issues for the physical implementation of IoT rather than on data definition or integration. Other works have successfully integrated IoT data into different systems. For instance, [37] provides additional semantics to the design of wireless sensors networks to ease the further use of generated data. [26] defines a conceptual model for integrating IoT data into digital twins. Besides, [2] provides a meta-model for the integration of IoT data into web services.

In the section, we present existing works about designing complex and IoT data. [40] and [41] provide a survey of existing conceptual models for sensors and IoT data. They state that no existing work defines a data-centric model for data issued from sensors and IoT devices, respectively. So the authors propose a UML profile for modeling IoT data, which makes transparent all technical details related to the implementation. Moreover, [39] extends [40] to integrate sensors data with Stream Data Warehouses. However, these works do not support volatile data in the form of continuous queries, polyglot data associations are not possible, and non-functional requirements. Consequently, even through multiple approaches for building IoT architectures have been proposed, to the best of our knowledge, these approaches fail to provide a comprehensive formalism for representing volatile and persistent data while hiding the complex technical issues of their implementations.

In the context of database systems, numerous modeling methods (based on ER or UML) have been contributed for standard and temporal databases. For example, [11], proposes the usage of UML to represent temporal data properties, while [55] details an extension of the ER model. In the same way, some studies propose conceptual models for Data Warehouses, which can be considered as a particular kind of persistent data, through

UML [10] or ER extensions [12]. However, all these works do not take into account volatile data.

Conceptual models for stream data have not received much attention from the research community so far. To the best of our knowledge, only [9] proposed a UML profile for representing continuous queries integrated into a data warehouse approach. However, sensors data cannot be modeled nor non-functional requirements.

Numerous works studied the conceptual representation of Extraction, Transformation, Loading (ETL) processes. Extraction and loading are the most frequently researched operators (tasks). [50] and [3] proposed formal representations of the operators. They are similar to our *Input* and *Store* associations stereotypes. ETL models based on formal process notations, such as BPMN or UML Activity diagrams, are not adapted for our data-centric approach (as reviewed in [3]) since they model data and operations into separate diagrams. Only [50] adopted a data-centric approach. However, the approach is applicable to modeling only relational systems, and it ignores QoS notions, such as the *Latency* constraint, which we found necessary to model streaming-data applications as in IoT.

Non-Functional Requirements (NFR) can be defined as constraints attributes to define system quality and how it should perform. They are considered as one of the main aspects for the success of a system, because they provide an enormous help to understand at an early stage various problems of the system implementation. According to [21], NFR are goals to be achieved in the design pattern. Not taking into account NFR may generate more risks than functional requirements (FR) [20]. Nowadays NFR are more and more demanded by stakeholders, but they are mostly neglected or poorly handled. This affects the stakeholders decision making process, thus understanding and integrating NFR in system design can lead to better user experience and cost reduction.
In this context, several works have tried to give more importance to NFR, for instance [51] introduces an architecture with a process to separate, identify and integrate NFR, and [15] proposes a UML extension to define NFR and integrate them into different UML diagrams. However those works do not address IoT data.

In the context of IoT, to the best of our knowledge the majority of artifacts focus on the reliability and usability of the solution, but other NFR are not well taken into account [25]. [45] presents an agile approach to handle NFR such as (security and performance) in scrum, and [47] presents a NFR template and shows that NFR (cost, sensitivity, design complexity, storage, development process, environmental impact) can be very helpful to enhance IoT systems. [30] provides a UML-based approach to represent a variety of NFR in telecommunication domains, [8] proposes an MDA approach to handle the energy consumption of wireless sensor network using SysML and Modelica languages. However, these works do not focus on data design, and they do not take into account data and network NFR at the same time.

In the context of database systems, [36] shows that most of database design do not address NFR, and that in the future of database performance era, more NFR should be taken into consideration. Some works insist on the importance of NFR in the conceptual design [16],[14]. Indeed, some works propose to integrate NFR of the design process, such as [19] that propose an approach based on MDA and NFR integration to build database design, and [35] that details an approach with five steps to take NFR in consideration

before suggesting the appropriate database design (conceptual model, FR definition, NFR definitions, model fragmentation, database model).

[16] propose an ER framework that integrates NFR (validation, delivery,reliability and authorization) in the conceptual data models. In the same way, [27] propose to enrich the ER model with the workload of each entity in order to automatically generate the best NoSQL logical schema. These contributions remain focused on static persistent data, ignoring the particular features and challenges of volatile data, and network NFR.

Some works present NFR for real-time databases. [24] provides a UML profile for real time database modeling with temporal constraints and data quality, and [18] presents a UML package for real time objects with temporal constraints. Those works focus on temporally-constrained data, which is similar to our volatile data. Nonetheless, they disregard the existence of multiple subsystems that rely on particular technologies (e.g., sensors as the data generators) and the explicit association of different kinds of data.

Table 1 provides a summary and a comparison of most important works based on a data-centric approach previously described according to the requirements we have defined in Section 2: (i) Integrated design and implementation, which means the usage of a formal framework that can be used for representing also dynamic aspects of the system, (ii) The non-functional requirements for data and network, (iii) Types of data supported.

Table 1. Comparison of related work

Work	Integrated design and implementation	Non-Functional Requirements	Data Types
[40]	Yes	No	Partial (sensors data)
[41]	Yes	No	Partial (IoT devices data)
[39]	Yes	No	Partial (sensors data and stream data warehouse)
[11]	Yes	No	Partial (dictionary data)
[55]	No (ER formalism)	No	Partial (dictionary data)
[9]	Yes	No	Partial (not sensors data)
[16]	No (ER formalism)	Data (Dynamic - such as Usability, and Static - such as Accuracy)	Partially (dictionary data)
[27]	No (ER formalism)	Data (such as Volume)	Partially (dictionary data)
[24]	Yes	Data (such as Temporality)	Partial(sensors and dictionary data)
[18]	Yes	Data (Temporality)	Partial (dictionary and stream data)
Our Approach	Yes	Data (Temporality and Performance), Network (Latency)	All (persistent and volatile data)

From table 1 that reports only data-centric proposals, we can notice that all the works focus on NFR that concern data, and the majority of works only target few data types, and some of them are based on the ER formalism, which does not allow to represent dynamic aspects.

To conclude, existing works do not provide a unique and global conceptual representation of all involved data for complex IoT-based applications.

7. Conclusion and Future Work

IoT technologies are more and more used in all application domains, such as urban, health, tourism, etc. IoT provides decision-makers with complex real-time data at different spatio-temporal scales.

However, the adoption and deployment of IoT technologies raises a challenging research agenda related to standardizing design artifacts for IoT, sketching scalable architectures, and devising new algorithms for efficiently managing and processing IoT data at different levels. In particular, in the agriculture and agroecology context, IoT projects feature complex requirements, involving both heterogeneous hardware systems (e.g., robots, sensors, networks' hardware, and protocols, both in-situ and cloud servers), heterogeneous software systems, and complex spatio-temporal data. This complexity makes the design of conceptual data models of agroecology IoT applications mandatory for successful projects. The modeling has to encompass: (1) all heterogeneous data involved (i.e., from streamed sensors data, spatio-temporal data, to classical static and computed data) and (2) communication networks features.

Therefore, motivated by the lack of such a comprehensive conceptual framework, in [7], we proposed a UML profile to design data-centric agroecology IoT applications. We applied the UML profile for the monitoring of autonomous agricultural robots. Apart from feasibility implementation of the UML profile in a Big Data architecture, [7] does not present any validation of the proposed approach. Therefore, in this work, we provide a theoretical assessment of the UML profile based on existing metrics. Our experiments show the efficacy of our UML proposal from a theoretical point of view. Moreover, we provide a design and implementation methodology based on our approach, that can be integrated in classical existing software engineering methodologies. In order to validate this feature, we have shown by means of our real case study, how the UML profile class diagrams can be transparently used by UML sequence and activity diagrams.

Our UML profile does not allow representing multimedia data (video and images) that are commonly used in agriculture applications. Therefore, we plan to extend our profile to also represent multimedia data. Finally, setting the optimal (in practice sub-optimal) QoS values is challenging and it is considered a difficult optimization problem. A promising approach to supporting parameters and performance tuning is based on machine learning (ML) algorithms, e.g., [22,23,42]. Such algorithms require large volumes of test data to learn reliable performance models. Thus, excessive experimental evaluations are needed to provide performance data, to feed ML algorithms. In our project, tuning the parameters will be based on excessive experiments, therefore, we will address this issue in future work.

Acknowledgments. This work is supported by the French National Research Agency projects ANR-19-LCV2-0011 *Tiara*, and French government IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25).

References

1. Al-Sarawi, S., Anbar, M., Alieyan, K., Alzubaidi, M.: Internet of things (iot) communication protocols: Review. In: Proceedings of the 8th International Conference on Information Technology (ICIT). pp. 685–690. IEEE, Amman, Jordan (2017)
2. Alulema, D., Criado, J., Iribarne, L., Fernández-García, A.J., Ayala, R.: A model-driven engineering approach for the service integration of iot systems. Cluster Computing 23, 1937–1954 (2020)
3. Awiti, J., Vaisman, A.A., Zimányi, E.: Design and implementation of ETL processes using BPMN and relational algebra. Data & Knowledge Engineering 129, 101837 (2020)

4. Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., Aggoune, E.M.: Internet-of-things (IoT)-based smart agriculture: Toward making the fields talk. IEEE Access 7, 129551–129583 (2019)
5. Baccelli, E., Gündoğan, C., Hahm, O., Kietzmann, P., Lenders, M.S., Petersen, H., Schleiser, K., Schmidt, T.C., Wählisch, M.: Riot: An open source operating system for low-end embedded devices in the iot. IEEE Internet Things Journal 5(6), 4428–4440 (2018)
6. Basciani, F., Di Rocco, J., Di Ruscio, D., Iovino, L., Pierantonio, A.: A tool-supported approach for assessing the quality of modeling artifacts. Journal of Computer Languages 51, 173–192 (2019)
7. Belhassena, A., Bimonte, S., Battistoni, P., Cariou, C., Chalhoub, G., Corrales, J.C., Laneurit, J., Moussa, R., Plazas, J.E., Wrembel, R., Sebillo, M.: On modeling data for iot agroecology applications by means of a UML profile. In: Proceedings of the 13th International Conference on Management of Digital EcoSystems. pp. 120–128. ACM, Virtual Event,Tunisia (2021)
8. Berrani, S., Hammad, A., Mountassir, H.: Mapping sysml to modelica to validate wireless sensor networks non-functional requirements. In: Proceedings of the 11th International Symposium on Programming and Systems (ISPS). pp. 177 – 186. IEEE, Algiers, Algeria (2013)
9. Bimonte, S., Schneider, M., Boussaid, O.: Business intelligence indicators: Types, models and implementation. International Journal of Data Warehousing and Mining 12(4), 75–98 (2016)
10. Boulil, K., Bimonte, S., Pinet, F.: Conceptual model for spatial data cubes: A UML profile and its automatic implementation. Computer Standards & Interfaces 38, 113–132 (2015)
11. Cabot, J., Olivé, A., Teniente, E.: Representing temporal information in UML. In: Proceedings of the 6th International Conference The Unified Modeling Language, Modeling Languages and Applications. pp. 44–59. Springer, San Francisco, CA, USA (2003)
12. Combi, C., Oliboni, B., Pozzi, G., Sabaini, A., Zimányi, E.: Enabling instant- and interval-based semantics in multidimensional data models: the t+multidim model. Information Sciences 518, 413–435 (2020)
13. Costa, B., Pires, P.F., Delicato, F.C.: Towards the adoption of omg standards in the development of soa-based iot systems. Journal of Systems and Software 169, 110720 (2020)
14. Cysneiros, L.M., Julio Cesar, S.d.P.L.: Integrating non-functional requirements into data modeling. In: Proceedings of the 4th International Symposium on Requirements Engineering. pp. 162–171. IEEE Computer Society, Limerick, Ireland (1999)
15. Cysneiros, L.M., Julio Cesar, S.d.P.L.: Non functional requirements: From elicitation to conceptual models. IEEE Transactions On Software Engineering 30(5), 328–350 (2004)
16. Cysneiros, L.M., do Prado Leite, J.C.S., de Melo Sabat Neto, J.: A framework for integrating non functional requirements into conceptual models. Requirements Engineering 6(2), 97–115 (2001)
17. Dalgaard, T., Hutchings, N., Porter, J.: Agroecology, scaling and interdisciplinarity. Agriculture, Ecosystems & Environment 100(1), 39–51 (2003)
18. DiPippo, L.C., Ma, L.: A uml package for specifying real-time objects. Computer Standards & Interfaces 22(5), 307–321 (2000)
19. Dubielewicz, I., Hnatkowska, B., Huzar, Z., Tuzinkiewicz, L.: Feasibility analysis of mda-based database design. In: Proceeding of the International Conference on Dependability of Computer Systems. pp. 19–26. IEEE Computer Society, Szklarska Poreba, Poland (2006)
20. Ebert, C.: Putting requirement management into praxis:dealing with nonfunctional requirements. Information and Software Technology 40(3), 175–185 (1998)
21. Gross, D., Yu, E.: From non-functional requirements to design through patterns. Requirement engineering 6(1), 18–36 (2001)
22. Hernández, Á.B., Pérez, M.S., Gupta, S., Muntés-Mulero, V.: Using machine learning to optimize parallelism in big data applications. Future Generation Computer Systems 86, 1076–1092 (2018)

23. Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, F.B., Babu, S.: Starfish: A self-tuning system for big data analytics. In: Proceedings of the Fifth Biennial Conference on Innovative Data Systems Research. pp. 261–272. www.cidrdb.org, Asilomar, CA, USA (2011)
24. Idoudi, N., Duvallet, C., Bouaziz, R., Sadeg, B., Gargouri, F.: How to model a real-time database? In: Proceedings of the IEEE International Symposium on Object Component Service-Oriented Real-Time Distributed Computing. pp. 321–325. IEEE, Tokyo, Japan (2009)
25. Joseane, O.V.P., Rossana, M.C.A., Rainara, M.C.: Evaluation of non-functional requirements for iot applications. In: Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS. pp. 111–119. SCITEPRESS, Virtual event (2021)
26. Kirchhof, J.C., Michael, J., Rumpe, B., Varga, S., Wortmann, A.: Model-driven digital twin construction: synthesizing the integration of cyber-physical systems with their information systems. In: Proceedings of the 23rd International Conference on Model Driven Engineering Languages and Systems. pp. 90–101. ACM, Virtual event (2020)
27. Lima, C., Mello, R.S.: A workload-driven logical design approach for nosql document databases. In: Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services. pp. 73:1–73:10. ACM, Brussels, Belgium (2015)
28. Ma, Z., He, X., Liu, C.: Assessing the quality of metamodels. Frontiers of Computer Science 7(4), 558–570 (2013)
29. Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I.A.T., Siddiqa, A., Yaqoob, I.: Big IoT data analytics: Architecture, opportunities, and open research challenges. IEEE Access 5, 5247–5261 (2017)
30. Mehrdad, S., Cicchetti, A., Sjödin, M.: Uml-based modeling of non-functional requirements in telecommunication systems. In: Proceedings of the Sixth International Conference on Software Engineering Advances (ICSEA). pp. 213–220. The Institute of Electrical and Electronics Engineers, Barcelona, Spain (2011)
31. Melouk, M., Rhazali, Y., Hadi, Y.: An approach for transforming CIM to PIM up to PSM in MDA. In: Proceedings of the The 11th International Conference on Ambient Systems, Networks and Technologies. pp. 869–874. Elsevier, Warsaw, Poland (2020)
32. Muller, R.J.: Database design for smarties: using UML for data modeling. Morgan Kaufmann (1999)
33. Nathan Marz, J.W.: Big Data:Principles and best practices of scalable realtime data systems. Manning (2015)
34. Nepomuceno, T., Carneiro, T., Maia, P.H., Adnan, M., Nepomuceno, T., Martin, A.: Autoiot: a framework based on user-driven mde for generating iot applications. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC). pp. 719–728. ACM, Brno, Czech Republic (2020)
35. Noa, R., Shoval, p., Sturm, A.: A method for database model selection. In: Proceedings of the 20th International Conference, BPMDS Enterprise, Business-Process and Information Systems Modeling. pp. 261–275. Springer, Rome, Italy (2019)
36. Noa, R., Sturm, A.: Design methods for the new database era: a systematic literature review. Springer, Software Systems Modeling 19(2), 297–312 (2020)
37. Novacek, J., Kühlwein, A., Reiter, S., Viehl, A., Bringmann, O., Rosenstiel, W.: Lemons: Leveraging model-based techniques to enable non-intrusive semantic enrichment in wireless sensor networks. In: Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). pp. 561–568. IEEE, Portoroz, Slovenia (2020)
38. Omoniwa, B., Hussain, R., Javed, M.A., Bouk, S.H., Malik, S.A.: Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues. IEEE Internet of Things Journal 6(3), 4118–4149 (2019)
39. Plazas, J.E., Bimonte, S., Corrales, M.S.J.C.: Self-service business intelligence over on-demand iot data: A new design methodology based on rapid prototyping. In: Proceedings of the New Trends in Databases and Information Systems (ADBIS). pp. 84–93. Springer, Lyon, France (2020)

40. Plazas, J.E., Bimonte, S., Schneider, M., de Vaulx, C., Battistoni, P., Sebillo, M., Corrales, J.C.: Sense, transform & send for the internet of things (sts4iot): Uml profile for data-centric iot applications. Data & Knowledge Engineering 139, 101971 (2022)
41. Plazas, J.E., Bimonte, S., de Vaulx, C., Schneider, M., Nguyen, Q., Chanet, J., Shi, H., Hou, K.M., Corrales, J.C.: A conceptual data model and its automatic implementation for iot-based business intelligence applications. IEEE Internet Things Journal 7(10), 10719–10732 (2020)
42. Popescu, A.D., Ercegovac, V., Balmin, A., Branco, M., Ailamaki, A.: Same queries, different data: Can we predict runtime performance? In: Proceedings of the ICDE Workshops. pp. 275–280. IEEE Computer Society, Arlington, VA, USA (2012)
43. Robinson, S., Arbez, G., Birta, L.G., Tolk, A., Wagner, G.: Conceptual modeling: definition, purpose and benefits. In: Proceedings of the Winter Simulation Conference. pp. 2812–2826. IEEE/ACM, Huntington Beach, CA, USA (2015)
44. Russom, P.: Data lakes: Purposes, practices, patterns, and platforms (2017), TDWI white paper
45. Sachdeva, V., Chung, L.: Handling non-functional requirements for big data and iot projects in scrum. In: Proceedings of the 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence. pp. 216–221. IEEE, Noida, India (2017)
46. Sapp, C.: Hyperscaling streaming analytics: Comparing stream analytics in the cloud with Amazon, IBM and Microsoft (2016), Gartner
47. Shubham, N.M., Keertikumar, B.M., Banakar, R.M.: Non functional requirement analysis in iot based smart traffic management system. In: Proceedings of the IEEE International Conference on Computing Communication Control and automation. IEEE, Pune, India (2016)
48. Souza, A.: Lambda architecture — how to build a big data pipeline (2019), https://dzone.com/articles/lambda-architecture-how-to-build-a-big-data-pipeli, dZone
49. Sørensen, C.G., Bochtis, D.: Conceptual model of fleet management in agriculture. Biosystems Engineering 105(1), 41–50 (2010)
50. Trujillo, J., Luján-Mora, S.: A UML based approach for modeling ETL processes in data warehouses. In: Proceedings of the Int. Conf. on Conceptual Modeling (ER). pp. 307–320. Springer, Chicago, IL, USA (2003)
51. Umar, M., Muhammad Naeem, A.K.: A framework to separate non-functional requirements for system maintainability. Kuwait Journal of Science Engineering (39), 211–231 (2012)
52. Vaisman, A.A., Zimányi, E.: Data Warehouse Systems - Design and Implementation. Springer (2014)
53. Wanner, J., Wissuchek, C., Janiesch, C.: Machine learning and complex event processing. A review of real-time data analytics for the industrial internet of things. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. 15, 1:1–1:27 (2020)
54. Zaharia, M., Ghodsi, A., Xin, R., Armbrust, M.: Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. In: Proceedings of the Conf. on Innovative Data Systems Research, CIDR. www.cidrdb.org, Virtual event (2021)
55. Zimányi, E., Minout, M.: Implementing conceptual spatio-temporal schemas in object-relational dbmss. In: Proceedings of the On the Move to Meaningful Internet Systems: OTM Workshops. pp. 1648–1657. Springer, Montpellier, France (2006)
56. Zimanyi, E., Parent, C., Spaccapietra, S., Vangenot, C.: Multiple representation modeling. In: Liu, L., Ozsu, M.T. (eds.) Encyclopedia of Database Systems, 2nd Edition, p. 2419–2425. Springer (2018)

Sandro Bimonte is Research Director at French National Research Institute for Agriculture, Food and the Environment (France), and more exactly he is Member of the TSCF Laboratory. He received his PhD from INSA-Lyon, France (2004–2007). He is an Editorial Review Board Member of International Journal of Data Warehousing and Mining,

International Journal of Decision Support System Technology, and international conferences such as ER, DOLAP, etc. He has published more than 100 papers in refereed journals and international conferences. His research activities concern spatial data warehouses and spatial OLAP, visual languages, geographic information systems, spatio-temporal databases, geovisualisation, Big Data, and IoT. He joined and coordinated several research projects (such as VGI4bio.fr and BEYOND) on the above areas.

Hassan Badir is a full Professor in the Department of Computer Science and Engineering. He received a Ph.D. in Computer Science from the INSA-Lyon and Claude Bernard University (France) in 2005. He is actually: Head of Data engineering System TEAM SDET, Head of Bioinformatics and Data Science Master (BISD), Head of High Specialized Master study on Compute Engineering, Security and Decision, Co-Investigator of Human Heredity and Health African Bioinformatics Network H3ABionet, Head and founding member of the Moroccan Innovation and New trends of Information System Society. His research interests are in the areas of big data management, sensor and scientific data management, cloud computing and security, data management systems and distributed systems. I am also serving as an Associate Editor and Track Editor of International Journal of Database Management Systems(IJDBMS), Cluster Computing – The Journal of Networks, Software Tools and Applications (CLUSTE COMPUTING), International Society for Computers and Their Applications ISCA, Journal Concurrency and Computation: Practice and Experience. I am co-organiser of several workshops, programme chair of international conferences EGCM, ASD, EDA and INTIS. I am also serving as the scientific committee member of many reputed international conferences, AICSSA, MEDES, SITIS, EDA, ASD, EGC, INTIS, ASONAM, MedICT, SFC, DSC, ICCCI, BDCA, ICICS, JERI, BioMining, ICWIT, ICT-DM.

Pietro Battistoni is ending his PhD in Computer Science this year, 2021. He acquired an in depth understanding of software and hardware development project lifecycles by over 30 years of field research professional work experience in industrial automation. Twenty two years as owner of a privately held small company grant him experience in project management and technical skills in microcomputer-controlled system design, 3D CAD design, and electronic-CAD to develop hardware and firmware of electronic devices. Additional certified knowledge is in Cybersecurity, Cybersecurity Risk Management, Network Security, and Computer Forensics. His latest research interests are the IoT, distributed computing architectures and edge computing, geo-visual analytics, machine learning, human–computer interaction, robot programming, autonomous vehicles, and territorial intelligence, accomplished with 18 peer-review articles published in the lasts 2 years.

Houssam Bazza is a PhD student in computer science in Abdelmalek Essaadi University (Tangier-Morocco) and Membre of IDS Team, he held a master's degree in bioinformatics and data science from the national school of applied science morocco, his current research topics include: conceptual modeling, big data, business intelligence, stream data management and the internet of things.

Amina Belhassena is a Postdoctoral Researcher at Universite´ Clermont Auvergne, TSCF, INRAE, France. She received the PhD Degree in Computer Science from Harbin Institute

of Technology, China in 2018. She received the Master Degree of Technology in Computer Science from Abou Bakrbelkaid Tlemcen University, Algeria in 2012. Her research interest includes massive data computing, big data management, big streaming data management, data indexing and data mining.

Christophe Cariou is Research Engineer at INRAE, the French National Research Institute for Agriculture, Food and Environment. He received his Electrical Engineer Diploma in 1994, his Postgraduate Diploma in Electronics and Systems in 1995 and his PhD in 2012 on the control of poly-articulated robots. His research interests include the development of nonlinear, adaptive and predictive control for autonomous off-road mobile robots, as well as the optimal trajectory planning.

Gerard Chalhoub is an Associate Professor at University of Clermont Auvergne, France. He obtained his PhD Degree in Computer Science in 2009 at University of Blaise Pascal, France. He obtained his Habilitation Degree in Networking and Telecommunications in 2016 at University of Auvergne, France. His main research topics are reliability, quality of service and security in wireless networks. He has more than 70 publications in these domains.

Juan Carlos Corrales received the Dipl-Ing and master's degrees in telematics engineering from the University of Cauca, Colombia, in 1999 and 2004 respectively, and the Ph.D. degree in sciences, speciality computer science, from the University of Versailles Saint-Quentin-en-Yve-lines, France, in 2008. At present, he is a full professor and leads the Telematics Engineering Group (GIT) at the University of Cauca. His research interests focus on machine learning and data analytics.

Adrian Couvent is a Young Engineer and PhD Student in Robotics Applied to Agriculture. He is involved in robotics in several aspects (from control laws to user experience). His thesis on taking into account the competence of an operator in adjusting the degree of autonomy of a technical system will be defended in early 2022.

Jean Laneurit obtained his PhD at Institute Pascal, France in 2006. He is currently member of TSCF, INRAE, Clermont Ferrand. He is Research Engineering in Computer Science Applied to Robotics.

Rim Moussa is an Associate Professor at University of Carthage. She received her MSc and PhD in Distributed Databases from Universite´ Paris IX Dauphine (France). Her current research interests include scalable and distributed data management systems, Big Data architectures and spatial computing at scale.

Julian Eduardo Plazas is a Ph.D. in Telematics Engineering and Computer Science in Universidad del Cauca (Colombia) and Université Clermont Auvergne (France). He has worked in the definition of rural Early Warning Systems through Converged Services and Complex Event Processing for agricultural environments, in the implementation of Machine-Learning Classifiers for agricultural Decision Support Systems, and in Conceptual Modelling for agricultural Wireless Sensor Networks. His current research topics include the Internet of Things, Business Intelligence, Data Analytics and Conceptual Modelling for Intelligent Agriculture Systems.

Monica Sebillo received a Laurea Degree in Scienze dell'Informazione from the University of Salerno and a PhD in Applied Mathematics and Computer Science from the University of Naples. She is an Associate Professor in Computer Science at the Department of Computer Science (DI) at the University of Salerno. Monica is an ACM Senior Member. Her research interests include geographic information systems, GeoAI and geospatial databases.

Nicolas Tricot is currently Research Fellow at INRAE (National Research Institute for Agriculture, Food and the Environment) in the Research Unit TSCF (Technologies and Information Systems) at Clermont-Ferrand (France). He received an Engineer Degree and a Master Degree in Automation in 2001. He defended is PhD in 2005 on the Topic of Design and Evaluation of Advanced Driving Assistance Systems. He joined Irstea in 2006. In a first time, he worked on the integration of human factors in system design. In a second time, he joined the research team Romea (Robotic and Mobility for Environment and Agriculture) in ClermontFerrand to work on human and agricultural robot interactions.

Received: March 01, 2022; Accepted: August 22, 2022.

Computer Science and Information Systems 20(1):491–512 https://doi.org/10.2298/CSIS220131065F

Optimizing Data Locality by Executor Allocation in Spark Computing Environment*

Zhongming Fu[1], Mengsi He[1]**, Zhuo Tang[2], and Yang Zhang[3]

[1] Computer School, University of South China, and Hunan Provincial Base for Scientific and Technological Innovation Cooperation
Hengyang, Hunan, China, 421001
fuzhongming@hnu.edu.cn
mengsih@163.com

[2] College of Information Science and Engineering, Hunan University, and National Supercomputing Center
Changsha, Hunan, China, 410082
ztang@hnu.edu.cn

[3] Science and Technology on Parallel and Distributed Laboratory (PDL), National University of Defense Technology
Changsha, Hunan, China, 410073
yangzhang15@nudt.edu.cn

Abstract. Data locality is an important concept in big data processing. Most of the existing research optimized data locality from the aspect of task scheduling. However, as the execution container of tasks, the executors started on which nodes can directly affect the locality level achieved by the tasks. This paper tries to improve the data locality by executor allocation for reduce stage in Spark computing environment. Firstly, we calculate the network distance matrix of executors and formulate an optimal executor allocation problem to minimize the total communication distance. Then, when the network distance between executors satisfies the triangular inequality, an approximate algorithm is proposed; and when the network distance between executors does not satisfy the triangular inequality, a greedy algorithm is proposed. Finally, we evaluate the performance of our algorithms in a practical Spark cluster by using several representative micro-benchmarks (Sort and Join) and macro-benchmarks (PageRank and LDA). Experimental results show that the proposed algorithms can decrease the execution time of tasks for lower data communication.

Keywords: communication distance, data locality, executor allocation, spark framework.

1. Introduction

With the increasingly high response requirement of applications in the era of big data, Spark [1] attracts great attention in academia and has become the popular parallel com-

* This is an extension of work presented in the conference paper: Fu, Z., He, M., Tang, Z., Zhang, Y.: Optimizing Data Locality by Executor Allocation in Reduce Stage for Spark Framework. In: Parallel and Distributed Computing, Applications and Technologies, PDCAT 2021, Lecture Notes in Computer Science, Springer, Cham, Vol 13148. DOI: https://doi.org/10.1007/978-3-030-96772-7_32.

** Corresponding author

puting framework for massive data processing [28]. The core abstraction of Spark is resilient distributed dataset (RDD) that can be cached to memory, thus avoiding writing and reading data in HDFS between jobs. Therefore, compared with Hadoop [2], Spark can take advantage of in-memory computing to perform jobs more efficiently.

A typical Spark application contains one or more jobs, and a job usually consists of many stages. Since the stages are executed sequentially, the intermediate output of the former stage is used as the input of the later stage. When the tasks of a stage run in parallel on different nodes, the data communication across the network occurs [22]. As shown in Fig. 1, in the map stage (i.e., the first stage), each task reads a data block to process and outputs the intermediate data to local disks. In the reduce stage (i.e., the subsequent stages), each task fetches part of the intermediate data from the previous stage for processing. This is the so-called shuffle that is a many-to-many communication. The resulting large amount of network traffic in the two stages can congest the network and extend execution time, thereby hindering the system [23].

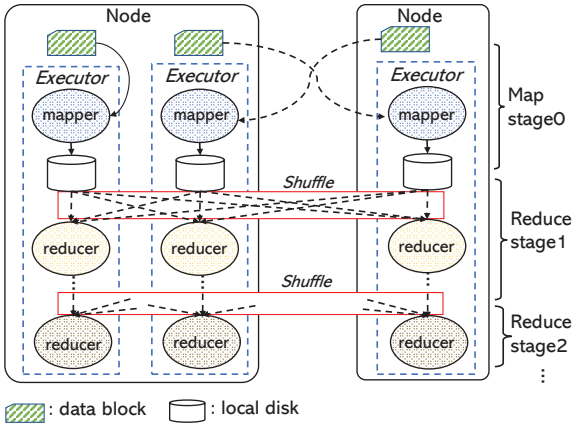


Fig. 1. Data communication between Spark stages.

For improving performance, data locality is a key factor considered by the task scheduling of Spark stages [10]. The task scheduling determines the executor on which node the task runs and the data locality refers to scheduling task close to data, so that the communication overload can be reduced [19], [15]. In particular, in the map stage, the *taskScheduler* uses the delay scheduling algorithm [34] that tries to assign the map task to the node which stores the data block, and in the reduce stage, the *taskScheduler* assigns the reduce task to one of the nodes that holds more intermediate data to the task, thus to minimize the data transfer volume.

In MapReduce frameworks (e.g. Hadoop), most of the existing research focused on optimizing data locality from the aspect of task scheduling [17], [21], [9]. In particular, Guo et al. [16] assign map task to maximize the number of node-local tasks, and Shang et al. [25] dispatch reduce task to minimize the network resources consumption. These task scheduling methods are the direct way to improve the data locality and achieves better performance, but they do not involve the problem of executor allocation in Spark.

As the execution container of tasks, the executors can restrict the nodes available for the task scheduling. This actually affects the locality level achieved by the tasks. On the one hand, if the executor is not started on the nodes in which the data block is located in the map stage, the map task is almost impossible to obtain data locally. On the other hand, if the executors are started on the nodes away from each other in the reduce stage, the reducer has to go through a long network distance to get data. Spark provides two algorithms: *spreadOut* and *noSpreadOut* to decide the executors start up. Unfortunately, neither of them fully exploit the benefits of data locality.

In this paper, we improve the data locality from the view of executor allocation considering the reduce stage in Spark computing environment. In general, the number of reduce stage is much greater than that of map stage, so the reduce stage has an important impact on the whole performance. Compared with the conference version [12], which proposed an approximate algorithm for the case that the network distance between executors satisfies the triangular inequality, this paper also focuses on the executor allocation for the case that the network distance between executors does not satisfy the triangular inequality. The main contributions of this paper are summarized as below.

- We calculate the distance matrix of executors, and formulate an executor allocation problem to minimize the total communication distance. This problem is proved to be an NP-Hard problem.
- When the network distance between executors satisfies the triangular inequality, we propose an approximate algorithm and prove the approximate factor is 2.
- When the network distance between executors does not satisfy the triangular inequality, we propose a greedy algorithm and prove the correctness of the algorithm.
- We implement our algorithms in Spark-3.0.1 and evaluate their performance on representative benchmarks. The experiment results explain that the proposed algorithms can reduce the task execution time for better data locality.

The rest of this paper is organized as follows. Section 2 reviews more related research. Section 3 describes the motivation of our optimization. Section 4 presents the proposed executor allocation algorithms. Experiments and performance evaluation are given in Section 5. Section 6 concludes this paper.

2. Related Work

Many research has been done to optimize the data locality in MapReduce frameworks, which can be categorized as follows:

Task scheduling. In the design of MapReduce, Dean et al. [11] took the locality of map tasks into account to save bandwidth consumption. The priority of tasks scheduled to nodes is classified into three levels: *node-local*, i.e., the task and its data block are on the same node; *rack-local*, i.e., the task and its data block are on different nodes but on the same rack; and *off-rack*, i.e., the task and its data block are on different racks but on a cluster.

Further, using the time-for-space strategy, Zaharia et al. [34] proposed the delay scheduling algorithm. If there is no map task can obtain data locally on the request node, a small amount of time is waited in the hope of obtaining better locality from subsequent nodes. In a cluster that quickly releases resources, the delay scheduling can achieve a higher proportion of node-local tasks while preserving fairness.

Besides the map stage, the data locality for reduce tasks also affects the performance [27], [8], [25]. Tang et al. [31] presented a minimum transmission cost reduce task scheduler (MTCRS). It decides the appropriate launching locations for reduce tasks according to two factors: the waiting time of each reducer and the transmission cost set, which is computed by the sizes and the locations of intermediate data partitions.

In order to alleviate data skew at the same time, Tang et al. [29] provided a reduce placement algorithm CORP. It first uses a reservoir algorithm for sampling the input data to estimate the distribution of keys/values, then on the basis of this, it calculates the distance and cost matrices among the cross node communication. Finally, the related map and reduce tasks are scheduled relatively nearby physical nodes.

Data pre-fetching. From another angle, Sun et al. [26] designed HPSO (High Performance Scheduling Optimizer), a prefetching service based task scheduler to improve data locality for MapReduce jobs. Their idea is to predict the most appropriate nodes to which future map tasks should be assigned and then pre-load the input data to memory without any delaying on running normal tasks.

In [35], Zhang et al. proposed a pre-fetching method based on pre-scheduling in Hadoop systems. The method hides the remote data access delay by pre-fetching, and controls the resource competition by adjusting resource allocation of reduce tasks. Nevertheless, the above pre-fetching techniques may incur additional overhead and could not help to alleviate the network traffic of cluster.

High speed network. In addition, some researchers were dedicated to finding high speed network to speed up the data transmission. Lu et al. [18] proposed a novel design (RPCoIB) of Hadoop RPC with RDMA over InfiniBand networks. RPCoIB provides a JVM-bypassed buffer management scheme and utilizes message size locality to avoid multiple memory allocations and copies in data serialization and deserialization.

In [32], Yan et al. introduced R3S, an RDMA-based in-memory RDD storage layer for Spark. R3S leverages high bandwidth networks and low-latency one-sided RDMA operations to allow Spark nodes to efficiently access intermediate output from a remote node.

In the above studies, these task scheduling methods are the direct way to improve the data locality, but they do not involve the problem of executor allocation in Spark computing environment. Therefore, the executors can restrict the nodes available for the task scheduling. In our early work [13], we proposed an optimal task scheduling algorithm and an executor allocation algorithm to optimize the data locality in the map stage. While in this paper, we focus on the executor allocation in the reduce stage, with the purpose of providing the possibility of better locality level when scheduling the reduce tasks.

3. Motivation

There are two methods: *spreadOut* and *noSpreadOut* provided by Spark to decide on which nodes the executors start. The idea of the *spreadOut* strategy tries to launch the required executors on as many nodes as possible, while *noSpreadOut* goes a inverse way, it launches the executors on as few nodes as far as possible. We show how the executor allocation affects the data locality in reduce task scheduling.

As shown in Fig. 2, suppose that a cluster has 4 idle nodes namely node0, node1, node2, and node3, and the number of executors allowed to start on each node is 3, 2, 2, and

1 respectively, as shown in Fig. 2(a). The number of executors required by the application is 5. For these nodes that are sorted according to the number of executors allowed to start, *spreadOut* takes turns to start the executor on each node until reaching the number requirement, as shown in Fig. 2(b). In contrast, *noSpreadOut* starts the allowed number of executors on a node in turn until the number requirement is met, as shown in Fig. 2(c). For simplicity, we use the hop count to calculated the network distance, so the sum of the communication distance between executors under *spreadOut* and *noSpreadOut* is 21 and 18 respectively. However, given an optimal executor allocation strategy shown by Fig. 2(d), the minimum total communication distance is 8. Then when scheduling the reduce tasks to the nodes and run in the executors, the tasks need to go through a longer network distance to get data under *spreadOut* and *noSpreadOut* than under the optimal strategy.

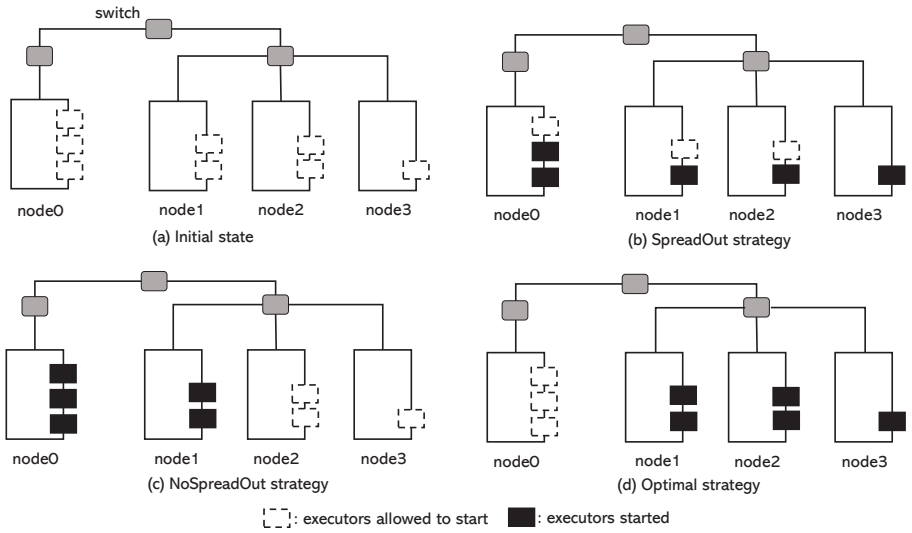


Fig. 2. Different executor allocation strategies.

From the above analysis, we can conclude that *spreadOut* and *noSpreadOut* may cause the executors to start on the nodes away from each other, bringing a great communication overhead when the reduce tasks fetch the intermediate data. Therefore, the poor data locality could be obtained in the task scheduling. It also should be noted that if there is little intermediate data generated in the previous stage, it is difficult to improve job performance by improving the data locality in the reduce stage.

4. Proposed Executor Allocation Algorithm

This section formulates the optimal executor allocation problem, and then presents an approximate algorithm and a greedy algorithm respectively.

4.1. Optimal Executor Allocation Problem

When a Spark application is submitted to the cluster and to be executed, the *master* registers with the resource manager (such as YARN [30]) and applies for the resources to start a group of executors. An executor is the container of running tasks, which is a collection of computing resources (i.e., cpu and memory). A task can be scheduled to run on a node requiring to have idle executors.

For illustrative purposes, some important variables involved in the model are declared in Table 1.

Table 1. Variable Declaration

Variable	Declaration
$N_l, 0 \leq l < \alpha$	The l^{th} node of the cluster
$R_r, 0 \leq r < \beta$	The r^{th} rack of the cluster
$e_i^l, 0 \leq i < m$	The i^{th} executor allowed to start, which is located on the l^{th} node
$d_{ij}, 0 \leq j < m$	The network distance between executor e_i and e_j

We first initialize the network topology of the cluster as the node set $\{N_0, N_1, \cdots, N_{\alpha-1}\}$ and rack set $\{R_0, R_1, \cdots, R_{\beta-1}\}$, $1 \leq \beta \leq \alpha$, where α is the number of nodes and β is the number of racks. In the initial state of allocating executor for an application, some particular data structures are defined as follows:

(1) E: A set of executors allowed to be started on the nodes, the number is m. The element e_i^l represents the i^{th} executor that can be started on the l^{th} node if marked. In the Spark framework, the number of executors allowed to start on each node can be calculated based on the free resources of the node, formalized as:

$$exe\_num_i = \min\{[\frac{free\_cpu_i}{cpu\_conf}], [\frac{free\_memory_i}{memory\_conf}]\}, \tag{1}$$

where exe_num_i indicates the number of executors allowed to start on node N_i, and cpu_conf and $memory\_conf$ are the number of CPUs and memory capacity configured by the executor respectively.

(2) D: A matrix of $m \times m$ represents the network distance between executors of E, represented as:

$$D = \begin{bmatrix} d_{00} & d_{01} & \cdots & d_{0(m-1)} \\ d_{10} & d_{11} & \cdots & d_{1(m-1)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{(m-1)0} & d_{(m-1)1} & \cdots & d_{(m-1)(m-1)} \end{bmatrix},$$

where d_{ij} represents the network distance between executor e_i and e_j. It is noted that d_{ij} is set to 0 when $i = j$.

With the aim of capturing the data locality, we divide the proximity level (PL) of two executors into three levels: (1) if two executors are on the same node, then PL is equal to

0; (2) if two executors are on different nodes of the same rack, then PL is equal to 1; (3) if two executors are on different nodes of different racks, then PL is equal to 2. Then the network distance d_{ij} of D can be specifically calculated as:

$$d_{ij} = \begin{cases} 0, if PL = 0 \\ 2 \times \left(\frac{1}{band_{NS}} + latency_{NS}\right), if PL = 1 \\ 2 \times \left(\frac{1}{band_{NS}} + latency_{NS}\right) + 2 \times \left(\frac{1}{band_{SS}} + latency_{SS}\right), if PL = 2 \end{cases}, \quad (2)$$

where $band_{NS}$ is the network bandwidth from node to switch, $band_{SS}$ is the network bandwidth from switch to switch, $latency_{NS}$ is the network delay from node to switch, and $latency_{SS}$ is the network delay from switch to switch.

In this model, our purpose is to start the required executors on nodes close to each other. Assuming that the number of executors required by the application is k, so the optimal executor allocation problem can be described as selecting a subset $E^{'} \in E$ to minimize the total communication distance between executors. This problem can be formalized as follows by using *Integer Programming*:

$$\begin{aligned} & min \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} d_{ij} \times (x_i \times x_j), \\ subject\ \ to\ \ & \sum_{i=0}^{m-1} x_i = k, x_i \in \{0,1\}, 0 \leq i < m-1, \end{aligned} \quad (3)$$

where x_i is a binary variable, whose value is 1 means that the executor e_i is selected, and value is 0 means that the executor is not selected.

Theorem 1. *The optimal executor allocation problem (abbreviated as the OEA problem) is NP-Hard.*

Proof. The k-clique problem in graph theory can be educible to the *OEA* problem. That is, for any instance of the k-clique, an instance of *OEA* can be created in polynomial time such that solving the instance of *OEA* solves the instance of k-clique as well. According to the NP completeness of the k-clique problem, the *OEA* problem can be proved to be NP-Hard [14].

Definition 1. *k-clique problem: Given a graph $G(V,U)$ and an integer k, determine whether G has a clique of size k.*

Let $P = (V,U,k)$ be an instance of k-clique, where $V = \{0,1,...,m-1\}$ is the vertex set, $U = \{u_{ij} | i,j \in V\}$ is the edge set, and k is a positive integer.

Then give an instance of *OEA*: $Q = (E,D,k)$, where E is the set of executors allowed to start, D is the distance matrix between executors, and k is a integer. For the executors e_i and e_j of E corresponding to i and j of V, the element d_{ij} of D is assigned to:

$$d_{ij} = \begin{cases} 0, & if\ u_{ij} \in U \\ 1, & otherwised' \end{cases} \quad (4)$$

We claim that a clique of size k exists if and only if E contains a subset $E^{'}$ such that the number of elements in the $E^{'}$ is equal to k, and $\sum_{e_i,e_j \in E'} d_{ij} = 0$.

Figure 3 shows an instance of k-clique, which contains the vertices 0, 1, 2, 3, 4, 5. On this basic, an instance of *OEA* can be created and its network distance matrix D is shown in Table 2.

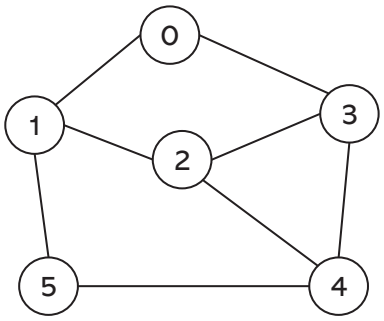


Fig. 3. An instance of k-clique.

Table 2. Network distance matrix

Executor	e_0	e_1	e_2	e_3	e_4	e_5
e_0	0	0	1	0	1	1
e_1	0	0	0	1	1	0
e_2	1	0	0	0	0	1
e_3	0	1	0	0	0	1
e_4	1	1	0	0	0	0
e_5	1	0	1	1	0	0

It can be seen from the above example that the instance of k-clique has a clique of size 3. In the meantime, *OEA* has a solution $E^{'} = \{2, 3, 4\}$ such that k=3, and $\sum_{e_i, e_j \in E'} d_{ij} = 0$.

Hence, the existence of k-clique is a necessary and sufficient condition for the *OEA* problem to be solved, with the size of k and the minimum total communication distance. Hence the proof.

4.2. Approximate Algorithm

We first consider that the network distance between two executors satisfies the triangular inequality [24], such as the homogeneous network of nodes, that is, the distance between any two executors satisfies the following relationship: $d_{ij} \leq d_{iv} + d_{vj}$.

Algorithm 1 describes the approximate algorithm for the optimal executor allocation problem. Firstly, the algorithm selects k nearest executors (including e_i itself) for each executor e_i. For executor e_i, the set of its k nearest executors is represented as $S(e_i)$, and

the sum of network distance from executor e_i to other $k-1$ executors is calculated and represented as $C(e_i)$. Then, find the smallest $C(e_v)$ among all executors, and assign the executor set $S(e_v)$ to $MinSet$. Finally, return to $MinSet$.

Algorithm 1: Approximate Algorithm

Input:
The set of executors allowed to start: E;
The network distance matrix: D;
The number of executors required: k;
Output:
The executors selected to start.

```
begin
    for each executor e_i ∈ E do
        find the set of its k nearest executors: S(e_i);
        calculate the sum of network distance from executor e_i to other k − 1 executors:
        C(e_i) = Σ_{e_j ∈ S(e_i)} d_ij;
    end
    find the smallest C(e_v) and the executor set: MinSet;
    return MinSet.
end
```

The algorithm takes $O(m)$ time to find the nearest k executors by using the optimal algorithm. For m executors, the time it takes is $m \times O(m)$. Therefore, the time complexity of Algorithm 1 is $O(m^2)$, where m is the number of executors allowed to start.

Theorem 2. *The approximate factor of the approximate algorithm to the optimal executor allocation problem is 2.*

Proof. The solution of the approximate algorithm for the optimal executor allocation is $MinSet$, so the total communication distance between executors of $MinSet$ can be represented as $MinCost$. Let $MinSet^*$ be the optimal solution, and the total communication distance between executors of $MinSet^*$ is $MinCost^*$. Then for $MinCost^*$, there is:

$$\begin{aligned} MinCost^* &= \frac{1}{2} \sum_{e_i \in MinSet^*} \sum_{e_j \in MinSet^*} d_{ij} \geq \frac{1}{2} \sum_{e_i \in MinSet^*} \sum_{e_j \in S(e_i)} d_{ij} \\ &= \frac{1}{2} \sum_{e_i \in MinSet^*} C(e_i) \geq \frac{1}{2} \sum_{e_i \in MinSet^*} MinCost = \frac{k}{2} \times MinCost. \end{aligned} \tag{5}$$

For $MinCost$, there is:

$$MinCost = \frac{1}{2} \sum_{e_i \in MinSet} \sum_{e_j \in MinSet} d_{ij}. \tag{6}$$

Let C_{e_v} gets the minimum total communication distance $MinCost$. According to the triangular inequality, there is:

$$\sum_{e_i \in MinSet} \sum_{e_j \in MinSet} d_{ij} \leq \sum_{e_i \in MinSet} \sum_{e_j \in MinSet} (d_{iv} + d_{vj})$$
$$= \sum_{e_i \in MinSet} \sum_{e_j \in MinSet} d_{iv} + \sum_{e_i \in MinSet} \sum_{e_j \in MinSet} d_{vj}$$
$$= \sum_{e_j \in MinSet} \left(\sum_{e_i \in MinSet} d_{vi} \right) + \sum_{e_i \in MinSet} \left(\sum_{e_j \in MinSet} d_{jv} \right)$$
$$= k \times \left(\sum_{e_i \in MinSet} d_{iv} \right) + k \times \left(\sum_{e_j \in MinSet} d_{jv} \right)$$
$$= k \times MinCost + k \times MinCost. \tag{7}$$

Therefore, for $MinCost$, there is:

$$\frac{1}{2} \times 2k \times MinCost = k \times MinCost. \tag{8}$$

According to equations (5), (6), (7), and (8), the approximate factor of our solution $MinSet$ is calculated as:

$$\sigma = \frac{MinCost}{MinCost^*} \leq \frac{k \times MinCost}{\frac{k}{2} \times MinCost} = 2. \tag{9}$$

Therefore, the approximate algorithm for the optimal executor allocation problem is a 2-approximate algorithm.

4.3. Greedy Algorithm

When the triangular inequality cannot be guaranteed in a data center, such as the heterogeneous network of nodes, we propose a greedy algorithm for the optimal executor allocation problem.

Algorithm 2 uses the distance *threshold* to select the executors. To minimize the total communication distance, firstly, the algorithm calculates the maximum and minimum distance between executors of E, and assigns the *threshold* to the minimum distance. Secondly, it finds all executor pairs in E whose network distance is equal to *threshold*, and puts them in $E^{'}$. Thirdly, the algorithm expands the executor set $E^{'}$ by searching executor $e_v \in U$ that satisfies the distance between e_v and any executor of $E^{'}$ is not greater than *threshold*. This process is repeated until e_v does not exist or the number of executors of $E^{'}$ equals k. Finally, if the number of executors of $E^{'}$ equals k, return $E^{'}$; Otherwise, increases *threshold* and cycles the above steps.

Because $k \leq m$, as long as the $threshold$ is set reasonably, it can always return an executor set of size k. The time complexity of Algorithm 2 is $O(k \times m^3)$, where k is the number of executors required.

The correctness of Algorithm 2 is proved as follows:

Algorithm 2: Greedy Algorithm

Input:
The set of executors allowed to start: E;
The network distance matrix: D;
The number of executors required: k;
Output:
The executors selected to start.

begin
  calculate the maximum and minimum distance between executors of E: $Max(D)$ and $Min(D)$;
  $Threshold = Min(D)$;
  while $Threshold \leq Max(D)$; **do**
    initialize $U = E$; $E^{'} = \varnothing$;
    repeat
      find executor pair $e_u, e_v \in E$, such that $d_{uv} == Threshold$;
      $E^{'} = E^{'} \cup \{e_u, e_v\}$;
      $U = U - \{e_u, e_v\}$;
    until e_u, e_v does not exist;
    repeat
      find $e_v \in U$ such that for each $e_u \in E^{'}$: $d_{uv} \leq Threshold$;
      $E^{'} = E^{'} \cup e_v$;
      $U = U - e_v$;
    until e_v does not exist or $|E^{'}| = k$;
    if $|E^{'}| = k$ **then**
      go to 21;
    end
    else
      $Threshold++$;
    end
  end
  return $E^{'}$.
end

Proof. Algorithm 2 returns an executor set $E^{'}$ with a $Threshold$. This means that no more executor set can be found so that the size is at least k and the threshold is less than $Threshold$. Assume that the executor set E^{*} has a $Threshold^{*}$ such that $\sum_{u,v \in E^{*}} d_{uv} < \sum_{u,v \in E^{'}} d_{uv}$ and $Threshold^{*} < Threshold$. For the executor set $E^{'}$, it starts from the minimum distance and gradually expands the executor set according to the selected executors. Because $Min(D) < Threshold^{*} < Threshold$, when $Threshold$ is $Min(D)$, the set cannot be expanded to a value size of k, Algorithm 2 will continue to increase the $threshold$ and expand the executor set; when the $threshold$ is $Threshold^{*}$, the executor set will be expanded to a size equal to k, and exit. At this time $Threshold = Threshold^{*}$, which contradicts the assumption.

5. Experimental Evaluation

In this section, we evaluate the performance of our proposed algorithms. The executor allocation strategies: *approximate algorithm* and *greedy algorithm* are implemented by modifying the function *scheduleExecutorsOnWorkers* of Spark-3.0.1 source codes. Hence the system can use our achievement when allocating the executors on idle nodes.

5.1. Experiment Setup

The experiments are carried out in a data center that contains 9 servers organized by 3 racks. These racks contain 2, 3, and 4 servers respectively, and each server has one Intel(R) Xeon(R) CPU E5-2678, 64GB RAM and 512GB Disk. The KVM virtualization technology is used to build medium-sized virtual machines, every VM is equipped with 4 virtual cores, 8GB RAM and 64GB disk space.

For software configuration, we use Java JDK-1.8, Apache Hadoop-2.7.3 and Apache Spark-3.0.1, and the deployment mode is YARN. Unless otherwise stated, all configurations are set by default. In particular, the block size of HDFS is set to 128MB and the replication factor is 3.

For the purpose of evaluating the performance, as shown in Table 3, two micro-benchmarks (join and sort) and two macro-benchmarks (pageRank and LDA (Latent Dirichlet Allocation)) are chosen for testing. These benchmarks are characterized by distinct workload types and representative in big data processing.

Table 3. Benchmark and workload type

Benchmark	Workload type
Sort	Map and Reduce-Input Heavy Job
Join	Reduce-Input Heavy Job
PageRank	Iterative Application
LDA	Iterative Application

We first compare the performance of *approximate algorithm* and *greedy algorithm* with *spreadOut* and *noSpreadOut* [3] provided by Spark, and then compare them with other recent executor allocation methods: *iSpark* [33] and *Warm-up Manager* [20]. *iSpark* aims to timely scale up or scale down the number of executors in order to fully utilize the allocated resources, and *Warm-up Manager* aims to reduce the initialization overhead and to enable latency-sensitive applications to apply dynamic strategies. For fairness, the following performance indicators are used:

Job execution time: the time from the start to the end of a job. Because the executor allocation can also affect the data locality in the map stage, thereby influencing the execution time of the job. Therefore, this indicator can comprehensively reflect the overall performance of relevant algorithms.

Reduce stage execution time: the time from the reduce task obtains intermediate data to the end of the task. Since there is a large amount of data communication in the reduce stage, the impact of different executor allocation algorithms on job performance can be directly observed through the execution time of the stage.

5.2. Performance

Satisfy the triangular inequality. We first deploy the Spark cluster on the data center with 18 nodes (each server starts 2 VMs). To provide the homogeneous network of nodes,

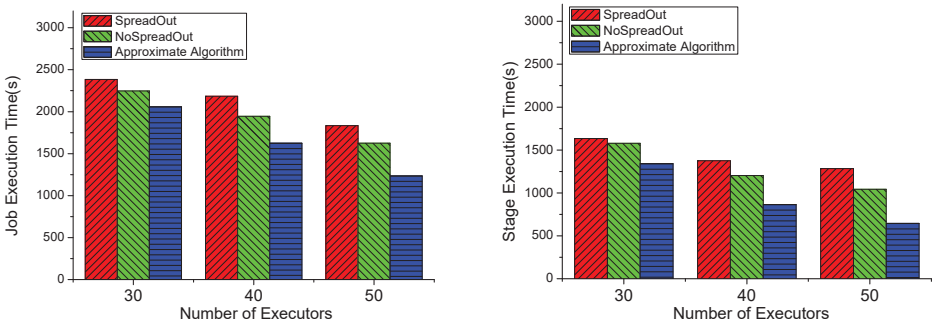
the bandwidth inside and outside the racks is set 10Gbps, so that the network distance matrix of executors satisfies the triangular inequality. Then we estimate the performance of *approximate algorithm*.

(1) Micro-benchmark

Sort is a popular application with the function of making data objects in order. The experiment uses 30GB data set of the *Wikipedia Corpus* [6]. This application contains a job with two stages: map stage and reduce stage, each stage has 240 tasks. To evaluate the performance under different numbers of executors, the number of executors required is set to 30, 40, and 50 respectively in the procedure.

Fig. 4(a) shows the performance comparison of the three executor allocation methods (*spreadOut*, *noSpreadOut* and *approximate algorithm*). It illustrates that *approximate algorithm* has a lower job execution time than other two methods. Meanwhile, as the number of executors increases, the job execution time is shorter due to the increase in the parallelism of tasks.

We further observe the performance comparison in the reduce stage, as shown in Fig. 4(b). In this stage, the reducer takes a lot of time to obtain the intermediate data from previous tasks. Since the reduce stage is considered in our optimization of data locality through executor allocation, it can be seen that *approximate algorithm* has a obvious reduction in the execution time. In particular, when the number of executors required is 40, comparing with *spreadOut* and *noSpreadOut*, *approximate algorithm* reduces the execution time by 37.1% and 28.2% respectively.



(a) Job execution time

(b) Reduce stage execution time

Fig. 4. Performance comparison under Sort.

Furthermore, Fig 5 displays the empirical CDF of these reduce tasks when the number of executors required is 40. We can see that the execution time of most tasks is between 100 and 200 seconds under *approximate algorithm*, which is better than the performance under *spreadOut* and *noSpreadOut*. Specifically, the average execution time of reduce tasks for *spreadOut*, *noSpreadOut*, and *approximate algorithm* is $229s$, $200s$, and $144s$, respectively.

To explore the reasons for performance improvement, we analyze the data locality of reduce tasks, which is divided into three levels: local access data, cross-node traffic, and cross-rack traffic. Table 4 shows the network traffic of reduce tasks during the stage

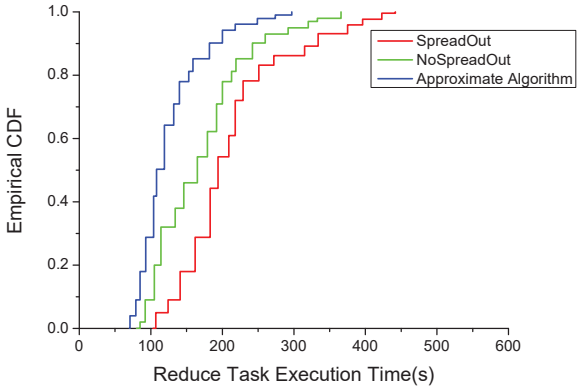


Fig. 5. The empirical CDF of reduce tasks.

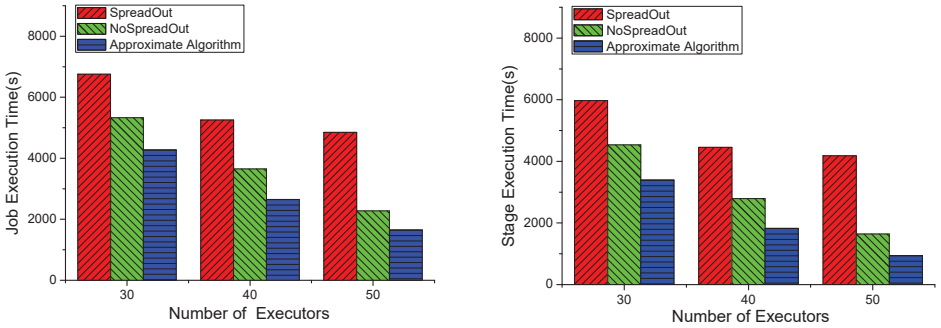
execution. In general, the locality level of *approximate algorithm* is much better than *spreadOut* and *noSpreadOut*, with more local access data and less cross-node/rack traffic. This is because *approximate algorithm* starts executors on the nodes close to each other, providing reduce tasks with the possibility of better locality in task scheduling. In contrast, *spreadOut* and *noSpreadOut* do not fully consider the data locality factor, leading to the reduce tasks to traverse longer network distances to obtain data, so the overhead of data communication is relatively high.

Table 4. Network Traffic of Reduce Tasks

Locality Level	spreadOut	noSpreadOut	approximate algorithm
Local access data	23.5%	39.4%	57.7%
Cross-node traffic	45.9%	42.5%	33.2%
Cross-rack traffic	30.6%	18.1%	9.1%

Join is a widely used operation in data query. The application utilizes the left-outer-join that connects a large data set to a small data set (i.e., 2GB$\times$512MB) from *Ratings and classification data* [4]. The same as before, we set the number of executors required to 30, 40, and 50 respectively in the procedure.

Fig. 6 shows the performance comparison of relevant methods. Fig. 6(a) explains that when the number of executors required is 50, by comparison with *spreadOut* and *noSpreadOut*, *approximate algorithm* reduces the job execution time by 66.0% and 27.5% separately. It observes that compared with the performance under the sort benchmark, the performance of *approximate algorithm* under the join benchmark is more significant. This is because join generates a larger amount of intermediate data, which leads to much more data communication for the reduce tasks. It in turn makes the effect of optimizing the data locality by executor allocation more prominent. Fig. 6(b) further illustrates that *approximate algorithm* outperforms others. In particular, when the required number of executors



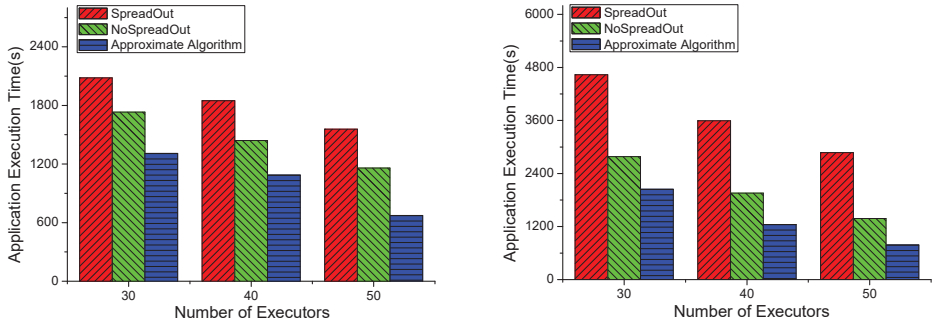
(a) Job execution time (b) Reduce stage execution time

Fig. 6. Performance comparison under Join.

is 30, compared with *spreadOut* and *noSpreadOut*, *approximate algorithm* decreases the reduce stage execution time by 43.2% and 25.2%, respectively.

(2) Macro-benchmark

To evaluate the performance under more complex applications, we select two popular machine learning algorithms pageRank and LDA from the Spark examples for testing. Since these two applications contain one or more jobs, in which every job usually contains a lot of stages, the application execution time is used for evaluation.



(a) PageRank (b) LDA

Fig. 7. Performance comparison under macro-benchmark.

PageRank is a widely recognized iterative algorithm for ranking web pages according to their importance. The experiment uses 10GB data set of the *WT10g* [7], and set the parameter *numIterations* to 10 in the procedure. The application execution consists of 1 job and 13 stages.

From the experimental result of Fig. 7(a), it can be seen that compared with *spreadOut* and *noSpreadOut*, *approximate algorithm* has the shortest application execution time. In

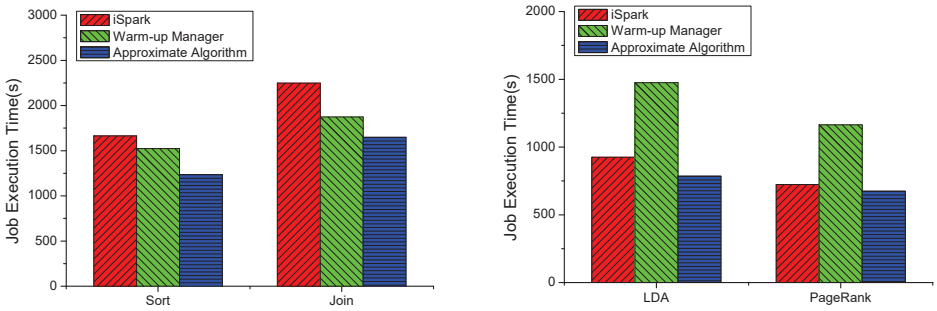
particular, when the number of executors required is 40, *approximate algorithm* reduces the application time by 41.2% and 24.6% respectively.

LDA is a document generation model in natural language processing, which identifies the hidden subjects in a large-scale documents. The experiment runs on 20GB *arXiv Bulk Data data set* [5] and the procedure sets the parameter $maxIterations$ to 20. This application is concretely executed as 26 jobs and 90 stages totally.

The experimental results illustrate that *approximate algorithm* has a greater performance advantage than other two methods, as shown in Fig. 7(b). In particular, when the number of executors required is 50, *approximate algorithm* decreases the application time by 72.7% and 43.2% compared with *spreadOut* and *noSpreadOut*, respectively. As we can see for the application with many jobs and stages, such as pageRank and LDA, optimizing the data locality by executor allocation in multiple reduce stages can bring a substantial performance benefit.

(3) Performance comparison with other methods

We also compare the performance of *approximate algorithm* with other two recent executor allocation methods: *iSpark* and *Warm-up Manager*. Fig. 8(a) shows the experimental results under the micro-benchmark: sort and join when the number of executors required is set to 50. We can see that *approximate algorithm* decreases more job execution time than *iSpark* and *Warm-up Manager*. In particular, under the join benchmark, the execution time of *approximate algorithm* is reduced by 26.7% and 12.1%, respectively. Meanwhile, Fig. 8(b) also explains that *approximate algorithm* has better performance than *iSpark* and *Warm-up Manager*, and *iSpark* outperforms *Warm-up Manager* under the macro-benchmark: LDA and pageRank which are iterative applications.



(a) Under micro-benchmark (b) Under macro-benchmark

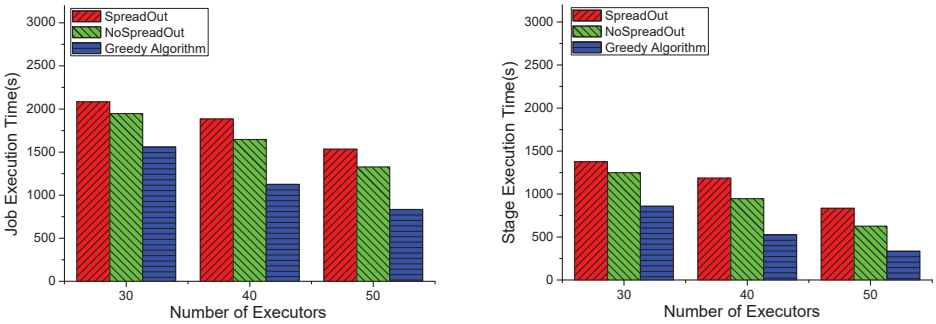
Fig. 8. Performance comparison with other methods.

Not satisfy the triangular inequality. We replace the bandwidth 10Gbps of half inside and outside racks with 20Gbps based on the original data center, so that the network distance matrix of executors does not satisfy the triangular inequality.

(1) Micro-benchmark

Fig. 9 shows the performance comparison of *spreadOut*, *noSpreadOut* and *greedy algorithm* under the sort benchmark. It can be seen that *greedy algorithm* has a shorter

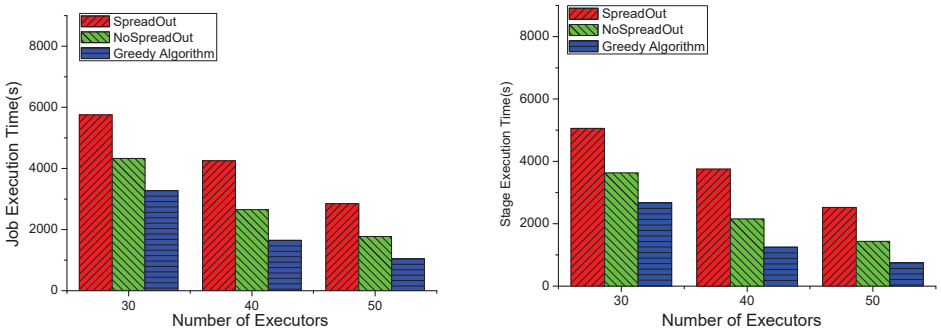
job running time than other methods. In particular, when the number of executors required is 30, *greedy algorithm* shortens the job execution time by 25.1% and 20.0% compared with *spreadOut* and *noSpreadOut*, respectively. Fig. 9(b) shows that when the number of executors required is 50, the execution time of reduce stage reduced by *greedy algorithm* is 59.7% and 46.3% respectively.



(a) Job execution time (b) Reduce stage execution time

Fig. 9. Performance comparison under Sort.

Fig. 10 depicts the performance results under the join benchmark. Fig. 10(a) shows that when the number of executors required is 40, *greedy algorithm* decreases the job running time by 61.4% and 37.9% over *spreadOut* and *noSpreadOut*, respectively. Fig. 10(b) further verifies the performance advantage of *greedy algorithm*: when the number of executors required is 50, it decreases the reduce stage execution time by 70.4% and 47.9%, respectively.



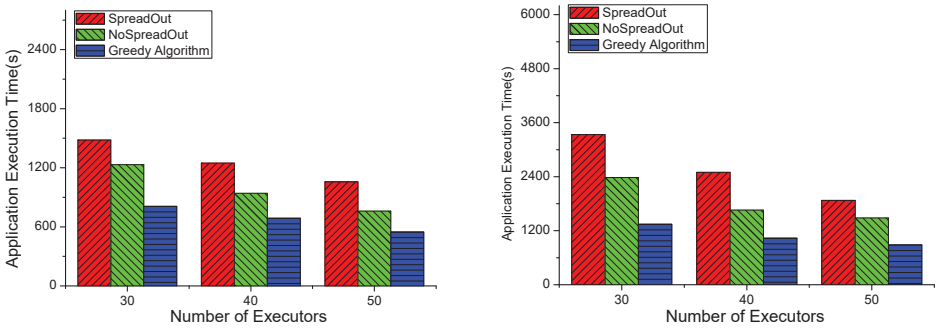
(a) Job execution time (b) Reduce stage execution time

Fig. 10. Performance comparison under Join.

(2) Macro-benchmark

The performance comparison under pageRank is exhibited in Fig. 11(a). It can be seen that compared with *spreadOut* and *noSpreadOut*, *greedy algorithm* has the shortest application execution time. In particular, when the number of executors required is 30, *greedy algorithm* decreases the application time by 45.4% and 34.3% compared with *spreadOut* and *noSpreadOut*, respectively.

Fig. 11(b) shows the experimental results under the LDA benchmark. It illustrates that *greedy algorithm* can run the LDA application faster than *spreadOut* and *noSpreadOut*. In particular, when the number of executors required is 50, *greedy algorithm* decreases the application execution time by 70.4% and 47.9%, respectively.



(a) PageRank (b) LDA

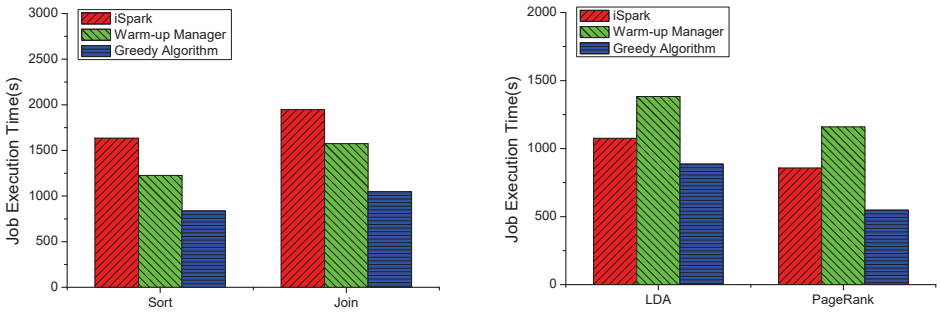
Fig. 11. Performance comparison under macro-benchmark.

(3) Performance comparison with other methods

When the number of executors required is set to 50, the performance comparison of *greedy algorithm* with *iSpark* and *Warm-up Manager* is shown in Fig. 12. The experimental results illustrate that under the micro-benchmark and macro-benchmark, *greedy algorithm* always has the shortest job execution time. In particular, under the pageRank benchmark, *greedy algorithm* reduces the execution time by 36.2% and 52.8% compared with *iSpark* and *Warm-up Manager*, respectively.

5.3. Time Overhead of Algorithm

During the above experiment process, we recorded the average time required to start a set of executors for an application, the results are shown in Table 5. Comparing with *spreadOut* and *noSpreadOut*, because *approximate algorithm* and *greedy algorithm* needs additional computation in order to select a subset from the executors that are allowed to start on each node, they will take more time to launch the executors. This has a negative impact on the performance of our proposed algorithms, especially when the number of executors required is large. In addition, the time complexity of *approximate algorithm* and *greedy algorithm* are $O(m^2)$ and $O(k \times m^3)$ respectively, where k is the number of executors required, and m is the number of executors allowed to start, so *greed algorithm* is slower than *approximate algorithm* for the executor start time. However, in contrast



(a) Under micro-benchmark (b) Under macro-benchmark

Fig. 12. Performance comparison with other methods.

with the application/job execution time, the time overhead of *approximate algorithm* and *greedy algorithm* only takes up only a small part, so it can be ignored.

Table 5. Average Executor Start Time

Number of Executors	30	40	50
spreadOut	43.7ms	49.4ms	54.6ms
noSpreadOut	27.2ms	29.5ms	32.8ms
approximate algorithm	1.25s	2.67s	3.38s
greedy algorithm	2.36s	3.73s	4.75s

6. Conclusion

This paper has attempted to optimize the data locality by executor allocation for the reduce stage in Spark computing environment. We first calculate the distance matrix of executors and formulate the optimal executor allocation problem to minimize the total communication distance. This problem is proved to be an NP-Hard problem. Then, for the cases where the network distance between executors satisfies and does not satisfy the triangular inequality, an approximate algorithm and a greedy algorithm are proposed respectively. Finally, we conduct extensive experiments and the results show that our algorithms can optimize the data locality for reduce tasks and improve the application/job performance. In general, for different workload types, the proposed algorithms can bring more performance gain to the reduce-input heavy jobs and iterative applications than the map and reduce-input heavy jobs.

Acknowledgments. The work is supported by Scientific Research Projects funded by Hunan Provincial Department of Education (22B0451), and Doctoral Research Startup Foundation of University of South China (No.200XQD083).

References

1. ”Apache Spark”,https://spark.apache.org/
2. ”Apache Hadoop”,https://hadoop.apache.org/
3. ”Apache Spark”,https://spark.apache.org/docs/3.3.0/job-scheduling.html/
4. ”Ratings and classification data.”,http://webscope.sandbox.yahoo.com
5. arxivbulkdata. https://arxiv.org/help/bulk_data_s3/
6. Wikipedia corpus. https://www.english-corpora.org/wiki/
7. Wt10g. http://ir.dcs.gla.ac.uk/test_collections/
8. Arslan, E., Shekhar, M., Kosar, T.: Locality and network-aware reduce task scheduling for data-intensive applications. In: Tang, W., Zhao, Y., Zheng, Z. (eds.) Proceedings of the 5th International Workshop on Data-Intensive Computing in the Clouds, DataCloud '14, New Orleans, Louisiana, USA, November 16-21, 2014. pp. 17–24. IEEE (2014)
9. Beaumont, O., Lambert, T., Marchal, L., Thomas, B.: Performance analysis and optimality results for data-locality aware tasks scheduling with replicated inputs. Future Gener. Comput. Syst. 111, 582–598 (2020)
10. Cheng, L., Wang, Y., Liu, Q., Epema, D.H.J., Liu, C., Mao, Y., Murphy, J.: Network-aware locality scheduling for distributed data operators in data centers. IEEE Trans. Parallel Distributed Syst. 32(6), 1494–1510 (2021)
11. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (Jan 2008)
12. Fu, Z., He, M., Tang, Z., Zhang, Y.: Optimizing data locality by executor allocation in reduce stage for spark framework. In: Shen, H., Sang, Y., Zhang, Y., Xiao, N., Arabnia, H.R., Fox, G., Gupta, A., Malek, M. (eds.) Parallel and Distributed Computing, Applications and Technologies. pp. 349–357. Springer International Publishing, Cham (2022)
13. Fu, Z., Tang, Z., Yang, L., Liu, C.: An optimal locality-aware task scheduling algorithm based on bipartite graph modelling for spark applications. IEEE Trans. Parallel Distributed Syst. 31(10), 2406–2420 (2020)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
15. Gu, H., Li, X., Lu, Z.: Scheduling spark tasks with data skew and deadline constraints. IEEE Access 9, 2793–2804 (2021)
16. Guo, Z., Fox, G., Zhou, M.: Investigation of data locality in mapreduce. IEEE/ACM International Symposium on Cluster Cloud & Grid Computing pp. 419–426 (2012)
17. Lee, S., Jo, J., Kim, Y.: Survey of data locality in apache hadoop. In: Iwashita, M., Shimoda, A., Chertchom, P. (eds.) 2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering, BCD 2019, Honolulu, HI, USA, May 29-31, 2019. pp. 46–53. IEEE (2019)
18. Lu, X., Islam, N.S., Wasi-ur-Rahman, M., Jose, J., Subramoni, H., Wang, H., Panda, D.K.: High-performance design of hadoop RPC with RDMA over infiniband. In: 42nd International Conference on Parallel Processing, ICPP 2013, Lyon, France, October 1-4, 2013. pp. 641–650. IEEE Computer Society (2013)
19. Ma, X., Fan, X., Liu, J., Li, D.: Dependency-aware data locality for mapreduce. IEEE Trans. Cloud Comput. 6(3), 667–679 (2018)
20. Morisawa, Y., Suzuki, M., Kitahara, T.: Flexible executor allocation without latency increase for stream processing in apache spark. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 2198–2206 (2020)
21. Naik, N.S., Negi, A., Bapu, B.R.T., Anitha, R.: A data locality based scheduler to enhance mapreduce performance in heterogeneous environments. Future Gener. Comput. Syst. 90, 423–434 (2019)

22. Neciu, L., Pop, F., Apostol, E.S., Truica, C.: Efficient real-time earliest deadline first based scheduling for apache spark. In: Potolea, R., Iancu, B., Slavescu, R.R. (eds.) 20th International Symposium on Parallel and Distributed Computing, ISPDC 2021, Cluj-Napoca, Romania, July 28-30, 2021. pp. 97–104. IEEE (2021)
23. Shabeera, T.P., Kumar, S.D.M.: A novel approach for improving data locality of mapreduce applications in cloud environment through intelligent data placement. Int. J. Serv. Technol. Manag. 26(4), 323–340 (2020)
24. Shabeera, T.P., Kumar, S.D.M., Chandran, P.: Curtailing job completion time in mapreduce clouds through improved virtual machine allocation. Comput. Electr. Eng. 58, 190–202 (2017)
25. Shang, F., Chen, X., Yan, C.: A strategy for scheduling reduce task based on intermediate data locality of the mapreduce. Clust. Comput. 20(4), 2821–2831 (2017)
26. Sun, M., Hang, Z., Zhou, X., Lu, K., Li, C.: Hpso: Prefetching based scheduling to improve data locality for mapreduce clusters. In: Conference on Design (2014)
27. Tan, J., Meng, S., Meng, X., Zhang, L.: Improving reducetask data locality for sequential mapreduce jobs. In: 2013 Proceedings IEEE INFOCOM. pp. 1627–1635 (2013)
28. Tang, S., He, B., Yu, C., Li, Y., Li, K.: A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications. IEEE Trans. Knowl. Data Eng. 34(1), 71–91 (2022)
29. Tang, Z., Ma, W., Li, K., Li, K.: A data skew oriented reduce placement algorithm based on sampling. IEEE Trans. Cloud Comput. 8(4), 1149–1161 (2020)
30. Vavilapalli, V.K., Murthy, A.C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowe, J., Shah, H., Seth, S., Saha, B., Curino, C., O'Malley, O., Radia, S., Reed, B., Baldeschwieler, E.: Apache hadoop YARN: yet another resource negotiator. In: Lohman, G.M. (ed.) ACM Symposium on Cloud Computing, SOCC '13, Santa Clara, CA, USA, October 1-3, 2013. pp. 5:1–5:16. ACM (2013)
31. Xia, T., Wang, L., Geng, Z.: A reduce task scheduler for mapreduce with minimum transmission cost based on sampling evaluation. International Journal of Database Theory & Application (2015)
32. Yan, X., Wong, B., Choy, S.: R3S: rdma-based RDD remote storage for spark. In: Proceedings of the 15th International Workshop on Adaptive and Reflective Middleware, ARM@Middleware 2016, Trento, Italy, December 12-16, 2016. pp. 4:1–4:6. ACM (2016)
33. Yang, D., Rang, W., Cheng, D., Wang, Y., Tian, J., Tao, D.: Elastic executor provisioning for iterative workloads on apache spark. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 413–422 (2019)
34. Zaharia, M., Borthakur, D., Sarma, J.S., Elmeleegy, K., Shenker, S., Stoica, I.: Delay scheduling:a simple technique for achieving locality and fairness in cluster scheduling. In: European Conference on Computer Systems. pp. 265–278 (2010)
35. Zhang, X., Luo, F., Jia, Z., Shen, J.: Prefetching method for hadoop mapreduce environments. Xi'an Dianzi Keji Daxue Xuebao/Journal of Xidian University 41(2), 191–196 (2014)

Zhongming Fu received the PhD degree at the College of Computer Science and Electronic Engineering, Hunan University, China, in 2020. He is currently a lecturer of School of Computer Science, University of South China. His research interests include the parallel computing, the improvement and optimization of task scheduling module in Hadoop and Spark platforms. He has published several papers in IEEE TCC and TPDS.

Mengsi He received the master degree at the College of Computer Science and Electronic Engineering, Hunan University, China, in 2020. She is currently a technician of School of Computer Science, University of South China. Her current research interests include

parallel computing, the improvement and optimization of the graph computation module in Spark platforms.

Zhuo Tang received the PhD degree in computer science from the Huazhong University of Science and Technology, China, in 2008. He is currently a professor with the College of Computer Science and Electronic Engineering, Hunan University. He is also the chief engineer with the National Supercomputing Center, Changsha. He has authored or coauthored almost 90 journal articles and book chapters. His research interests include distributed computing system, cloud computing, and parallel processing for big data, including distributed machine learning, security model, parallel algorithms, and resources scheduling and management in these areas. He is a member of the IEEE/ACM and CCF.

Yang Zhang received the Ph.D. degree in software engineering from the National University of Defense Technology (NUDT), China, in 2019. He is currently an Assistant Professor of the PDL laboratory, NUDT. His research interests include empirical software engineering, social network analysis, and DevOps.

Received: January 31, 2022; Accepted: December 10, 2022.

Computer Science and Information Systems 20(1):513–535 https://doi.org/10.2298/CSIS220131066X

Efficient Neural Network Accelerators with Optical Computing and Communication*

Chengpeng Xia[1], Yawen Chen[1]**, Haibo Zhang[1], Hao Zhang[1], Fei Dai[1], and Jigang Wu[2]

[1] Department of Computer Science, University of Otago
Dunedin 9016, New Zealand
chengpeng.xia@postgrad.otago.ac.nz
{yawen, haibo, travis}@cs.otago.ac.nz
hao.zhang@postgrad.otago.ac.nz
[2] School of Computers, Guangdong University of Technology,
Guangzhou 510006, China
asjgwuch@outlook.com

Abstract. Conventional electronic Artificial Neural Networks (ANNs) accelerators focus on architecture design and numerical computation optimization to improve the training efficiency. However, these approaches have recently encountered bottlenecks in terms of energy efficiency and computing performance, which leads to an increase interest in photonic accelerator. Photonic architectures with low energy consumption, high transmission speed and high bandwidth have been considered as an important role for generation of computing architectures. In this paper, to provide a better understanding of optical technology used in ANN acceleration, we present a comprehensive review for the efficient photonic computing and communication in ANN accelerators. The related photonic devices are investigated in terms of the application in ANNs acceleration, and a classification of existing solutions is proposed that are categorized into optical computing acceleration and optical communication acceleration according to photonic effects and photonic architectures. Moreover, we discuss the challenges for these photonic neural network acceleration approaches to highlight the most promising future research opportunities in this field.

Keywords: Optical neural networks, Optical interconnection networks, Neural network accelerator.

1. Introduction

The wide applications of Artificial Intelligence (AI), such as computer vision, speech recognition and language processing, call for efficient implementation of the model training and inference phases in machine learning [61]. Especially for Artificial Neural Networks (ANNs), due to the seminal work by Hinton et al. on deep learning in 2006, ANNs have reappeared in people's vision [30]. Multiple neural networks have been studied and applied in different fields. However, with large data sets and massively interconnected

* This is an extended version of The 22nd International Conference on Parallel and Distributed Computing, Applications and Technologies.

** Corresponding author

ANNs, the traditional computer architectures suffer from the efficient training and inference due to the limited device computing efficiency and energy consumption.

To increase computing performance and energy efficiency, both hardware and software acceleration have been studied extensively in academia and industry. The specifically tailored electronic solutions have been regarded as ideally suitable for ANNs training, such as Graphics Processing Units (GPU), Tensor Processing Unit (TPU) and Field Programmable Gate Arrays [77,48]. These novel electrical architectures focus on high inter-chip bandwidth for big data traffic, memory architectures for matrix multiplications and advanced numerical calculation method to support model parallelism and data reuse. Nevertheless, the demand for computing power in ANNs is continually growing, and electronic solutions are still limited by the energy consumption of physical limits [45].

Along with the development of photonic devices and integrated optics, it has been considered a possible alternative for electronic architectures in the future to use optical architectures. There are many optical solutions emerging for communications and computing acceleration of ANNs as the times require. To this aim, some studies focus on optical linear transformations in passive optical network that enable to be operated without power consumption and with minimal latency [53], and the optical logic gates has also been proposed in different structures [34]. Further, optical devices were integrated to implement ANNs, with the aim of increasing the training speed and the energy efficiency [1,58]. For accelerating the communication of ANNs, optical on/off chip network architectures with different parallelization strategies and topologies have also been designed for decreasing ANN training workload and increasing and data transmission speed.

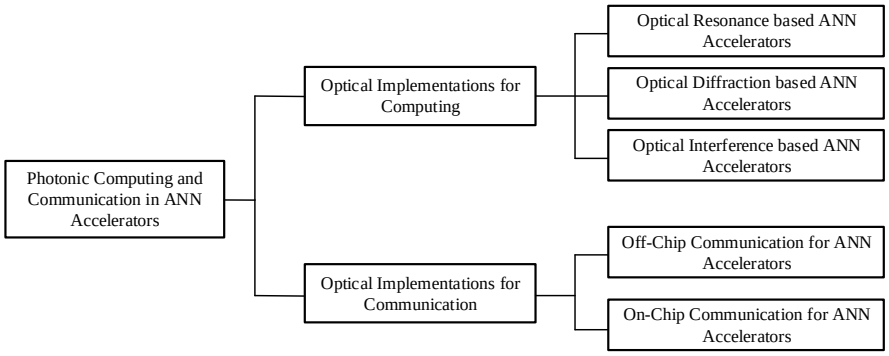


Fig. 1. Classification of Photonic Implementation in ANN Accelerators.

In this paper we present a survey of approaches for implementing optical Neural Network (ONN) accelerator. A classification of the existing solutions is proposed which includes two categories: optical computing implementations and optical communication implementations for ANN accelerators, as can be seen in Fig. 1. Existing reviews on optical ANN accelerators have either focused on reviewing performance and energy of a specific type of optical ANN computing architecture such as reservoir computing architectures

[54], [31] and Broadcast-and-Weight architectures [18], [68]), or proposed a simplified taxonomy of the realized neural network models such as CNNs, SNNs [20]. By comparison, we present a different and more comprehensive review of photonic ANN from two aspects including computing acceleration and communication acceleration approaches with a bottom-up classification across design-layer abstractions: from lower-level optical devices, to the neuron microarchitectures, and covering a variety of integrated neural network.

Recently, some works have focused on the computing acceleration in neural network and on bottlenecks of photonics technologies [42], [74]. However, these works have ignored the contribution of on-chip optical communication to neural networks acceleration. Compared to our previous work [72], this paper provides the review of optical devices from lower-level, and a more comprehensive summary of optical computing and communication acceleration neural networks. In addition, the advantages and disadvantages of existing works are summarized by comparing literature.

The remainder of this paper is organized as follows: In Section 2, we present the motivations behind ANN accelerator, and introduce a taxonomy of the approaches presented in the literature. The relevant optical devices is reviewed in terms of the application in ANN accelerators. In Section 3, we describe the optical architectures devised for the computing implementations in ANNs. In Section 4 the most relevant solutions are reviewed according to the categories of optical communication for ANNs acceleration. While in Section 5, we discuss the challenges and future research opportunities in this field, and Section 6 concludes the paper.

2. Background

2.1. Optical Neural Network Accelerators

The exiting researches of ANN accelerators are mainly focused on the development of electronic architecture specially tailored to neural network, such as GPU, TPU, FPGA and ASIC optimized neural network by adjusting the computing architecture. However, after decades of prosperous development of electronic computers, the current silicon-based computer circuits are reaching their physical limits [16]. Conventional high-performance computer architectures still cannot break through the bottleneck of the memory wall, and computing performance is limited by bandwidth and huge date processing workload and power consumption [14].

In recent years, some researches began to explore analog electronic circuits to address the memory wall challenges that can meet the ANN computation requirements. Quantum Neural Networks [23], Processing-in-Memory [13] and Memristor [2] have been specially designed to implement ANN acceleration. Processing-in-Memory employs the memory arrays themselves for computing to reduce the movement of data between CPU and memory, which obviates memory cell redesign and has low area overhead and friendly manufacture. However, The accuracy of Processing-in-Memory is limited by analog calculations [32]. Memristor-based Accelerators mainly consist of resistor array with memory and analog crossbars. The main drawback of Memristor is the concern on the high power consumption [71]. While all these three technologies lack mature development platforms and industry standards.

Photonics have demonstrated great potentialities for various application in ANN accelerators. In order to implement the functionality of neural networks in photonic networks, the present research efforts have been undertaken numerously. In comparison with state-of-the-art electrical architectures, the optical solutions are expected to enhance computational speed and energy efficiency when performing data transfer and training tasks [58]. The rationale behind optical architectures can considerably decrease the energy cost both in logical calculation and data transmission by using passive optical network to execute the linear operations in a typical ANN [28]. The application of passive components in an integrated optical circuit enables high-speed operation while consuming less than the transmitter and receiver energy limits. Hence, analog optical computing circuits are an exciting research field possibility for high-performance computing, especially for ANNs. On the other hand, optical interconnection networks have been well studied by a large amount of works, due to its unique advantage in data transmission such as high bandwidth density, low power consumption and immune to electromagnetic effect [49]. Optical network with the implementation of Wavelength-Division-Multiplexing technology (WDM), is also suitable for neural networks to accelerate on-chip or off-chip communications [17].

2.2. Photonic Devices

Over the past few years, optical communication has been generally applied in remote communications and data centers for cost-effective and high bandwidth interconnects. With the development of silicon photonics, photonic devices are becoming miniaturized and low power consumption that makes it increasingly possible to integrate photonic network architectures. The integrate optical devices are considered to develop optical computing chips with better performance. The computing platform using Wavelength-Division Multiplexing technology can carry more than 64 wavelengths of light in a single waveguide. Each optical signal enables wavelengths to carry different data at high transmission speed without any crosstalk [4]. In optical network, the multiplexed signals are switched and separated by the Microring Resonators, Mach-Zehnder Interferometers and other optical devices to achieve optical communication and computing. The optical signals are eventually converted into electrical signals for storage or further processing by photo detectors, optical to electronic converters and other devices. However, there are still several challenges for optical devices to support robust computation and communication at chip scale. Therefore, this paper reviews some basic optical devices that are used for computing and communication in optical acceleration.

Lasers The first challenging requirement for optical architecture is to develop an effective and stable chip-scale optical source. There are two types light source in the integrated optical circuit: on-chip laser and off-chip laser, and the different structures lead to different advantages and disadvantages for optical interconnections [81]. The off chip lasers can offer excellent luminous performance and stable temperature control but limited by high optical power loss because of the coupling. The on chip lasers could possibly offer a higher integrated ability and lower power loss, whereas the development of on chip lasers is limited by the low emission efficiency of silicon that is one main obstacle preventing the integration of optical interconnection [29,82].

Light sources are emitted by several ways. Firstly, Vertical-cavity surface emitting lasers (VCSELs) are now key optical sources in optical communications which the lasers

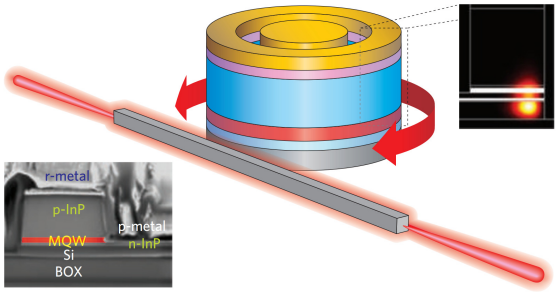


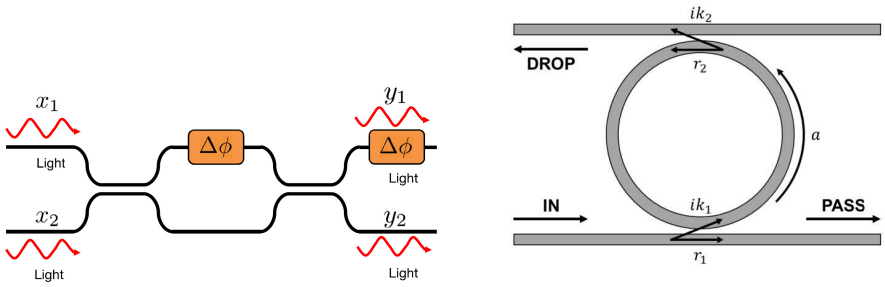
Fig. 2. A hybrid microring laser with a Si bus waveguide.

are perpendicular to the surface of substrates. VCSELs are widely utilized to the optical neural network as they meet the requires of cost-effective, integratable for array and high coupling efficiency to waveguides [73]. Meanwhile, hybrid silicon lasers with small size and a short cavity structure are verified have greater footprint efficiency. Secondly, as shown in Fig. 2 the hybrid microring lasers have been experimentally demonstrated on Si integrated platform that be contained about 400 laser devices in $1cm^2$ chip [64]. In the optical computing, lasers are employed to implement some functions of neural network. Light sources with integrated modulator can directly output light to carry data by adjusting the amplitude, power and phase of light [63]. However, the thermal impedance caused by high density lasers lying is still a major hurdle for large scale integration.

Mach-Zehnder Interferometers Mach-Zehnder Interfeometer (MZI) is broadly applied to develop optical modulators, switches and filters in photonic architectures [80,65]. A MZI is composed of two beam splitters and two phase shifters. Fig. 3(a) shows the layout of a MZI device. While the fixed 50:50 beam splitters are not configurable, the two phase shifters are configurable by adjusting the angle. The input optical signal is split proportionally as it passes through the beam splitter. By applying power to the two phase shifters, the MZI can be controlled to provide phase shifting or attenuation for the optical signals which pass through the two arms. This enables MZIs to work as the directional couplers or more simply as the optical switches. Based on the above functions, MZI has been utilized to realize fundamental logic operation by optical power level, phase adjusting and output port scheduling. In [6], authors designed a cascaded MZIs structure with interference of multiple beams that obtained various gates by detecting optical intensity at different points such as AND, OR, NAND, XNOR and so on. According to the research above, MZI-based logic gates are studied to support optical neural network computation [55]. The cascade MZIs array is used as the basic unit of matrix multiplication, which has been attracting a great deal of attention recently. The singular value segmentation is used to make it suitable for matrix multiplication operation in ANNs [58]. Such neural functions and their implementations are discussed in the next section.

Microring Resonators A Microring Resonator (MRR) can be seen as a closed waveguide circular and a common structure of MRR is that two bus waveguides border upon a ring waveguide. The ring waveguide resonates when the path-length of resonator cavity is

equal to the integer multiple of the input wavelength [7]. As shown in Fig. 3(b), the MRR contains a ring waveguide, an input bus waveguide and a drop bus waveguide. When the resonance occurs, the input lights will be passed the ring waveguide and turned to the drop waveguide, conversely, the lights will be routed to the pass waveguide. Hence, the MRRs are used as switches or filters in photonic communication, especially for Wavelength Division Multiplexing technology [76]. The WDM technology arrows optical signal with different wavelength to carry more data without interference, which increases parallel process in the optical network. Moreover, MRRs and WDM optical signal are employed to realize weight accumulation in optical neural network, in which the cascaded add-drop MRRs is grid in shape, and is called weight bank [67,78]. To sum up, MRRs and MZIs are designed as switches, filters and modulators in optical communication and computing. Compared to the MZI, since the dropped output of the MRR can be directly monitored at each wavelength of the WDM signal, it is more straightforward to set the elements of weight than using meshed-MZI [50].



(a) Mach-Zehnder Interferometer (b) Microring Resonator

Fig. 3. The structures of a Mach-Zehnder Interferometer and a Microring Resonator.

Photodetectors Photodetectors are commonly used to detect optical signals and convert optical signals into electrical signals. Therefore, the optical detector can be used as an optical signal output device to be connected at the end of the optical network [38]. In addition to being the photo-to-electric conversion devices in optical neural networks, photodetectors are also used to realize the operations of ANNs. In [59] the positive signals and negative signals are propagated and superimposed on different optical waveguides, and two balanced photodetectors are set at the end of the waveguides to detect the total optical power in positive and negative waveguides, respectively. The detected lights are converted into the currents by the photodetectors. The balanced photodetectors can calculate the difference between positive and negative currents, which enable optical neural network to realize the accumulation operation for different kernels.

The performance of a photodetector is related to the responsivity which is defined as the optical power that can be detected per unit area. An efficient photodetector with a lower responsivity have better detection accuracy for low-power input light sources, and a photodetector with high responsitivity needs higher power consumption to meet its detec-

tion accuracy demand. For driving the photodetector effectively, the optical power reaching the photodetector should be greater than the responsivity. The input optical power, energy loss and responsivity are considered to be the critical factors when choosing an input light source. Therefore, the performance of photodetector affects the bandwidth and power consumption of optical networks.

3. Optical Implementations for Computing

This section summarizes optical implementation for ANN computing acceleration. The innovation of these works mainly focus on constructing different types of neural networks, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Spiking Neural Network (SNN) and Multi-layer Perceptron (MLP). The use of optical fundamental principles and photonics components makes matrix multiplication available. The performance of silicon photonics ANN accelerator is also affected by the optical principle and architecture. Hence, these implementations will be categorized according to the primary photonics principles.

3.1. Optical Resonance based Neural Network Accelerators

Inspiration comes from the field of neurobiology in which each neuron communicates in the way of short pulses, the resonance-based photonic neural networks have been investigated widely. Wavelength division multiplexing is applied in optical neural networks depending on the resonance modulation property and wavelength specificity of MRRs. The WDM channel transmits multiple wavelengths in the same waveguide without interference, which reduces the number of optical devices in ANNs implementation to some extent. In [66], the authors integrated several MRRs to design an optical neural network with on-chip architecture that is called Broadcast-and-Weight (BW). The input signals are transmitted parallelly in a bus waveguide, and the weights of neurons are loaded into the MRRs. Fig. 4 shows that the multiple wavelengths are aggregated in a waveguide by multiplexer, and the MRRs act as the neurons. While the passive splitters are employed at the end of the bus to broadcast data, so the output signals of the bus is connected to all neurons. The weight bank is actually a set of reconfigurable filters composed of MRRs, which can map the weight to different network layers by attenuating the resonance wavelength.

Based on the BW protocol, authors presented an optical convolution neural network accelerator (PCNNA) in [46]. The PCNNA can propagate different CNN layers in a same optical circuit because of the using of single layer multiplexing architecture. The authors considered that convolution calculations of different kernels can be executed in parallel because each layer of PCNNA shares the same convolution kernel value. In the overall framework, the PCNNA is configured to run on two clock domains with different speeds because the optical circuits runs faster than the electronic circuits. Thus, convolution results and kernel weights of each layer can be stored in an off-chip Dynamic Random Access Memory (DRAM). Kernel weights are loaded into the ANN by tuning microrings in the MRR banks. The authors showed how their accelerator implements AlexNet, and they claimed a third-order cubic polynomial time of magnitude execution time improvement over electronic engines. An extension of WDM for CNN implementation was

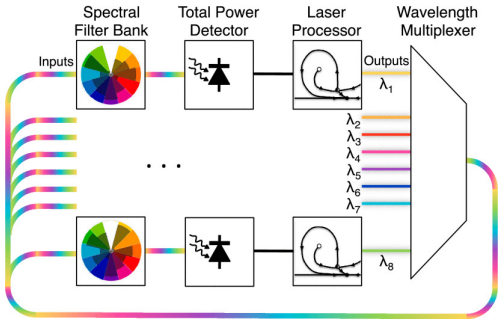


Fig. 4. The Broadcast-and-Weight architecture proposed by [66].

explored in [60] where authors combined MRR and MZI to design an all-optical multiplication and accumulation. Based on the optical resonance, an array of tuned MRRs is utilized to realize WDM and optical *AND* operations. The cascade MZIs can operate pure optical shift accumulation on each sequential *AND* operation.

The optical resonance is also widely used in the implementations of SNN. In [66], the spiking neuron unit is named processing network node (PNN), and each PNN is composed of weight banks, photodetectors (PDs) and laser diodes (LDs). The weights are divided into excitatory and inhibitory weights that are stored in the weight banks. Different types of weights are received by two PDs to complete the accumulation of PNN. Finally, a broadcast loop is used to propagate the WDM signals between these PNNs. In addition to the BW architecture, authors in [10] mentioned that the integration of MRRs and PCM material $(\mathrm{Ge_2Sb_5Te_5})$ were used to fire neurons. The bidirectional multi-ports integrating action of MRR was used to figure up the film potential in the effect of the weighted sum. Spikes output when the film potential of a neuron exceeds the thresholds. There is a few works on optical resonance implementation for reservoir computing (RC). A 4×4 swirl topology-based reservoir was proposed in [21]. The work utilized MRRs and basic Boolean operations, in which non-linear elements (MRRs) are the nodes of the recurrence network. The input signals/weight matrix is mixed in the swirl to realize computing.

3.2. Optical Diffraction based Neural Network Accelerators

In optical network, diffraction effects tend to limit the performance of optical units, while efficient optical neural networks can be realized by designing diffraction-based architectures using appropriate optical elements. For example, a holographic optical element (HOE), which is usually used for information storage, can be utilized to store weights and propagation directions in neural network [51]. Using HOEs, authors in [84] explored an all optical neural network. Firstly, according to the number of input optical signals, the spatial light modulators are divided into several regions, and the holograms of input signals can be obtained by superimposing phase gratings in front of each region. The optical matrix multiplication is implemented by diffraction of the optical signal in HOEs, where the weights of the ANN are loaded in the direction of the input optical signals. The convex lens are set behind the HOE to perform Fourier transform on the diffracted signal. Finally, all signals are output to the receiving plane to complete the accumulation operation.

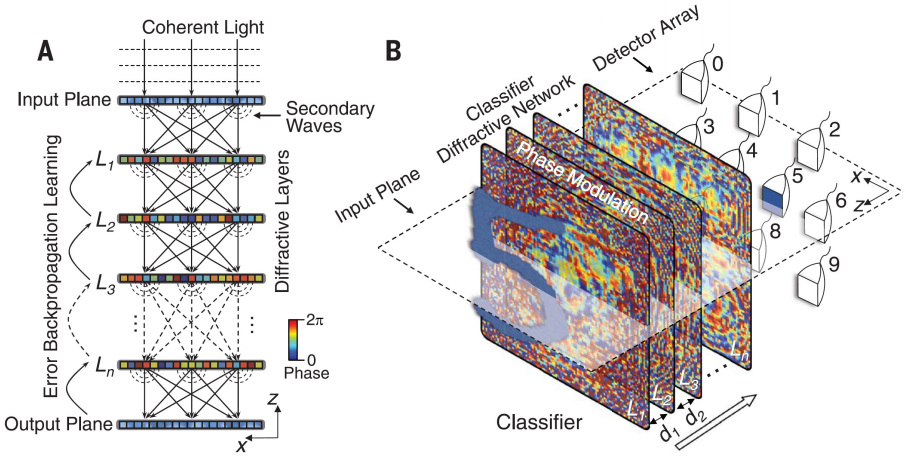


Fig. 5. Diffractive deep neural networks (D^2NN) depicted by [40].

Apart from holograms, an optical ANN with cascade phase mask structure was proposed in [40], which is called D^2NN. As illustration of Fig. 5, the fully connected layers consist of several sequential and hierarchical phase masks. These phase masks are 3D printed, and each mask represents a layer in the fully connected network. The grids inside the mask represent different neurons. The refractive index of the grids can be changed by setting different thickness. Hence, the D^2NN maps the weights of neurons to the state of grids and phase of masks. When the input optical signals pass through the mask, the matrix multiplication will perform because of diffraction effect. The output signals will directly enter the next mask that represents a direct fully connection with the next layer in ANN. Therefore, the cascade mask array forms a multi-layer fully connected optical neural network with variable weight. Finally, an array of detectors in the last mask is deposited to measure the intensity of the output light, which can be defined as the classification results of D^2NN. Specifically, Lin et al. [40] assumed that the light wavelength is λ, the size of neurons is usually about 0.5λ, and the axial distance between phase masks is usually set as about 40λ, such that 200×200 neurons can be packed in an area of $8 \times 8cm^2$ each layer with an axial distance of $3cm$ between layers. Considering there are five layers, approximately 8 billion connections were implemented. The feasibility of D^2NN network has also been confirmed by microwave [52] and broadband incoherent light source [44] experiments. Whereas, this architecture would be more sensitive to the assembling errors, which causes it difficult to manufacture.

To the best of our knowledge, there have been no studies of the SNNs implementation based on diffractive optics. Apart from SNNs, some works focused on the realization of the diffractive-based Reservoir Computing. A reservoir used a 4×4 swirl topology was discussed in [31], where the readout layer is composed of a nonlinear optical modulator. The authors in [21] also proposed an RC architecture using pillar silicon scatterers and cavities as passive elements. The work of [8] described an all optical large system that used digital micromirrors for diffraction in its output layer. However, due to the nonlinearity of electrical domain, the update rate is severely limited to 5Hz. This study demon-

strated a system with 2025 nonlinear nodes, and implemented in the form of pixels in a spatial light modulator. The SLM will show the status of the reservoir in the form of a specklegram that will be received by a camera, and then calculate the following step required for the reservoir and encode it into the system. Finally, the optical RC system is demonstrated through experiments. The system realizes that the random interconnection between neurons in the reservoir is a random diffraction behavior through the diffractive optical element (DOE). The authors believed that random diffraction has been proved to be suitable for optical nanocrystals. Nevertheless, diffraction-based ANNs are mostly realized by free-space optic devices, which take up a lot of space and do not support large-scale neural network acceleration.

3.3. Optical Interference based Neural Network Accelerators

Different from diffraction with a large amount of beams input, interference-based optical computation requires only a small number of lights to carry information, and the lights need to transmit through the waveguide. Optical matrix multiplication of neural network based on interference is mainly realized by cascading MZIs, in which the interference of light is carried out in the directional coupler and phase shifter in MZIs. The phase, amplitude and power of the input optical signals can be changed to achieve weight loading in the initial data by adjusting the couplers and phase shifters. The authors presented a pioneering work in [58], which a coherent nano-photonic circuit is designed for all-optical neural networks, and this work lays the foundation for future interference-based neural network accelerators. As shown in Fig. 6, the optical matrix multiplication between input data and weight is realized by singular value decomposition (SVD) [37]. Specifically, the matrix M is decomposed into $M = U \sum V$, where U and V are two unitary matrices, and $\sum$ is a diagonal matrix. Accordingly, MZIs are assembled as a cascaded array with three segments that implement the matrices U, $\sum$ and V, respectively. The cascade MZIs array represents the fully connected layers of ANN. If an input optical signal passes through MZI, two parallel lights will be applied to the two phase arms of MZI, and then the occurrence of parallel light interference will realize the matrix accumulative operation of ANN. The weight information is loaded into optical neural network by interference effect, and the weight can be changed by adjusting the shifters and amplitude of the coherent wave. Hence, the energy consumption is low in the entire training process. However, the depth of the ONN in [58] is limited to $2N - 1$ affected by light attenuation, which also means that the reduction in the size of the ANN may reduce the classification accuracy. Therefore, the authors in [22] improved unitary matrix multiplier, controlled the depth of MZI layer by cluster grid and statistical Fast Fourier Transform (FFT). The authors claimed that FFT-based network is inherently more robust than grid-based network, because it has a much smaller number of MZI layers for realizing a same unitary matrix multiplier. For example, a multiplier can be realized by using a FFT network with only $log_2(N)$ layers instead of the grid network with N layers, that means the depth of grid network is 32 times that of FFT network when $N = 2^8$.

Due to the area of MZI device and the large demand for nodes in the RC network, the interference effect has not received much attention in the research of RC. Authors in [36] provided an electrical-optical nonlinear modulation transfer function by using the delay coupling technology integrated MZIs. A long distance optical fiber is used to realize a

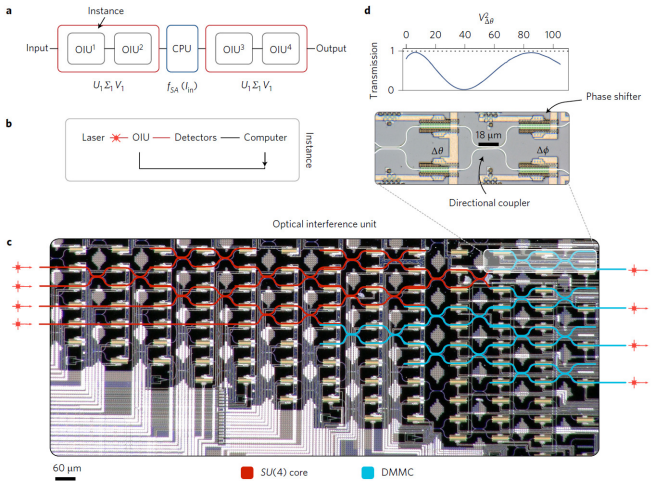


Fig. 6. Interference based photonic integrated circuit depicted by [58].

delay feedback loop, and a photodiode is used for optical detection. The electronic feedback circuit output is connected to MZI input electrode. Therefore, long distance optical fibers are divided into a number of subintervals to define virtual nodes. By extracting the virtual node state at the end of each subinterval, [36] simulated the nodes of conventional RC network.

3.4. Summary

The literature concerning photonic neural network architectures is vast, and so are the techniques and devices used to realize these architectures. In this section, we reviewed different architectures and divided the literature into resonance-based implementations, interference-based implementations, diffraction-optics based implementations. We have provided a summary of the literature on architectures covered as part of Section 3 in Table 1. The table has the devices prominently used in the architecture (first column); a brief summary of the advantages (second column); a brief summary of the disadvantages (third column); the references to the works (fourth column).

4. Optical Implementations for Communication

Existing ANNs have been challenged by the fact of high computational complexity, large amount of computational data, frequent memory access and high parallelism requirements that are widespread in current neural network workloads. In the latest ANNs, tens to hundreds of megabytes of parameters are required to execute a single inference pass. Over one billions of operations will generate large amounts of memory access requirements from the processing elements (PE) which makes existing architectures face the challenge of memory wall. In the processing of model training, a large amount of reusable data are usually generated. For example, a huge amount of filter data, input feature map data

Table 1. A summary of selected proposed optical ANN using computing and communication acceleration

	Implementation	Advantages	Disadvantages	References
computing acceleration	Use MRR weight banks for synapse; photodetector for optoelectrical conversions	Use WDM to offer high bandwidth; Use passive devices compatible with low power consumption	consumes power in electro-optical and optoelectrical conversions; low cascadability; sensitive to temperature	[66], [10], [60]
	Use HOEs	Use passive devices compatible with low power consumption, low energy loss	sensitive to the assembling errors, difficult to manufacture; area inefficient	[40], [84], [52]
	Splitters and MZIs	Higher reliability and low power overhead in comparison with using several wavelengths	Low bandwidth because of using one wavelength; area inefficient; exact splitting ratios are hard to achieve after fabrication due to variations; susceptible to noise in phase and splitting ratios	[36], [58], [22]
communication acceleration	use optical circuit switch-based topology for data reusing	high workloads parallelism; demand of ANN workloads and the singleness and repeatability of transmis-sion in workloads,		

and partial sum data are created in the processing of convolution in CNN, in which these data can be regarded as reusable resources. Hence, the CNN accelerators using dataflow optimization with Network-on-Chip (NoC) architecture have been proposed in [9] which has good acceleration performance and system efficiency.

However, electrical signal based ANN accelerators with NoC architecture still face the challenges of energy consumption and time delay. To break the communication bottleneck, recent advances in CMOS-compatible optical devices have suggested that optical networks on chip (ONoC) could be a promising solution [41]. In contrast to electrical NoC, the data is is transmitted in the optical domain via waveguides, which has lower power consumption and higher performance than electrical signal transmission. This makes ONoC uniquely capable of performing data-intensive and high-throughput off/on-chip communications, in which needs a huge data movement among processing units or chips to accelerate parallel processing of ANN workloads.

4.1. Off-Chip Communication for Neural Network Accelerators

The research on photonic interconnection in datacenter has a long history. To improve communication performance, exist works have showed the improvement of communica-

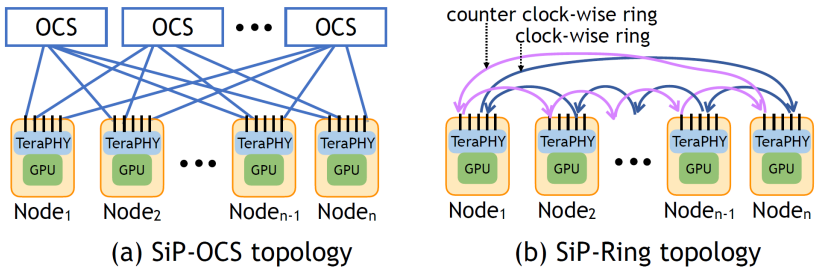


Fig. 7. Two topologies for SiP-ML proposed by [33].

tion performance in datacenter networks by designing photoelectric hybrid connection in reconfigurable topologies [24,47] or designing all-optical interconnects [3,11]. Whereas, compared with the optical technology in computing implementation [79], only a handful of researches focus on using photonic interconnection to accelerate ANNs communication. Authors in [33] presented all photonic interconnects for ANNs acceleration named Silicon Photonic Machine Learning (SiP-ML) which has powerful scalability of ANN training jobs by using SiP chips. The authors argued that ANN training jobs are predictable and periodical that include mostly large data transfers instead of unpredictable behavior and short data flow workloads in conventional datacenter. Authors considered the parallel demand of ANN workloads and the singleness and repeatability of transmission in workloads, and then explored two data-reusing topologies. As illustration of Fig. 7, an Optical Circuit Switch (OCS) based topology called SiP-OCS is designed with commercially available optical switches. Each OCS is linked to each GPUs by port in a flat topology. By setting a 10ms reconfiguration delay, Sip-OCS can transfer data infrequently across the ANN workloads. Furthermore, a switch-free topology without any switching elements was proposed by embedding MRRs in SiP ports named Sip-Ring. As a filter, the activated MRR enables to replace the switch by selecting and forwarding optical signals. Compared to SiP-OCS, SiP-Ring can reuse the signals in non-overlapping portion of MRR, and can reconfigure the resonant wavelength in each port to enrich logically topologies. However, with the increase of process units, the communication costs of large ANN model training increase greatly in SiP-ML, and the communication mapping performance of ANN training is affected by the parallelization strategy and AllReduce.

Motivated by the optical switch based topologies in [33], authors in [70] proposed a co-optimizes network topology and a parallelization strategy for ANN training system named TOPOOPT. The proposed scheme searches over the parallelization strategy space with a fixed topology, and returns the communication demands to the system. A topology is then reconfigured with the searched parallelization strategy, which enables system to alternate between optimizing the parallelization strategy and optimizing the network topology. This looped process helps system to find an optimized parallelization strategy and an optimized topology. The TOPOOPT system is an optical shareable interconnect, in which interfaces of server are connected to the processing unit layer by optical switches. The optical switches enable to achieve the target topology by partitioning the cluster dedicated partitions for each training workload. When the system finds an optimized parallelization

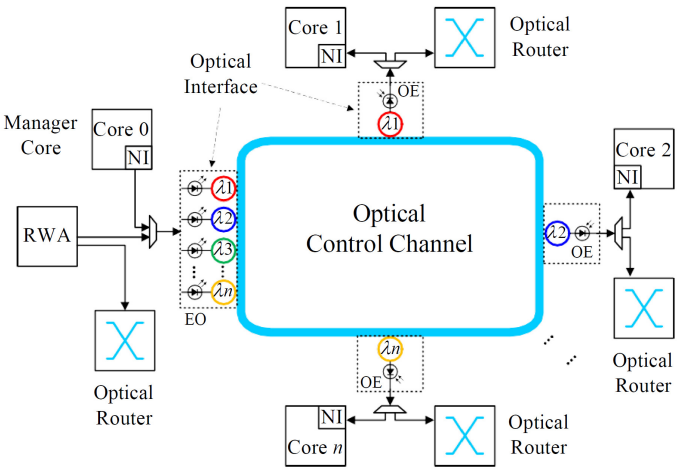


Fig. 8. The optical control channel in [17].

strategy or network topology, interconnection between the server and the processing unit can be changed to the corresponding topology by optical switches reconfiguration.

In addition to all optical network architecture for ANNs workloads, authors proposed a hybrid optical/electrical network architecture that optical switches are employed to offer long-term ANN training communication in [69]. Zhu et al. [83] proposed a silicon photonic reconfigurable architecture with a fat tree topology, which optical switches are applied to link top-of-rack interfaces and aggregate electronic packet interfaces. The optimized optical switches control scheme was designed to reduce the complexity of control implementation, which enables optical switches to apply for large-scale systems integrating. The experiments in hardware test platform show that the silicon photonic reconfigurable architecture can execute ANNs training jobs efficiently. A similar distributed ANN training application with fat topology was proposed in [27], where the experiments were built in commercial rack servers to test the performance of optical switch based distributed ANN acceleration.

Furthermore, authors in [75] proposed an Inter/Intra-Chip silicon photonic network for rack-scale computing systems. To void the challenge of photonic buffering, the architecture employs circuit switching for the ONoC that can also avoid the large overhead in optical devices assembly and unloading. [75] utilized the inter-node interface as the medium to coordinate the request from both local ONoC and optical switch. A channel partition and dynamic path priority control scheme is designed to reduce the control complexity and arbitration overhead. Feng et al. in [25] proposed a variant architecture that is optimized by floorplan optimized delta optical network switch architecture and the preemptive chain feedback scheme. In [43], an arrayed waveguide grating routers based hybrid optical-electrical architecture was proposed, in which a complete bipartite graph is employed to enhance the transmission bandwidth and interconnection scale of machine learning.

4.2. On-Chip Communication for Neural Network Accelerators

In [35], the authors considered that electrical interconnection in the existing manycore platform would not be sustainable for handling the massively increasing bandwidth demand of big data driven AI applications. Hence, a rapid topology generation and core mapping of ONoC (REGO) for heterogeneous multicore architecture was proposed. Based on the genetic algorithm, REGO receives an application task graph including the number of cores and ONoC parameters as inputs, which further includes the available router structure, loss and noise factors of the optical elements. Thus, the REGO can accommodate various router structures and optical elements because it calculates the worst-case OSNR through loss and noise parameters obtained in advance through the parameters of optical.

A fine-grained parallel computing model for ANNs training was depicted in [17] on ONoC, in which the trade-off between computation and communication can be analyzed to support the ANN acceleration. As shown in 8, The optical control channel was designed to configure the state of cores and optical routers. To minimize the total training time, three mapping strategies were designed in each ANN training stage which has the optimal number of cores. The advantages and disadvantages for each mapping strategy are discussed and analyzed in terms of hotspot level, memory requirement and state transitions. Furthermore, an optoelectronic hybrid on chip architecture was demonstrated for CPU and GPU heterogeneous systems in [12], which the first layer is connected by waveguide, the second layer is a electrical mesh with 8×8 nodes, and all layers are linked by the through-silicon-via. The proposed architecture utilized the reservation-based single write multiple reader bus to reduce the number of optical switches that can reduce energy consumption.

Table 2. Challenges and Future research directions

Scalability	Low complexity architectures
	Noise resilient optical devices
	Low loss optical devices
Robustness	Effective photonic crosstalk mitigation
	Phase noise correction
	Noise resilient photodetection
Design Space Exploration	Parallelism of models
	Devices reuse Architectures
	Data reuse topologies
Optical Nonlinear Activation	All-electronic nonlinear activation function
	Photoelectric hybrid nonlinear activation function
	All-optical nonlinear activation function

5. Challenges and Opportunities

In this paper, we reviewed the optical approaches to accelerate neural networks from two aspects, i.e., computing and communication. In recent years, with the maturity of ANN theory and the development of silicon optical technology, one of the areas with growing concerns is the implementations of optical ANNs. Meanwhile, the addition of some sophisticated optical devices, such as optical frequency comb [57], makes it possible for accelerators to train ANN models at extremely high speeds. Nevertheless, there are still some outstanding challenges that limit the inference accuracy, reliability and scalability of optical ANNs. Hence, we summarize the challenges and opportunities to offer suggestions for future research, as shown in Table 2.

Scalability: The exiting works that have been discussed in this review mainly focus on three approaches to accelerate ANNs model training, which are small optical neural network implementation, matrix vector multiplication acceleration and optical network architectures for communication accelerating. The two major issues of the above approaches are high area consumption and energy attenuation of the optical devices. The schemes in [53] and [15] described that the optical depth (the number of MZI units traversed through the longest path) for the unitary matrix is limited to $2N - 3$ and N, in an ANN with N number of neurons, respectively. Therefore, the optical depth of singular value decomposition encoding based ANNs is also limited to $2N - 1$ and $2N + 1$, in which the diagonal matrix is realized by MZIs. The optical depth is positively associated with the number of layers in ANNs that will cause the additional loss as the optical depth increases. The additional loss could exceed the budget and dramatically increase the ratio of signal to noise, which will reduce the computing accuracy in an optical network with limited power consumption. Therefore, more studies are expected to design photonic devices that are noise resilient and low loss to improve scalability for the large scale ANNs, and design novel architectures to reduce the optical integration complexity.

Robustness: With the optical integration scale up, robustness is becoming an important factor for system stability. For example, the performance of MZI-based computing architecture is directly affected by crosstalk, environment temperature and manufacturing process, in which the slight phase change will cause a cascaded calculation error. The experiments in [58] showed that the accuracy in small optical ANN outperform that of large scale optical ANN about 20%. Moreover, the smaller ANN also shows better robustness if added signal noise on optical devices. Whereas the on-chip thermal crosstalk can be suppressed, the finite encoding precision on phase settings will remain as the fundamental limitation for the optical ANNs with high computational complexity. The phase errors, in particular, accumulate when the lightwave signal traverses the MZI mesh with an optical depth of $2N + 1$. In addition, such errors propagate through each layer of the network, which ultimately restricts the depth of the neural network. In order to realize robust photonic accelerator, research is needed to achieve effective photonic crosstalk mitigation, phase noise correction, and noise resilient photodetection.

Design Space Exploration: Early demonstrations of photonic solutions for ANN and RC acceleration were implemented with bulky free-space optics [39], which have strict requirements for accurate phase matching and great difficulty for optical devices footprint reducing. Even in recent optical neural networks based on singular value decomposition encoding, a $m \times n$ weight matrix needs the number of $m(m-1)/2 + n(n-1)/2 + \max(m, n)$ MZIs to realize. This hardware complexity can limit the actually implemen-

tation scale of optical ANNs, especially when the size of an MZI reaches up to 100 μm. Moreover, The extensive use of optical control switches will also cause energy loss in off-chip optical network. Research is thus needed to consider the predictability and periodicity of ANN workloads, parallelism of models, and design architectures or topologies with data and optical devices reusability.

Optical Nonlinear Activation: There are mainly three types for nonlinear activation function implementation in optical ANNs: 1) all electronic nonlinearity, 2) photoelectric hybrid nonlinearity and 3) all-optical nonlinearity. The traditional full electronic nonlinearity receives the weight and output data from the buffer pool. While the photoelectric hybrid way requires the support of optical to electronic converter, and the optical outcomes from modulator will be converted to the electrical results. Examples include semiconductor excitable lasers [19] and electro-absorption modulators [26]. However, the optical-electrical conversion noise and energy loss limit the computing power and expansion of photoelectric hybrid based ANNs. All-optical nonlinear activation functions are still the most promising solutions, which can improve the throughput of ANNs and reduce the latency and power consumption in computation. Currently, the generally used all optical nonlinear activation is saturated absorption of optical materials including monolayer graphene, two photon excitation as well as photonic superlattices [5,56,62]. In addition, the nonlinearities of MRRs can also be designed for nonlinear activation implementation [21]. Whereas, the all optical solutions could reduce the computing accuracy and efficiency of the nonlinear modulation due to the space of nonlinear unit and the speed of devices operational. The implementations of all optical ANNs represent the long term goals, and the hybrid optoelectronic or full electric architectures remain a promising alternative to all-optical networks in the short term.

6. Conclusion

In this paper, we provide a comprehensive survey for optical implementation of ANN accelerators, including photonic computing acceleration and photonic communication acceleration. We first review the fundamental photonic devices that are employed to realize optical accelerator. For the optical neural networks, we present the current ANN accelerators that are realized by the optical effects, including resonance based optical ANN accelerators, diffraction based optical ANN accelerators and interference based optical ANN accelerators. For the optical interconnection, we introduce the existing studies from the perspectives of off-chip communication and on-chip communication for ANN accelerator. Furthermore, we point out the open challenges and the future research opportunities for photonic neural network accelerator, which is expected to provide guidance and insight for future researchers and developers on this research field.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant Nos. 62106052 and 62072118. The authors wish to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities as part of this research.

References

1. Abu-Mostafa, Y.S., Psaltis, D.: Optical neural computers. Scientific American 256(3), 88–95 (1987)
2. Ankit, A., Hajj, I.E., Chalamalasetti, S.R., Ndu, G., Foltin, M., Williams, R.S., Faraboschi, P., Hwu, W.m.W., Strachan, J.P., Roy, K., et al.: Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 715–731 (2019)
3. Ballani, H., Costa, P., Behrendt, R., Cletheroe, D., Haller, I., Jozwik, K., Karinou, F., Lange, S., Shi, K., Thomsen, B., et al.: Sirius: A flat datacenter network with nanosecond optical switching. In: Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. pp. 782–797 (2020)
4. Banerjee, A., Park, Y., Clarke, F., Song, H., Yang, S., Kramer, G., Kim, K., Mukherjee, B.: Wavelength-division-multiplexed passive optical network (wdm-pon) technologies for broadband access: a review. Journal of optical networking 4(11), 737–758 (2005)
5. Bao, Q., Zhang, H., Ni, Z., Wang, Y., Polavarapu, L., Shen, Z., Xu, Q.H., Tang, D., Loh, K.P.: Monolayer graphene as a saturable absorber in a mode-locked laser. Nano Research 4(3), 297–307 (2011)
6. Bhardwaj, R., Saxena, S.B., Sharma, P., Jaiswal, V., Mehrotra, R.: Experimental realisation of parallel optical logic gates and combinational logic using multiple beam interference. Optik 128, 253–263 (2017)
7. Bogaerts, W., De Heyn, P., Van Vaerenbergh, T., De Vos, K., Kumar Selvaraja, S., Claes, T., Dumon, P., Bienstman, P., Van Thourhout, D., Baets, R.: Silicon microring resonators. Laser & Photonics Reviews 6(1), 47–73 (2012)
8. Bueno, J., Maktoobi, S., Froehly, L., Fischer, I., Jacquot, M., Larger, L., Brunner, D.: Reinforcement learning in a large-scale photonic recurrent neural network. Optica 5(6), 756–760 (2018)
9. Bytyn, A., Ahlsdorf, R., Leupers, R., Ascheid, G.: Dataflow aware mapping of convolutional neural networks onto many-core platforms with network-on-chip interconnect. arXiv preprint arXiv:2006.12274 (2020)
10. Chakraborty, I., Saha, G., Sengupta, A., Roy, K.: Toward fast neural computing using all-photonic phase change spiking neurons. entific Reports 8(1) (2018)
11. Chen, L., Chen, K., Zhu, Z., Yu, M., Porter, G., Qiao, C., Zhong, S.: Enabling wide-spread communications on optical fabric with megaswitch. In: 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17). pp. 577–593 (2017)
12. Cheng, T., Wu, N., Yan, G., Zhang, X., Zhang, X.: Poet: A power efficient hybrid optical noc topology for heterogeneous cpu-gpu systems. In: IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society. vol. 1, pp. 3091–3095. IEEE (2019)
13. Chi, P., Li, S., Xu, C., Zhang, T., Zhao, J., Liu, Y., Wang, Y., Xie, Y.: Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory. ACM SIGARCH Computer Architecture News 44(3), 27–39 (2016)
14. Choi, H., Park, S.: A survey of machine learning-based system performance optimization techniques. Applied Sciences 11(7), 3235 (2021)
15. Clements, W.R., Humphreys, P.C., Metcalf, B.J., Kolthammer, W.S., Walmsley, I.A.: Optimal design for universal multiport interferometers. Optica 3(12), 1460–1465 (2016)
16. Crawley, D., Nikolic, K., Forshaw, M.: 3D Nanoelectronic Computer Architecture and Implementation. CRC Press (2020)
17. Dai, F., Chen, Y., Zhang, H., Huang, Z.: Accelerating fully connected neural network on optical network-on-chip (onoc). arXiv preprint arXiv:2109.14878 (2021)

18. De Lima, T.F., Peng, H.T., Tait, A.N., Nahmias, M.A., Miller, H.B., Shastri, B.J., Prucnal, P.R.: Machine learning with neuromorphic photonics. Journal of Lightwave Technology 37(5), 1515–1534 (2019)
19. De Lima, T.F., Shastri, B.J., Tait, A.N., Nahmias, M.A., Prucnal, P.R.: Progress in neuromorphic photonics. Nanophotonics 6(3), 577–599 (2017)
20. De Marinis, L., Cococcioni, M., Castoldi, P., Andriolli, N.: Photonic neural networks: A survey. IEEE Access 7, 175827–175841 (2019)
21. Denis-Le Coarer, F., Sciamanna, M., Katumba, A., Freiberger, M., Dambre, J., Bienstman, P., Rontani, D.: All-optical reservoir computing on a photonic chip using silicon-based ring resonators. IEEE Journal of Selected Topics in Quantum Electronics 24(6), 1–8 (2018)
22. Fang, M.Y.S., Manipatruni, S., Wierzynski, C., Khosrowshahi, A., DeWeese, M.R.: Design of optical neural networks with component imprecisions. Optics express 27(10), 14009–14029 (2019)
23. Farhi, E., Neven, H.: Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002 (2018)
24. Farrington, N., Porter, G., Radhakrishnan, S., Bazzaz, H.H., Subramanya, V., Fainman, Y., Papen, G., Vahdat, A.: Helios: a hybrid electrical/optical switch architecture for modular data centers. In: Proceedings of the ACM SIGCOMM 2010 Conference. pp. 339–350 (2010)
25. Feng, J., Wang, Z., Wang, Z., Chen, X., Chen, S., Zhang, J., Xu, J.: Scalable low-power high-performance rack-scale optical network. In: 2019 IEEE/ACM Workshop on Photonics-Optics Technology Oriented Networking, Information and Computing Systems (PHOTONICS). pp. 1–6. IEEE (2019)
26. George, J.K., Mehrabian, A., Amin, R., Meng, J., De Lima, T.F., Tait, A.N., Shastri, B.J., El-Ghazawi, T., Prucnal, P.R., Sorger, V.J.: Neuromorphic photonics with electro-absorption modulators. Optics express 27(4), 5181–5191 (2019)
27. Glick, M., Wu, Z., Yan, S., Zhu, Z., Bergman, K.: Flexible optical interconnects for efficient resource utilization and distributed machine learning training in disaggregated architectures. In: Proc. of SPIE Vol. vol. 12027, pp. 1202703–1 (2022)
28. Hamerly, R., Bernstein, L., Sludds, A., Soljačić, M., Englund, D.: Large-scale optical neural networks based on photoelectric multiplication. Physical Review X 9(2), 021032 (2019)
29. Heck, M.J., Bowers, J.E.: Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources. IEEE Journal of Selected Topics in Quantum Electronics 20(4), 332–343 (2013)
30. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation 18(7), 1527–1554 (2006)
31. Katumba, A., Freiberger, M., Laporte, F., Lugnan, A., Sackesyn, S., Ma, C., Dambre, J., Bienstman, P.: Neuromorphic computing based on silicon photonics and reservoir computing. IEEE Journal of Selected Topics in Quantum Electronics 24(6), 1–10 (2018)
32. Khan, K., Pasricha, S., Kim, R.G.: A survey of resource management for processing-in-memory and near-memory processing architectures. Journal of Low Power Electronics and Applications 10(4), 30 (2020)
33. Khani, M., Ghobadi, M., Alizadeh, M., Zhu, Z., Glick, M., Bergman, K., Vahdat, A., Klenk, B., Ebrahimi, E.: Sip-ml: high-bandwidth optical network interconnects for machine learning training. In: Proceedings of the 2021 ACM SIGCOMM 2021 Conference. pp. 657–675 (2021)
34. Kim, J.Y., Kang, J.M., Kim, T.Y., Han, S.K.: All-optical multiple logic gates with xor, nor, or, and nand functions using parallel soa-mzi structures: theory and experiment. Journal of Lightwave Technology 24(9), 3392 (2006)
35. Kim, Y.W., Choi, S.H., Han, T.H.: Rapid topology generation and core mapping of optical network-on-chip for heterogeneous computing platform. IEEE Access 9, 110359–110370 (2021)

36. Larger, L., Soriano, M.C., Brunner, D., Appeltant, L., Gutiérrez, J.M., Pesquera, L., Mirasso, C.R., Fischer, I.: Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. Optics express 20(3), 3241–3249 (2012)
37. Lawson, C.L., Hanson, R.J.: Solving least squares problems. SIAM (1995)
38. Li, N., Mahalingavelar, P., Vella, J.H., Leem, D.S., Azoulay, J.D., Ng, T.N.: Solution-processable infrared photodetectors: materials, device physics, and applications. Materials Science and Engineering: R: Reports 146, 100643 (2021)
39. Liang, Y.Z., Liu, H.K.: Optical matrix–matrix multiplication method demonstrated by the use of a multifocus hololens. Optics letters 9(8), 322–324 (1984)
40. Lin, X., Rivenson, Y., Yardimci, N.T., Veli, M., Luo, Y., Jarrahi, M., Ozcan, A.: All-optical machine learning using diffractive deep neural networks. Science 361(6406), 1004–1008 (2018)
41. Liu, F., Zhang, H., Chen, Y., Huang, Z., Gu, H.: Wrh-onoc: A wavelength-reused hierarchical architecture for optical network on chips. In: 2015 IEEE Conference on Computer Communications (INFOCOM). pp. 1912–1920. IEEE (2015)
42. Liu, J., Wu, Q., Sui, X., Chen, Q., Gu, G., Wang, L., Li, S.: Research progress in optical neural networks: theory, applications and developments. PhotoniX 2(1), 1–39 (2021)
43. Lu, Y., Gu, H., Yu, X., Chakrabarty, K.: Lotus: A new topology for large-scale distributed machine learning. ACM Journal on Emerging Technologies in Computing Systems (JETC) 17(1), 1–21 (2020)
44. Luo, Y., Mengu, D., Yardimci, N.T., Rivenson, Y., Veli, M., Jarrahi, M., Ozcan, A.: Design of task-specific optical systems using broadband diffractive neural networks. Light: Science & Applications 8(1), 1–14 (2019)
45. Markram, H., Muller, E., Ramaswamy, S., Reimann, M.W., Abdellah, M., Sanchez, C.A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al.: Reconstruction and simulation of neocortical microcircuitry. Cell 163(2), 456–492 (2015)
46. Mehrabian, A., Al-Kabani, Y., Sorger, V.J., El-Ghazawi, T.: Pcnna: A photonic convolutional neural network accelerator. In: 2018 31st IEEE International System-on-Chip Conference (SOCC). pp. 169–173. IEEE (2018)
47. Mellette, W.M., McGuinness, R., Roy, A., Forencich, A., Papen, G., Snoeren, A.C., Porter, G.: Rotornet: A scalable, low-complexity, optical datacenter network. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication. pp. 267–280 (2017)
48. Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., Marr, D.: Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and asic. In: 2016 International Conference on Field-Programmable Technology (FPT). pp. 77–84. IEEE (2016)
49. O'Connor, I., Nicolescu, G.: Integrated optical interconnect architectures for embedded systems. Springer Science & Business Media (2012)
50. Ohno, S., Toprasertpong, K., Takagi, S., Takenaka, M.: Si microring resonator crossbar array for on-chip inference and training of optical neural network. arXiv preprint arXiv:2106.04351 (2021)
51. Psaltis, D., Brady, D., Wagner, K.: Adaptive optical networks using photorefractive crystals. Applied Optics 27(9), 1752–1759 (1988)
52. Qian, C., Lin, X., Lin, X., Xu, J., Sun, Y., Li, E., Zhang, B., Chen, H.: Performing optical logic operations by a diffractive neural network. Light: Science & Applications 9(1), 1–7 (2020)
53. Reck, M., Zeilinger, A., Bernstein, H.J., Bertani, P.: Experimental realization of any discrete unitary operator. Physical review letters 73(1), 58 (1994)
54. Van der Sande, G., Brunner, D., Soriano, M.C.: Advances in photonic reservoir computing. Nanophotonics 6(3), 561–576 (2017)
55. Sasikala, V., Chitra, K.: All optical switching and associated technologies: a review. Journal of Optics 47(3), 307–317 (2018)
56. Schirmer, R.W., Gaeta, A.L.: Nonlinear mirror based on two-photon absorption. JOSA B 14(11), 2865–2868 (1997)

57. Scott, A., Diddams: The evolving optical frequency comb [invited]. Journal of the Optical Society of America B 27(11), B51–B62 (2010)
58. Shen, Y., Harris, N.C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D., et al.: Deep learning with coherent nanophotonic circuits. Nature Photonics 11(7), 441–446 (2017)
59. Shiflett, K., Karanth, A., Bunescu, R., Louri, A.: Albireo: Energy-efficient acceleration of convolutional neural networks via silicon photonics. In: 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). pp. 860–873. IEEE (2021)
60. Shiflett, K., Wright, D., Karanth, A., Louri, A.: Pixel: Photonic neural network accelerator. In: 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). pp. 474–487. IEEE (2020)
61. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature 529(7587), 484–489 (2016)
62. Soljačić, M., Ibanescu, M., Johnson, S.G., Fink, Y., Joannopoulos, J.D.: Optimal bistable switching in nonlinear photonic crystals. Physical Review E 66(5), 055601 (2002)
63. Sorrentino, T., Quintero-Quiroz, C., Torrent, M., Masoller, C.: Analysis of the spike rate and spike correlations in modulated semiconductor lasers with optical feedback. IEEE Journal of Selected Topics in Quantum Electronics 21(6), 561–567 (2015)
64. Spuesens, T., Liu, L., de Vries, T., Romeo, P.R., Regreny, P., Van Thourhout, D.: Improved design of an inp-based microdisk laser heterogeneously integrated with soi. In: 2009 6th IEEE International Conference on Group IV Photonics. pp. 202–204. IEEE (2009)
65. Stanley, A., Singh, G., Eke, J., Tsuda, H.: Mach–zehnder interferometer: A review of a perfect all-optical switching structure. In: Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing. pp. 415–425. Springer (2016)
66. Tait, A.N., Nahmias, M.A., Shastri, B.J., Prucnal, P.R.: Broadcast and weight: an integrated network for scalable photonic spike processing. Journal of Lightwave Technology 32(21), 4029–4041 (2014)
67. Tait, A.N., Wu, A.X., De Lima, T.F., Zhou, E., Shastri, B.J., Nahmias, M.A., Prucnal, P.R.: Microring weight banks. IEEE Journal of Selected Topics in Quantum Electronics 22(6), 312–325 (2016)
68. Totović, A.R., Dabos, G., Passalis, N., Tefas, A., Pleros, N.: Femtojoule per mac neuromorphic photonics: An energy and technology roadmap. IEEE Journal of selected topics in Quantum Electronics 26(5), 1–15 (2020)
69. Truong, T.N., Takano, R.: Hybrid electrical/optical switch architectures for training distributed deep learning in large-scale. IEICE TRANSACTIONS on Information and Systems 104(8), 1332–1339 (2021)
70. Wang, W., Khazraee, M., Zhong, Z., Jia, Z., Mudigere, D., Zhang, Y., Kewitsch, A., Ghobadi, M.: Topoopt: Optimizing the network topology for distributed dnn training. arXiv preprint arXiv:2202.00433 (2022)
71. Wang, Y.G.: Applications of memristors in neural networks and neuromorphic computing: A review. Int. J. Mach. Learn. Comput 11, 350–356 (2021)
72. Xia, C., Chen, Y., Zhang, H., Zhang, H., Wu, J.: Photonic computing and communication for neural network accelerators. In: International Conference on Parallel and Distributed Computing: Applications and Technologies. pp. 121–128. Springer (2022)
73. Xiang, S., Wen, A., Pan, W.: Emulation of spiking response and spiking frequency property in vcsel-based photonic neuron. IEEE Photonics Journal 8(5), 1–9 (2016)
74. Xu, R., Lv, P., Xu, F., Shi, Y.: A survey of approaches for implementing optical neural networks. Optics & Laser Technology 136, 106787 (2021)
75. Yang, P., Pang, Z., Wang, Z., Wang, Z., Xie, M., Chen, X., Duong, L.H., Xu, J.: Rson: An inter/intra-chip silicon photonic network for rack-scale computing systems. In: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1369–1374. IEEE (2018)

76. Yao, Z., Wu, K., Tan, B.X., Wang, J., Li, Y., Zhang, Y., Poon, A.W.: Integrated silicon photonic microresonators: emerging technologies. IEEE Journal of Selected Topics in Quantum Electronics 24(6), 1–24 (2018)
77. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., Cong, J.: Optimizing fpga-based accelerator design for deep convolutional neural networks. In: Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays. pp. 161–170 (2015)
78. Zhang, H., Gu, M., Jiang, X., Thompson, J., Cai, H., Paesani, S., Santagati, R., Laing, A., Zhang, Y., Yung, M., et al.: An optical neural chip for implementing complex-valued neural network. Nature Communications 12(1), 1–11 (2021)
79. Zhang, Q., Yu, H., Barbiero, M., Wang, B., Gu, M.: Artificial neural networks enabled by nanophotonics. Light: Science & Applications 8(1), 1–14 (2019)
80. Zhao, Y., Zhao, H., Lv, R.q., Zhao, J.: Review of optical fiber mach–zehnder interferometers with micro-cavity fabricated by femtosecond laser and sensing applications. Optics and Lasers in Engineering 117, 7–20 (2019)
81. Zhao, Z., Gu, J., Ying, Z., Feng, C., Chen, R.T., Pan, D.Z.: Design technology for scalable and robust photonic integrated circuits: Invited paper. In: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 1–7 (2019)
82. Zhou, Z., Yin, B., Michel, J.: On-chip light sources for silicon photonics. Light: Science & Applications 4(11), e358 (2015)
83. Zhu, Z., Teh, M.Y., Wu, Z., Glick, M.S., Yan, S., Hattink, M., Bergman, K.: Distributed deep learning training using silicon photonic switched architectures. APL Photonics 7(3), 1–11 (2022)
84. Zuo, Y., Li, B., Zhao, Y., Jiang, Y., Chen, Y.C., Chen, P., Jo, G.B., Liu, J., Du, S.: All-optical neural network with nonlinear activation functions. Optica 6(9), 1132–1137 (2019)

Chengpeng Xia received B.E. degree from Lanzhou Jiaotong University in 2017 and M.E. degree from Guangdong University of Technology in 2020. He is now working toward Ph.D. degree in Computer Science from University of Otago. His main research interests include optical computing, optical neural network accelerator and distributed computing.

Yawen Chen (Member, IEEE) received the PhD degree in computer science from the University of Adelaide, Adelaide, Australia, in 2008. She is a senior lecturer at the University of Otago in New Zealand. Her research interests include resource optimization and performance evaluation in computer networking and computer architecture (optical network-on-chips, interconnection network, wired, and wireless networking).

Haibo Zhang (Senior Member, IEEE) received the PhD degree from the University of Adelaide, Adelaide, Australia, in 2009. From 2009 to 2010, he was a postdoctoral research associate with the Automatic Control Lab, Royal Institute of Technology in Sweden. He is currently a senior lecturer at the Department of Computer Science, University of Otago, New Zealand. His current research interests include wireless communication, optical network-on-chips, wireless body sensor networks, protocol design.

Hao Zhang received B.E. degree and M.E. degree from Shandong University of Science and Technology in 2017 and 2020, respectively. He is now working toward Ph.D. degree in Computer Science from University of Otago. His main research interests include optical network on chip, optical computing and parallel computing.

Fei Dai obtained BSc of computer science and MSc of software engineering in 2014 and 2019 from Guilin University of Technology, China. He is currently working towards the PhD degree in computer science at University of Otago, New Zealand. His research interests include optical interconnect communications, multicore architecture, parallel computation, deep learning accelerators, IoT system, etc.

Jigang Wu received B.Sc. degree from Lanzhou University, and Ph.D. degree from the University of Science and Technology of China. Now, he is distinguished professor of School of Computer Science and Technology, Guangdong University of Technology. His research interests include network computing and machine learning.

Received: January 31, 2022; Accepted: December 10, 2022.

Computer Science and Information Systems 20(1):537–550 https://doi.org/10.2298/CSIS220131067G

Human Action Recognition Based on Skeleton Features

Yi Gao[1*], Haitao Wu[1*], Xinmeng Wu[1], Zilin Li[1], and Xiaofan Zhao[2*]

[1] College of Intelligence and Computing,
Tianjin University, Tianjin, China,
gaoyi_art@tju.edu.cn
[2] School of Information Technology and Cyber Security,
People's Public Security University of China, Beijing, China
zhaoxiaofan@ppsuc.edu.cn

Abstract. Based on human bone joints, skeleton information has clear and simple features and is not easily affected by appearance factors. In this paper, an improved feature of Gist, ExGist, is proposed to describe the skeleton information of human bone joints for human action recognition. The joint coordinates are extracted by using OpenPose and the thermodynamic diagram, and ExGist is used for feature extraction. The advantage of ExGist is that it can effectively characterize the local and global features of skeleton information while maintaining the original advantages of Gist feature. Compared with Gist, ExGist achieves better results on different classifiers. Additionally, compared with C3D and APTNet, our model also obtains better results with an accuracy rate of 89.2%.

Keywords: Human Action Recognition, Gist, OpenPose, Euclidean Distance, Thermodynamic Diagram.

1. Introduction

Human motion recognition which captures the changing process of human motion by transforming the original video sequence has been one of the research focuses in the field of computer vision for a long time. The key to motion recognition is extracting the features that can represent human motion information from the region where the moving object is located. Many researchers have proposed a large number of technologies and methods based on feature representation. According to the feature extraction methods, human action recognition can be classified into manual feature extraction and feature extraction based on deep learning.

Manual Feature Extraction: The human action recognition method based on manual feature extraction firstly samples the continuous frames of data and obtains the sampling points. According to the designed manual feature extraction method, the features of the sampling points are extracted, which are encoded into feature vectors. Then the encoded feature vectors are input into the behavior classifier for training. Finally, the manual feature vectors extracted from the test video are input into the trained classifier to obtain the classification results. By using the method, researchers from all over the world have proposed gait recognition 1, silhouette 2, human junction

[*] These authors contributed equally to this work and should be considered co-first authors.

34, space-time interest points 56, movement trajectory 78 and other human behavior recognition methods.

Deep Learning: The human action recognition method based on deep learning uses a trainable feature extraction model to automatically learn behavior representations from videos in an end-to-end manner to complete classification. Up to now, the network structure of action recognition methods based on deep learning mainly includes convolutional neural network (CNN) 91011, cyclic neural network (RNN) 1213, graph convolution neural network 1415 and hybrid network 161718. Other researchers have proposed Restricted Boltzmann Machine 19, recurrent neural network 20, independent subspace analysis 21, etc., which also got good results.

There have been some researches regarding to Gist. A static human behavior classification method that combined local constraint linear coding (LLC) and global feature descriptor Gist was proposed 23. This method was limited to the processing of static human behavior images, and did not apply to the field of videos. In addition, another new combined feature called global Gist feature and local patch coding was also proposed 24: Gist feature included the spectrum information of the action in the global view. Then, according to the frequency of the action variance, Gist feature located in different grids in the action center area was divided into four blocks and local patch coding was adopted. What's more, a new method was proposed for recognizing person-to-person interaction behavior based on the statistical features of key frame feature library 25. This method firstly extracted the global Gist and the regional HOG features, and then used K-means clustering algorithm to construct the key frame feature library corresponding to the action category. At the same time, according to the similarity measure, the frequency of each key frame in the feature library in the interactive video was counted, and a statistical histogram feature representation of the action video was obtained, which was trained for classification and recognition. The calculation of the above three methods was complex and the accuracy rate was relatively low.

In this paper, to describe the skeleton information of human bones and identify human behavior, a new feature descriptor based on the improved Gist 22 is proposed, named ExGist, which extends the classic feature descriptor Gist for scene understanding into the field of video processing. OpenPose 26 and thermodynamic diagram 27 are used to extract joint coordinates, then ExGist is used for feature extraction. Finally, the extracted features are input into the classifiers for classification. According to the experimental results, when SVM 28 is used as the classifier, the accuracy of this method is as high as 89.2%. Compared with C3D 29 and APTNet 30 models, our method has achieved a better result.

The main contributions are as follows:

1. A new feature ExGist based on Gist is proposed. After being compared on different classifiers, ExGist achieves better results than Gist.
2. A new human action recognition method based on the ExGist feature description is proposed. This method is not only superior to some current research methods of human action recognition using Gist, but also has a better recognition accuracy than C3D and APTNet models.
3. This method shows that the classical feature Gist can apply to not only the field of scene understanding but also the field of video processing. The improved version of Gist achieves good results, providing new ideas and methods for more related researches.

2. Related Work

2.1. Skeleton-based Human Action Recognition by the Integration of Euclidean distance

In our previous work, a skeleton-based model was presented for human action recognition 31. Firstly, Euclidean distance was combined with OpenPose and thermodynamic diagram to estimate human poses. Additionally, the attention mechanism was integrated into the human pose estimation process, to gain both the overall features and the partial features. Finally, MLP classifier was used for classification. This method could also be applied to real-time recognition of multi-person behavior. On the KTH and ICPR datasets, the accuracy of the mode verified by changing several parameters was tested. The highest accuracy rate of single-person behavior recognition was 0.821, and the highest accuracy rate of multi-person behavior recognition was 0.812. The high running speed enabled the mode to be a real-time model.

On the basis of previous work, the following improvements are made in this paper: Firstly, a new feature extraction method ExGist is proposed by fusing the global feature descriptor Gist. ExGist makes up for a lack of the local feature description of Gist and achieves unexpected good results in the performance comparison of different classifiers. Secondly, the data sets UCF10132 and HMDB5133 are selected for their more abundant action categories and wider application.

2.2. Multi-person Pose Estimation

There are two prominent algorithms for multi-person pose estimation. The first one is AlphaPose 34, where the human body on the graph is detected first, and then the key points and skeleton of each human body are obtained (top-down). Another algorithm is OpenPose 26, where the key position is gained firstly, and then different human skeletons are distinguished by the correlation between joint points(bottom-up). AlphaPose is more accurate, but as the number of people on the picture increases, the speed slows down. The accuracy rate of OpenPose is lower, but the speed is not affected by the number of people on the graph. In order to ensure real-time performance, OpenPose is adopted in our work. However, OpenPose is easy to introduce the interference of non-human objects, resulting in the confusion of detection results. Additionally, thermodynamic diagram 27 is used to obtain the number of people on the picture, and then establish the corresponding partition. In the image processing of target detection, thermodynamic diagram represents the key points by using two-dimensional Gaussian kernel.

2.3. Multi-person Pose Estimation

Support Vector Machine (SVM) 28is often used for classification problems and is widely used in pedestrian monitoring as a feature classifier. SVM classifies features by solving the maximum margin hyperplane. Kernel used in our experiment includes RBF, Linear and Poly.

Multilayer Perceptron (MLP) 35is a fully connected multi-layer neural network, which is a supplement of feed forward neural network. It generally consists of three types of layers: the input layer, output layer and hidden layer. The input layer receives the input signal to be processed. The required tasks such as prediction and classification are performed by the output layer. An arbitrary number of hidden layers that are placed between the input and output layer are the true computational engine of the MLP. If it has more than one hidden layer, it is called Artificial Neural Network (ANN). MLPs are designed to approximate any continuous function and can solve problems which are not linearly separable. The major applications of MLP are pattern classification, recognition, prediction and approximation.

K-Nearest Neighbors (KNN)36 firstly calculates the distance between the training set and the sample (L2 Norm is generally used). Then, k pieces of data closest to the sample are selected. The category that contains the most selected points is the predicted category. It is worth noting that the larger the k value is, the larger the approximate error will be. When the k value is small, the overfitting will occur.

Decision tree 37 and random forest 38 are also commonly used classification methods. Although decision trees are common supervised learning algorithms, they are prone to bias and overfitting. However, when multiple decision trees form an ensemble in the random forest algorithm, they will predict more accurate results, especially when the individual trees are not correlated with each other. The random forest algorithm is an extension of the bagging method in that it exploits bagging and feature randomness to create unrelated decision tree forests. Feature randomness, also known as feature bagging or the random subspace approach, generates random subsets of features, thus ensuring low correlation among decision trees. This is the key difference between decision trees and random forests. While decision trees consider all possible feature segmentation, random forests only select a subset of these features.

3. Model

3.1. Gist Feature Descriptor

Gist 2239is a low-dimensional signature vector of a scene, representing global feature information, and is often used for feature extraction of scene recognition and classification tasks. To extract the global features of Gist, the image needs to be divided into several grids of equal size, and then Gabor filters with different direction scales are used to correlate these grids. Finally, the calculation results of these grid regions are

averaged ions are averaged to obtain the required feature information. The steps are as follows:

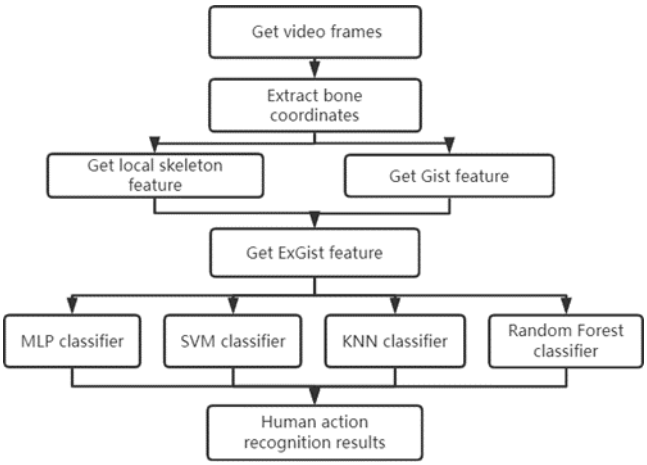


Fig. 1. Flow chart of feature extraction

Assume that the original gray-scale image to be processed is $I(x, y)$, and its size is $M \times N$. It is first divided into $n \times n$. Each net block represents an area, and $n_g = n_b \times n_b$ is used to record the total number of net blocks. Each of the mesh blocks after the image division is marked with B_i, where $I = 1 \ldots g$. In order to simplify the calculation and processing, each area is the same size, and its size is $M' \times N'$.

Gabor filter has great similarity with human visual perception function. By changing the mother wavelet of the filter, different Gabor filters can be obtained by some mathematical operations. The mother wavelet of Gabor filter is expressed as follows:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\left(x^2/{\sigma_x}^2 + x^2/{\sigma_y}^2\right)\right] * \cos(2\pi f_0 x + \varphi). \quad (1)$$

Where, x and y represent coordinate information of pixel points, σ_x and σ_y represent Gaussian standard deviation of x-axis and y-axis respectively, f_0 represents center frequency and φ represents phase shift.

A group of Gabor filters with different scales and directions can be obtained by corresponding mathematical processing on the mother wavelet. The specific calculation formula is as follows:

$$g_{mn}(x, y) = a^{-m} g(x', y'), a > 1. \quad (2)$$

$$x' = a^{-m}(x\cos(\theta + y\sin\theta). \quad (3)$$

$$y' = a^{-m}(-x\sin(\theta + y\cos\theta). \quad (4)$$

$\theta = \frac{n\pi}{n+1}$ represents the rotation angle, a^{-m} represents the scale factor. m represents the scale number, and n represents the direction number.

The Gabor filters obtained by calculation firstly implement the same processing for the different regions divided in the original image, and then use cascade operation to obtain the block Gist features of the image, as follows:

$$G_i^B = cat\big(I(x,y) * g_{mn}(x,y)\big), (x,y) \in B_i. \quad (5)$$

Where, G^B represents the gist feature of the block, the dimension is $m \times n \times M' \times N'$, and cat represents the concatenation operation, and $*$ represents the convolution operation. For each different filter, average the obtained block gist features, and then integrate the calculation results by line to obtain the gist global features of the image, which are shown as follows:

$$G = \left(\overline{G_1^B}, \overline{G_2^B}, \ldots \overline{G_n^B}\right). \quad (6)$$

Gist global feature is to describe an image as a whole. We use the corresponding feature operator to extract the features of the image, and record the relevant category information with the calculated multi-dimensional features. In the whole process, there is no need to consider much complex local information, which can reduce the impact of some small noises on clustering and reduce the additional errors caused by unnecessary processing.

3.2. ExGist Feature Descriptor

In order to classify video actions, features from video frames should be extracted. Both static and dynamic features based on skeleton information are proposed to represent video actions. Firstly, skeleton information is extracted from these frames using OpenPose. Then, feature extraction is carried out for skeleton information, which is mainly divided into two categories, namely static feature and dynamic feature. After that, these features are normalized.

Firstly, the matrix X is defined to represent the coordinates of the nodes between different frames. Most methods directly use X as the feature, and some improvements have been made for the matrix X.

$$X_i = [(x_{i1}, y_{i1}), \ldots (x_{in}, y_{in})]. \quad (7)$$

$$X = \{x_1, x_2, \ldots x_n\}. \quad (8)$$

Then the matrix X are normalized to adapt to images of different sizes, so as to represent features effectively. Inspired by the spatial partitioning method of ST-GCN40, the nodes are divided into different subsets to represent height, arm span and stationary center of gravity, respectively. In ST-GCN, a domain of nodes is divided into three subsets. The first subset is a node further away from the whole skeleton than the root node. Another subset is a node closer to the center, and the third subset is the root node itself, which is used to represent the motion characteristics of centrifugal motion,

centripetal motion and stationary motion respectively. This strategy is to add "weight" to the key parts of rapid and accurate recognition. In this paper, data refers to the coordinates of skeletons ranging from 50 to 400, which is larger than expected. In that case, the distance from node 1, the head, to the barycenter is used as the height H, which will be used in the normalization. Node 8 and node 11 are the two sides of human's waist, so the average coordinate of node 8 and node 11 is used as the coordinate of barycenter.

$$H = \sqrt{(x_1 - x_b)^2 + (y_1 - y_b)^2}. \quad (9)$$

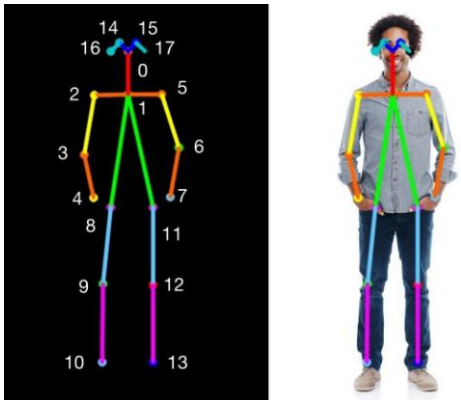


Fig. 2. Spanning tree of human posture features and mapping relation diagram of human joints.

Thirdly, the process of transforming the coordinates into the index of action label is challenging. In that case, the height H is used to converge the unwieldy data into normalized matrix X. The coordinates of some nodes in skeleton graph are less relevant to the accuracy of the action frame to be tested, such as node 0 and node 1. As a result, we decide to subtract the coordinates of these nodes.

$$X_i = [(x_{i1}/H, y_{i1}/H), \dots (x_{in}/H, y_{in}/H)]. \quad (10)$$

$$X_i = [(d_{i0} - d_{i0}, t_{i0} - t_{i0}), \dots (d_{in} - d_{i0}, t_{in} - t_{i0})]. \quad (11)$$

Then the normalized matrix X is obtained, which includes the coordinates of frame. Up to now, features from the normalized matrix X are extracted. Some parts of the skeleton features are gained at first by means of ST-GCN. The data of the previous frame in some parts of skeleton is subtracted from the data of the next frame, except for the first frame because there is no frame before the first one. As the frames are continuous, the data obtained by subtracting the normalized data can reflect an action. Additionally, it should be extracted as a symbol of one action. Now that the frames are continuous, this work should be done as soon as the next normalized data arrives.

$$Y = X_{i=1}^{n}[i + step][0:3] - X_{i=1}^{n}[i][0:3]. \quad (12)$$

Then the data of all nodes in the next frame is subtracted from the data in the previous frame.

$$Z = X_{i=1}^{n}[i + step][:] - X_{i=1}^{n}[i][:]. \quad (13)$$

Up to now, two parts of features have been gained. One part is coordinates, and another is the relationship among continuous frames. Matrix F is used to store all the features.

$$F = [X, Y(10), Z]. \quad (14)$$

Additionally, Euclidean distance D between joints is calculated and is stored as a characteristic.

$$F = [X, Y(10), Z, D]. \quad (15)$$

The compactness of skeleton C and the rate of change of compactness between different frames $\triangle C$ are defined.

$$F = [X, Y(10), Z, D, C, \Delta C]. \quad (16)$$

In addition, three-dimensional features are used to characterize the degree of skeleton integrity, which represent the degree of upper body integrity, lower body integrity and face integrity respectively.

$$S = [s_1, s_2, s_3]. \quad (17)$$

$$F = [X, Y(10), Z, D, C, \Delta C, S, G]. \quad (18)$$

ExGist adds some more detailed local features on the basis of Gist. The 18 human bone points are divided into three groups, namely the facial bone points, the upper body bone points and the lower body bone points. According to the coordinates of bone points from each group, the geometric characteristics of human movements and their changes in the dimension of time are obtained.

3.3. Defective Skeleton Graph

When extracting a skeleton feature, the model collects skeleton coordinates, partial features, overall features and Euclidean distances, all of which need a complete skeleton graph. Actually, in real-world applications, the skeleton graphs are more likely to be defective. In that case, firstly, a standard skeleton graph is prepared, from which all the joint coordinate pairs can be acquired. Then, when the picture of human body is incomplete, previous joint coordinates can fill the vacancy. As a result, all the obtained skeleton graphs are integrated.

3.4. Multi-person Pose Estimation

MLP Classifier, a supervised learning algorithm, is adopted in the model. It's divided into three types of layers named input layer, hidden layer and output layer. LBFGS and stochastic gradient descent are used to optimize the logarithmic loss function. MLP Classifier performs iterative training, because the partial derivative of the loss function is calculated at each step when the parameters are updated. The model uses the sigmoid function and the tanh function to activate.

$$G(a) = softmax(a). \quad (19)$$

$$f(x) = G\left(b^{(2)} + W^{(2)}\left(s\left(b^{(1)} + W^{(1)}x\right)\right)\right). \quad (20)$$

Different strides and learning-rate are used to obtain a better model. Weights can be initialized in the model, and the function is used to calculate the probability when testing the action label of a picture.

4. Experiments

4.1. Datasets

HMDB51. HMDB5133 is a large collection of realistic videos from various sources, including movies and web videos. The dataset is composed of 6,849 video clips from 51 action categories (such as "jump", "kiss" and "laugh"), with each category containing at least 101 clips. The original evaluation scheme uses three different training/testing splits. In each split, each action class has 70 clips for training and 30 clips for testing. The average accuracy over these three splits is used to measure the final performance.

UCF101. UCF10132 is an extension of UCF50 and consists of 13,320 video clips, which are classified into 101 categories. These 101 categories can be classified into 5 types (Body motion, Human-human interactions, Human-object interactions, Playing musical instruments and Sports). The total length of these video clips is over 27 hours. All the videos are collected from YouTube and have a fixed frame rate of 25 FPS with the resolution of 320 × 240.

Fig. 3. HMDB51 dataset presentation

Fig. 4. UCF101 dataset presentation

Due to the large amount of data, the sub-set of HMDB51 and UCF101 is selected for the experiment, and the training set and test set are randomly divided according to 7:3.

It can be seen that the accuracy of ExGist proposed by us is much higher than the accuracy of Gist on all the classifiers. Especially, when ExGist feature is extracted and put into SVM classifier, the accuracy can reach to 0.892. In conclusion, the experimental results show that ExGist can effectively represent human actions, thus improving the accuracy of action classification task.

Compared with C3D and APTNet, our model also obtains better results with a higher accuracy rate. In that case, the feature ExGist and SVM classifier can achieve relatively good human action recognition results.

The location information of bone joint extracted by OpenPose is clear and simple, which can make up for the shortage of global feature description in Gist. When using Gist to extract the global features of bone joint nodes, the method inevitably ignores the local features and connections of joints in the extraction of bone joint information. Therefore, it is necessary to improve and supplement more local features on the basis of global features of Gist. The coordinate information of joint nodes, the position relation of different joint nodes and the change rate between different frames are taken as local information to supplement Gist, so as to obtain ExGist. ExGist can not only give full play to the advantages of global feature description of Gist, but also greatly avoid the problems of high global feature dimensions and large computation.

Fig. 5. HMDB51 dataset performance of the model

Table 1. Results for different classifiers

Classifier	Feature	Accuracy
SVM(poly)	Gist	0.771
SVM(poly)	ExGist	0.892
SVM(rdf)	Gist	0.751
SVM(rdf)	ExGist	0.847
SVM(linear)	Gist	0.747
SVM(linear)	ExGist	0.851
MLP(lbfgs)	Gist	0.755
MLP(lbfgs)	ExGist	0.871
MLP(adam)	Gist	0.747
MLP(adam)	ExGist	0.871
KNN	Gist	0.712
KNN	ExGist	0.847
Random Forest	Gist	0.723
Random Forest	ExGist	0.863
Decision Tree	Gist	0.591
Decision Tree	ExGist	0.755
AdaBoost	Gist	0.703
AdaBoost	ExGist	0.795
Gaussian Naive Bayes	Gist	0.618
Gaussian Naive Bayes	ExGist	0.6797
Linear Discriminant	Gist	0.651
Linear Discriminant	ExGist	0.759

Table 2. Results for different models

Model	Accuracy
C3D	0.838
APTNet	0.872
ExGist+SVM	0.891

5. Conclusions

ExGist, an improved feature of Gist, is proposed to describe the skeleton information for human action recognition. Firstly, OpenPose is combined with thermodynamic diagram to estimate human poses and get skeleton coordinates. Additionally, ExGist is used to gain both the global features and the local features. Finally, MLP, KNN, SVM and other classifiers are used for classification. This method can also be applied to real-time recognition of multi-person behavior. After being tested on two data sets, our improvement proves to be of great help to improve the accuracy rate of behavior recognition.

References

1. Vinay Kukreja, Deepak Kumar, and Amandeep Kaur. Deep learning in human gait recognition: An overview. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pages 9–13. IEEE, 2021.
2. Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space- time shapes. IEEE transactions on pattern analysis and machine intelligence, 29(12):2247– 2253, 2007.
3. Xiaodong Yang and YingLi Tian. Effective 3d action recognition using eigenjoints. Journal of Visual Communication and Image Representation, 25(1):2–11, 2014.
4. Faisal Mehmood, Enqing Chen, Muhammad Azeem Akbar, and Abeer Abdulaziz Alsanad. Human action recognition of spatiotemporal parameters for skeleton sequences using mtln feature learning framework. Electronics, 10(21):2708, 2021.
5. Ivan Laptev. On space-time interest points. International journal of computer vision, 64(2):107–123, 2005.
6. Konstantinos Rapantzikos, Yannis Avrithis, and Stefanos Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 1454–1461. Ieee, 2009.
7. Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In Proceedings of the IEEE international conference on computer vision, pages 3551–3558, 2013.
8. Jogendra Nath Kundu, Maharshi Gor, Phani Krishna Uppala, and Venkatesh Babu Radhakrishnan. Unsupervised feature learning of human actions as trajectories in pose embedding manifold. In 2019 IEEE winter conference on applications of computer vision (WACV), pages 1459–1467. IEEE, 2019.
9. Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1725–1732, 2014.
10. Kyo-Min Hwang and Sang-Chul Kim. A study of cnn-based human behavior recognition with channel state information. In 2021 International Conference on Information Networking (ICOIN), pages 749–751. IEEE, 2021.
11. SH Basha, Viswanath Pulabaigari, and Snehasis Mukherjee. An information-rich sampling technique over spatio-temporal cnn for classification of human actions in videos. Multimedia Tools and Applications, pages 1–19, 2022.
12. Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In Proceedings of the ieee conference on computer vision and pattern recognition, pages 5308–5317, 2016.

13. Pankaj Khatiwada, Matrika Subedi, Ayan Chatterjee, and Martin Wulf Gerdes. Automated human activity recognition by colliding bodies optimization-based optimal feature selection with recurrent neural network. arXiv preprint arXiv:2010.03324, 2020.
14. Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Thirty-second AAAI conference on artificial intelligence, 2018.
15. Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3595–3603, 2019.
16. Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. Recurrent convolutional network for video-based person re-identification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1325–1334, 2016.
17. Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In International conference on machine learning, pages 843–852. PMLR, 2015.
18. Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1725–1732, 2014.
19. Graham W Taylor and Geoffrey E Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In Proceedings of the 26th annual international conference on machine learning, pages 1025–1032, 2009.
20. Wan-Jin Yu, Zhen-Duo Chen, Xin Luo, Wu Liu, and Xin-Shun Xu. Delta: A deep dual-stream network for multi-label image classification. Pattern Recognition, 91:322–331, 2019.
21. Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real- time object recognition. In 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 922–928. IEEE, 2015.
22. Antonio Torralba, Kevin P Murphy, William T Freeman, and Mark A Rubin. Context-based vision system for place and object recognition. In Computer Vision, IEEE International Conference on, volume 2, pages 273–273. IEEE Computer Society, 2003.
23. Ende Wang, Qiaoying Liu, and Li Yong. Classification of static human behaviors based on llc and gist features. Computer Engineering, 44(8):268–272, 2018.
24. Yangyang Wang, Yibo Li, and Xiaofei Ji. Human action recognition based on global gist feature and local patch coding. International Journal of Signal Processing, Image Processing and Pattern Recognition, 8(2):235–246, 2015.
25. Xiaofei Ji and Xinmeng Zuo. Couple interaction behavior recognition based on static features of key-frame feature library. Computer Application, 36(8):2287–2291, 2016.
26. Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7291–7299, 2017.
27. Kazumasa Tsutsui and Koji Moriguchi. A computational experiment on deducing phase diagrams from spatial thermodynamic data using machine learning techniques. Calphad, 74:102303, 2021.
28. Shan Suthaharan. Support vector machine. In Machine learning models and algorithms for big data classification, pages 207–235. Springer, 2016.
29. Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 4489–4497, 2015.
30. Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 6450–6459, 2018.

31. Yi Gao, Zhaokun Liu, Xinmeng Wu, Guangyuan Wu, Jiahui Zhao, and Xiaofan Zhao. Skeleton- based human action recognition by the integration of euclidean distance. In 2021 The 9th International Conference on Information Technology: IoT and Smart City, pages 47–51, 2021.
32. Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012.
33. Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In Proceedings of the IEEE international conference on computer vision, pages 3192–3199, 2013.
34. Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE international conference on computer vision, pages 2334–2343, 2017.
35. Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp- mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems, 34:24261–24272, 2021.
36. T. Abeywickrama, M. A. Cheema, and D. Taniar. k-nearest neighbors on road networks: A journey in experimentation and in-memory implementation. Proceedings of the VLDB Endowment, 9(6), 2016.
37. Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6):275–285, 2004.
38. Jean-Franc¸ois Le Gall. Random trees and applications. Probability surveys, 2:245–311, 2005.
39. Liangmin Pan. Research on clustering algorithm of phishing websites based on gist global feature. PhD thesis, Central South University Of Forestry And Technology, 2018.
40. Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Thirty-second AAAI conference on artificial intelligence, 2018.

Yi Gao undergraduate student at the Intelligence and Computing Department of Tianjin University.

Haitao Wu undergraduate student at the Intelligence and Computing Department of Tianjin University.

Xinmeng Wu undergraduate student at the Intelligence and Computing Department of Tianjin University.

Zilin Li undergraduate student at the Intelligence and Computing Department of Tianjin University.

Xiaofan Zhao undergraduate student at the Intelligence and Computing Department of Tianjin University.

Received: January 31, 2022; Accepted: December 10, 2022.