Com SIS
Computer Science and Information Systems
Volume 19, Number 1, January 2022

Contents

ISSN: 1820-0214 (Print) 2406-1018 (Online)

Computer Science and Information Systems
Vol 19, No 1, January 2022

Com SIS

Computer Science and Information Systems

Published by ComSIS Consortium

Volume 19, Number 1
January 2022

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 19, Number 1, 2022
Novi Sad

Computer Science and Information Systems

ISSN: 1820-0214 (Print) 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 2020 two-year impact factor 1.167,
- 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 (MS Word or 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 Word or 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 19, Number 1, January 2022

CONTENTS

Special Section: Applications of intelligent systems

Computer Science and Information Systems 19(1): i–iii https://doi.org/10.2298/CSIS220100iI

Editorial

Mirjana Ivanović[1], Miloš Radovanović[1], and Vladimir Kurbalija[1]

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

This first ComSIS issue of Volume 19 for 2022 contains 14 regular articles and 7 articles in the special section "Applications of Intelligent Systems." Published papers cover a wide range of attractive contemporary topics, and we believe that readers will enjoy reading and sharing them among their colleagues. We are thankful for the hard work and diligence of all our authors and reviewers, without whom the current issue, and journal publication in general, would not be possible.

The first regular article, "Teaching Computational Thinking in Primary Schools: Worldwide Trends and Teachers' Attitudes" by Valentina Dagienė et al. begins this issue by examining worldwide tendencies in teaching computational thinking in primary education. A comprehensive survey and case study was performed to identify the level of teacher understanding of the subject and its integration approach in class activities, the results of which can be useful to primary school educators, educational initiatives, government authorities, policy makers, as well as e-learning system and content developers.

The second article, "Link Quality Estimation Based on Over-Sampling and Weighted Random Forest" by Linlan Liuet al. proposes a link quality estimation method which combines the K-means synthetic minority over-sampling technique (K-means SMOTE) and weighted random forest in order to address the problem of wireless link sample imbalance. Experimental results show that the proposed link quality estimation method has better performance with samples processed by K-means SMOTE, outperforming naive Bayesian, logistic regression and K-nearest Neighbor estimation methods.

In "Comparative Analysis of HAR Datasets Using Classification Algorithms, " Suvra Nayak et al. perform an experimental analysis on two publicly available human activity recognition (HAR) data sets using several classifiers: support vector machines, random forest and logistic regression. All algorithms perform very well on the given problems, with random forest notably outperforming the others on one of the data sets.

Maria Gorbunova et al., in their article "Distributed Ledger Technology: State-of-the-Art and Current Challenges" provide an exhaustive topical review of the state-of-the-art of distributed ledger technology applicability in various sectors, outlining the importance of the practical integration of technology-related challenges, as well as potential solutions.

"Entropy-based Network Traffic Anomaly Classification Method Resilient to Deception" authored by Juma A. Ibrahim and Slavko Gajin addresses the weakness of general entropy-based anomaly detection related to its susceptibility to deception by adding spoofed data that camouflage the anomaly. The article proposes two approaches focusing on, respectively, protection mechanism against entropy deception based on the analysis of changes in different entropy types, and extending the existing entropy-based anomaly detection approach with anomaly classification.

Davor Sutic and Ervin Varga, in "Scaling Industrial Applications for the Big Data Era" tackle the problems of power flow and island detection in power networks, and general

graph sparsification, focusing on scalability to large data sets. The authors introduce open source and distributed solutions involving algorithms for solving systems of linear equations, graph connectivity and matrix multiplication, and spectral sparsification of graphs, all featured in a released toolkit.

The article "A Graph-based Feature Selection Method for Learning to Rank Using Spectral Clustering for Redundancy Minimization and Biased PageRank for Relevance Analysis" by Jen-Yuan Yeh and Cheng-Jung Tsai addresses the feature selection problem in learning to rank (LTR). The proposed approach consists of four steps: (1) using ranking information to construct an undirected feature similarity graph; (2) applying spectral clustering to cluster the features; (3) utilizing biased PageRank to assign a relevance score with respect to the ranking problem to each feature; and (4) applying optimization to select features from each cluster.

"Deep RNN-Based Network Traffic Classification Scheme in Edge Computing System" by Kwihoon Kim et al. proposes a deep recurrent neural network based traffic classification scheme (deep RNN-TCS) for classifying applications from traffic patterns in a hybrid edge computing and cloud computing architecture, performing training on the cloud server and classification at the edge nodes. Extensive simulation-based experiments, show that the proposed approach achieves a notable improvement in accuracy while operating several orders of magnitude faster compared to the conventional scheme.

In "Building of Online Evaluation System Based on Socket Protocol, "Peng Jiang et al. describe the process of developing and online evaluation system for educational institutions based on the socket protocol, involving function design of students and teachers, data flow design, evaluation difficulty grading design, and system implementation. In addition, the article presents a method for difficulty classification of the evaluation, and of the test questions, laying the foundation for carrying out personalized testing and evaluation.

Miroslav Štampar and Krešimir Fertalj, in "Applied Machine Learning in Recognition of DGA Domain Names" address the problem of recognizing domain names generated by domain generation algorithms (DGAs) using machine learning algorithms. The authors engineered a robust feature set, and accordingly trained and evaluated 14 machine learning (ML), 9 deep learning (DL), and two comparative models on two independent data sets. Results show that if ML features are properly engineered, there is a marginal difference in overall score between top ML and DL representatives, which achieve performance comparable to the state-of-the-art.

"Semantic Web Based Platform for Harmonization of Teacher Education Curricula" by Milinko Mandić describes a semi-automatic software platform for harmonization of informatics curricula at all levels of education, involving ontology matching, as well as mapping informatics curricula to ontological models. Comparison of the informatics teacher education curriculum from the Republic of Serbia with the reference ACM K12 model, indicates that it is necessary to consider the improvement of teacher education curriculum, and application of new matching techniques.

Zorica Srđević et al. in "How MCDM Method and the Number of Comparisons Influence the Priority Vector" tackle the issue in multi-criteria decision making of determining the number of requited judgments a decision-maker/analyst needs to perform. The article presents a comparison of the results obtained by standard analytic hierarchy process (AHP), limited AHP and best-worst method (BWM) if the number of criteria is 6, 7, and 8.

The examples show that BWM's results are comparable with the results if standard AHP is used, while the limited version of AHP is generally inferior to the other two methods.

The article "Explainable Information Retrieval using Deep Learning for Medical Images" by Apoorva Singh et al. proposes an efficient deep learning model for image classification which tackles the following problems: (1) numerous image features; (2) complex distribution of shapes, colors and textures; (3) imbalance data ratio of underlying classes; (4) movements of the camera, objects; and (5) variations in luminance for site capture. Experimental evaluation on a real-world streaming data set demonstrates comparatively better performance than traditional methods.

The final regular article "RICNN: A ResNet&Inception Convolutional Neural Network for Intrusion Detection of Abnormal Traffic" authored by Benhui Xia et al. proposes a ResNet and inception-based convolutional neural network (RICNN) model for abnormal traffic classification. Experimental results on the show that RICNN not only achieves superior overall accuracy compared to a variety of CNN and RNN models, but also has a high detection rate across different categories, especially for small samples.

Computer Science and Information Systems 19(1): v–vi https://doi.org/10.2298/CSIS220100vB

Guest Editorial – Applications of intelligent systems

Amelia Badica[1], Costin Badica[1], Vladimir Kurbalija[2], Ladjel Bellatreche[3], and Mirjana Ivanović[2]

[1] University of Craiova, Romania
[2] University of Novi Sad, Serbia
[3] National Engineering School for Mechanics and Aerotechnics, France

This special section includes extended versions of selected papers from the International Conference on INnovations in Intelligent SysTems and Applications (INISTA 2020) held on August 24-26, 2020, in Novi Sad, Serbia and online due to the Covid-19 pandemic (hybrid approach). There were 51 accepted papers in the conference, and 7 of them was selected for this special issue. All of 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 intelligent systems.

The first paper "Hyper-parameter Optimization of Convolutional Neural Networks for Classifying COVID-19 X-ray Images" by Grega Vrbančič, Špela Pečnik and Vili Podgorelec presented an approach to transfer learning based classification method for detecting COVID-19 from X-ray images. The authors employed different optimization algorithms for solving the task of hyper-parameter settings. This approach achieved overall accuracy of 84.44% on a dataset of 1446 X-ray images, which outperformed both conventional CNN method as well as the compared baseline transfer learning method. The authors also conducted a qualitative in-depth analysis and gain some in-depth view of COVID-19 characteristics and the predictive model perception.

The second paper "A Fast Non-dominated Sorting Multi-objective Symbiotic Organism Search Algorithm for Energy Efficient Locomotion of Snake Robot" by Yesim Aysel Baysal and Ismail Hakki Altas deals with energy efficient locomotion of a wheel-less snake robot. The optimum parameters for the energy efficient locomotion of the snake robot are obtained with two different multi-objective algorithms based on symbiotic organism search algorithm. These parameters are tuned considering both minimizing the average power consumption and maximizing the forward velocity of the robot. The paper also investigates the energy efficient locomotion of the snake robot under different environment conditions.

The problem of employing gated architectures in Echo State Networks (ESNs) is explored in the paper "On the effectiveness of Gated Echo State Networks for data exhibiting long-term dependencies" by Daniele Di Sarli, Claudio Gallicchio and Alessio Micheli. Gated architectures have contributed to the development of highly accurate machine learning models that can tackle long-term dependencies in the data, but with the cost of highly demanding algorithms which require backpropagation through time. On the other side, ESNs can produce models that can be trained efficiently thanks to the use of fixed random parameters. However, these algorithms are not ideal for dealing with data presenting long-term dependencies. The authors concluded that using pure reservoir

computing methodologies is not sufficient for effective gating mechanisms, while instead training even only the gates is highly effective in terms of predictive accuracy.

The fourth paper "A Comparison of Deep Learning Algorithms on Image Data for Detecting Floodwater on Roadways" by Salih Sarp, Murat Kuzlu, Yanxiao Zhao, Mecit Cetin and Ozgur Guler concentrates on object detection and segmentation algorithms. More precisely it investigates detection and segmentation of (partially) flooded roadways for the purpose of vehicle routing and traffic management systems. This paper proposes an automatic floodwater detection and segmentation method utilizing the Mask Region-Based Convolutional Neural Networks (Mask-R-CNN) and Generative Adversarial Networks (GAN) algorithms. The results show that the proposed Mask-R-CNN-based floodwater detection and segmentation outperform previous studies, whereas the GAN-based model has a straightforward implementation compared to other models.

The fifth paper "An Approach for Selecting Countermeasures against Harmful Information based on Uncertainty Management" by Igor Kotenko, Igor Saenko, Igor Parashchuk and Elena Doynikova explores the possibilities for the counteraction against the spread of harmful information in the Internet. The paper considers models, algorithms and a common techniques based on an assessment of the semantic content of information objects under conditions of uncertainty. An experimental evaluation has shown that it is possible to eliminate uncertainties of any type and, thereby, to increase the efficiency of choosing measures to counter harmful information.

The problem of using the consensus of collectives for solving problems is studied in the paper "An Effective Method for Determining Consensus in Large Collectives" by Dai Tho Dang, Thanh Ngo Nguyen and Dosam Hwang. The rapid development of information technology has facilitated the collection of distributed knowledge from autonomous sources to find solutions to problems. However, due to rapid increment of collectives, determining consensus for a large collective is very time-consuming and expensive operation. Therefore, this paper proposes a vertical partition method (VPM) to find consensus in large collectives. The authors show, both theoretically and experimentally, that the computational complexity of the VPM is lower than the basic consensus method.

The last paper in this special section "Automatic Derivation of the Initial Conceptual Database Model from a Set of Business Process Models" by Drazen Brdjanin, Aleksandar Vukotic, Danijela Banjac, Goran Banjac and Slavko Maric presents a possibility of automatically deriving the initial conceptual database model from a set of business process models. The proposed approach proposes the incremental synthesis of the target model by iteratively composing the partial conceptual database models that are derived from the models contained in the source set. The experimental evaluation shows that the implemented approach enables effective automatic derivation of the initial conceptual database model.

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 19(1):1–24 https://doi.org/10.2298/CSIS201215033D

Teaching Computational Thinking in Primary Schools: Worldwide Trends and Teachers' Attitudes*

Valentina Dagienė[1], Tatjana Jevsikova[1], Gabrielė Stupurienė[1], and Anita Juškevičienė[1]

[1] Vilnius University Institute of Data Science and Digital Technologies, Akademijos str. 4, LT-08412 Vilnius, Lithuania
{valentina.dagiene, tatjana.jevsikova, gabriele.stupuriene, anita.juskeviciene}@mif.vu.lt

Abstract. Computational thinking (CT) as one of the 21st century skills enters early years education. This paper aims to study the worldwide tendencies of teaching CT through computing in primary education and primary school teachers' understanding of CT. A survey of 52 countries has been performed and complemented by a qualitative study of 15 countries. In order to identify teachers' understanding-level of CT and its integration approach in the class activities, a case study of 110 in-service teachers from 6 countries has been performed. The implications of the research results may be useful for primary school educators, educational initiatives, government authorities, policy makers, e-learning system and content developers dealing with support for teachers aiming to improve their CT professional development qualification.

Keywords: informatics education; computational thinking development; early years' education; 21st century skills; teacher professional development.

1. Introduction

In the 2000s, education policy makers of many countries started to bring computer science into school curricula as a response to demands of 21st century learning skills. Instead of the term "computer science" many countries especially in multilingual Europe use other names (e.g., computing or informatics) or enrich with technology-based components (e.g., digital fluency, digital competence). The terms "informatics", "computing" and "computer science", used in this paper, refer to more or less the same subject, that is, the entire discipline.

In European countries it is common to use informatics or its similar translations. Several documents have declared a strong separation between informatics and digital literacy [9, 7], whereas other international initiatives, like DigComp, has included

* This is an extension of work presented in the conference paper: Dagienė, V., Jevsikova, T., Stupurienė, G.: Introducing informatics in primary education: curriculum and teachers' perspectives. In: Informatics in Schools. New Ideas in School Informatics. ISSEP 2019, Lecture notes in computer science, Springer, Cham, Vol. 11913, 83–94 (2019). DOI: https://doi.org/10.1007/978-3-030-33759-9_7.

elements of informatics (e.g., programming) into the digital competence area "Digital content creation" [16].

A decade ago the term "computational thinking" [33, 34] started to flourish in the United States, and Europe recognized the importance of an appropriate informatics education in schools.

Computational thinking (CT) has been actively promoted in schools in an integrative way or as a part of informatics subject, as it addresses concepts and learning goals of informatics. We may notice that interest in teaching informatics in primary school has increased during the last decade [24]. As recommended for informatics education in the report by the Committee on European Computing Education, "All students must have access to ongoing education in informatics in the school system. Informatics teaching should preferably start in primary school…" [9, p. 3].

Recognizing the importance of introduction of CT in primary school, the path towards this goal has many challenges, e. g. curriculum design and implementation, in-service and pre-service teacher training [23] due to novelty and possible misconceptions of CT, e.g. [18].

The **aim** of this paper is two-fold: to study worldwide informatics curriculum tendencies in primary education and determine primary school teachers' understanding of CT.

The main **Research Questions** we aim to answer via this study are:

1. What are the latest tendencies in introducing informatics and computational thinking through informatics in primary education in various countries and what are the recommendations based on the experts' experiences?
2. What is primary school teachers' understanding (or their possible misconceptions) of computational thinking?

In the next Section we discuss CT and existing practices of its implementation in primary school. In Section 3, we present research methodology. The results of quantitative and qualitative studies hold with experts, representing countries, and in-service teachers are discussed in the Results section. Finally, we discuss limitations, provide conclusions, discussion and recommendations.

2. Computational Thinking and Informatics in Primary Education

Computational thinking became popular after Jeannette Wing published a paper declaring that "computational thinking represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" [33, p. 33] and that the U.S. National Science Foundation included this idea in a funding call in 2007. Despite S. Papert's initiatives in computing education and publication of his 1980 book "Mindstorms" [29], many educators and other people heard arguments about the value of informatics (computer science) in education for the first time. Seymour Papert coined the concept of "computational thinking" in "Mindstorms" and used it for teaching procedural thinking to children.

Later J. Wing gave a more concrete definition, stating that CT involves solving problems, designing systems, and understanding human behavior, by drawing on the

concepts of CT [34]. It includes naturally a range of mental tools that reflect the breadth of the field of informatics. So computational thinking became a new wave in the movement to provide students with powerful tools for solving real world problems. The rapid growth of newcomers in this area led to considerable confusion about learning objectives and the essence of CT [13].

CT is a term applied to describe the increasing attention on students' knowledge development about designing computational solutions to problems, algorithmic thinking, and coding. Based on a systematic literature review the dimensions of CT are discussed and a three-stage model for developing computational thinking is proposed [28].

Peter J. Denning and Matti Tedre have in their book [14] portrayed CT in all aspects, its richness and deepness, and presented a long history of computing and connected to current practical challenges. "There has never been a consensus about what computational thinking "really" is. <...> We should embrace the lack of a fixed definition as a sign of the vitality of the field rather than our own failure to understand an eternal truth." [14, p. 217].

Many countries have brought informatics into secondary school curricula with the intention to teach students important skills that no other subject does (e.g. computing design, programming). Until the early 2000s informatics was part of the upper secondary school curriculum (grades 10–12). Teaching fundamental computational thinking skills such as computer modeling or programming is much harder than introducing pupils to text processing or other application tools. Because of the shortage of qualified teachers, teaching informatics was replaced by focusing more and more on application of information and communication technologies (ICT). This resulted in a heavy focus on using computers and tools instead of informatics concepts, as teachers were not used to or trained in the latter [4].

Several countries have included informatics as a part of digital literacy or information technologies curricula for secondary schools. For example, the Netherlands published the report "Digital Literacy in Secondary Education - Skills and Attitudes for the 21st Century" and included the elective informatics subject in the upper grades [20]. Furthermore, in that report, CT is considered to be an integral part of digital literacy. Lithuania renamed informatics as information technologies and developed new curricula for grades 5 to 12 in 2005 [10, 11].

Research on education in the early years has shown that it is important to encourage children to reflect upon the modern world around them and focus on real-world problems using various technologies [27]. Primary schools have recognized the importance of including elements of CT in their teaching and started to incorporate them in lessons [1]. However, there are difficulties with teacher training and professional development in the computing discipline. Extensive and detailed methodological support should be provided to ensure that the relevant competences would be achieved [32].

International organizations like Computer Science Teacher Association (CSTA), the British Computing at School (CAS), Australian Curriculum Assessment, and Reporting Authority (ACARA) developed frameworks for computational thinking education in schools starting from elementary or primary schools to upper secondary schools. The CAS introduced a new subject of computing that replaced information technology in UK schools [6]. Australia has developed a curriculum called Digital Technologies: this is a new national subject within the Technologies learning area since 2016 [2, 3]. The

subject is mandatory from kindergarten to year 8, with elective choice for pupils in year 9 and 10. The digital technologies curriculum includes fundamental ideas from the academic disciplines of computer science, information systems and informatics. Media arts and online safety are integrated correspondingly into arts, health and physical education, while ICT is integrated across all subjects. CSTA developed detailed K-12 Computer Science standards that delineate a core set of learning objectives designed to provide the foundation for a complete informatics curriculum and its implementation at schools [8]. These three curricula have been analyzed and notable insights have been provided [15]. One of the important conclusions is that introducing informatics at an early age can broaden the pupils' perception of computing and create their interest in technological disciplines. The review of informatics education in Australia, England, Estonia, Finland, New Zealand, Norway, Poland, South Korea, and Sweden the US provide information on initiatives taking part in primary education [21, 22].

When introducing informatics in primary education, we face several challenges. Peter Hubwieser et al. [23] have concluded in their research on informatics education: 1) proper teacher education in substantial extent and depth seems to be one of the most critical factors for the success of rigorous informatics education on the one hand and also one of the hardest goals to achieve on the other; 2) there is a convergence towards computational thinking as a core idea of the K-12 curricula; 3) programming in one form or another, seems to be absolutely necessary to introduce to the students. So pre-service teacher education and in-service teacher training are the crucial points when introducing CT to primary schools. In order to motivate teachers to learn about CT, they have to see positive benefits for both them and their pupils. Having access to high-quality training courses and engaging educational resources is hence crucial [25, 26].

A detailed study on coding in primary schools was conducted by Rich et al. [30]. Some international trends in teaching informatics at primary-school level were drawn. More than 55% of 310 teachers (from 23 countries) had no or very little training with computing/coding prior to deciding to teach it in the classroom. Average experience of teaching computing/coding is 4.6 years (although such experience varies greatly by country). Teachers revealed that they don't necessarily have a high confidence in their computing/coding ability, but that seems not to be impeding them from teaching it as a subject (even though this was their most prevalent apprehension and challenge).

An instrument to survey teachers about their implementation of informatics curriculum to understand pedagogy, practice, resources and experiences in classrooms around the world was developed and implemented [17]. The researchers reviewed and analyzed pilot data from 244 teachers across seven countries (Australia, England, Ireland, Italy, Malta, Scotland and the United States). The resulting instrument combines a country-level report template and a teacher survey that will provide teachers with a means to communicate their experience enacting informatics curricula.

Fessakis and Prantsoudi [18] presented the research of the perceptions, attitudes and beliefs on CT of informatics teachers (N=136) who teach in various types of schools (primary, secondary, higher education) in Greece. Results show that for successful integration of CT in classroom lessons, teachers need to have clarified the practices and dimensions of the concepts. Also, an appropriate knowledge of informatics didactics is needed. Concerning the relation between CT and informatics, misconceptions also occur, as most teachers mistakenly consider informatics as a subset of CT or as a totally different field. Only ¼ of them perceive that CT and informatics intersect and are not

totally different cognitive fields. These misconceptions need to be treated with proper in-service professional development and/or pre-service education.

3. Research Methodology and Respondents

3.1. Research Design

This research study employs a mixed method approach in order to investigate the current situation in informatics education in primary schools and teachers' understanding of CT in different countries, i.e., both quantitative and qualitative approaches are used in order to collect and analyze data. From the methodological viewpoint, our research was designed as a survey. Detailed composed questionnaires were used for the data collection processes.

Our research consisted of two studies and three phases as presented in Table 1.

Table 1. Studies and phases of the present research

First study: experts' opinions		Second study: in-service teachers' opinions
Phase I Quantitative research	Phase II Qualitative research	Phase III Quantitative and Qualitative research
Survey on experts' opinions about the most up-to-date information about the practice and situation of introduction of informatics in primary education in 52 countries.	Continuous survey on selected experts' from 15 countries that have informatics curriculum introduced or where it is being developed at the moment.	A case study on primary and pre-primary teachers' understanding about CT and its integration in the class activities in 6 different countries.

In order to answer Research Question 1, we survey the situation on informatics in primary education in different countries (52 countries included, phase I of the first study). Our quantitative data are supported with qualitative data collected from the countries that have implemented informatics curriculum or undergo its development process (phase II of the first study). In order to answer Research Question 2, the survey of 110 in-service teachers (a case study) has been conducted (phase III, the second study).

3.2. The First Study: Experts' Opinions

The Phase I of the first study included the quantitative analysis of answers provided by experts from 52 countries. This study was conducted during the spring-summer period of 2019, and the first results were presented to the international community during the ISSEP 2019 (Informatics in Secondary Schools: Evolution and Perspectives) conference [12]. The study of experts' opinions let access the most up-to-date and generalized information about the practice and situation of introduction of informatics in primary education in different countries.

The requirements for the respondents (experts), participating in the study were very high: an expert was involved in the creation of the national education system, developed curriculum and methodological material in informatics (or similar discipline depending on how it is called in respondent's country), and was knowledgeable of the situation at the primary education level. If an expert could not answer all the questions, the expert suggested another expert to whom the questions were redirected.

In addition, we asked each expert to self-evaluate the level of confidence of their answers on the scale from 1 (low) to 5 (high). General confidence level was evaluated by the experts as high (median: 5, mean: 4.6). The list of countries, represented by the experts, includes 34 countries from the European region (Austria, Belarus, Belgium, Bosnia and Herzegovina, Bulgaria, Croatia, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Macedonia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Russia, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Ukraine, United Kingdom), and 18 non-European countries (Algeria, Australia, Cyprus, Cuba, India, Indonesia, Iran, Japan, Malaysia, Palestine, Philippines, Singapore, South Africa, South Korea, Thailand, Tunisia, Turkey, Uzbekistan).

The phase II of the first study was aimed at extending the study to qualitative research. For this purpose, only countries that have an informatics curriculum introduced or where it is being developed were selected. The questionnaire was completed by experts from the following 15 countries: Australia, Belarus, Bosnia and Herzegovina, Estonia, Greece, Indonesia, Malta, Netherlands, Poland, Romania, Russia, Sweden, Switzerland, Ukraine, and the UK. The goal of this phase was to gain a deeper understanding of the background for the answers given by the closed-type questions in the quantitative study in phase I, by probing the knowledge and recommendation from the experts in these countries. Usually, there is considerable variation between schools and districts in a country. Therefore, we selected as much as possible experts who were considered as national experts and presented recommendations based on their experience. We note that the comments provided by the experts reflect their personal opinion and do not reflect those of the organization they are representing. This study has been conducted during the summer-autumn period in 2019.

The qualitative questionnaire included the following questions to the experts representing the countries:

- What are the main areas of competence (content topics), included in the newest version of primary school informatics curriculum?
- What learning theories and methodologies are most usually used in primary school informatics in your country?

- What tools are most usually used in informatics primary education?
- What best practices and failures in informatics integration in primary education do you notice in your country? If there were just a start of informatics integration in primary education right now, what would you have done differently? Consider the following aspects: informatics as a separate/integrated subject; Age of students; Pre-service and in-service primary school teacher training and teacher support; Curriculum development and update; Research; Organizational aspects.

The preliminary results of the first study have shown that there is a lack of teacher training activities. Therefore, we decided to study the situation with focus on the primary education in-service teachers' level. At this phase we decided not to limit to informatics as a subject concept, but use CT as a universal way of addressing skills of informatics and other related subjects, no matter if the country has introduced an informatics curriculum or not. The study was aimed at examining the understanding of CT by the in-service teachers (the second study, phase III).

3.3. The Second Study: In-Service Teachers' Opinions

This study corresponds to phase III (Table 1). 110 pre-primary and primary teachers from 6 countries answered the questionnaire about their own understanding of computational thinking and its integration approach in the class activities. The distribution of teachers by their teaching type is as follows: class teachers (70.91%), STEM subject teachers (mainly mathematics, science, and technology teachers) (11.82%), special education teachers (4.55%), and others (e.g., administration) (12.73%). By a class teacher we consider a primary school teacher who usually teaches all subjects from preschool through fourth to sixth grade in most cases. In the study, there were teachers from 6 countries: Belgium (43 respondents), Finland (11 respondents), Lithuania (23 respondents), Portugal (6 respondents), Spain (20 respondents), and Sweden (7 respondents). This study has been conducted during autumn 2019.

As not all surveyed countries at the time of the survey had officially approved the curriculum of informatics as a subject in primary school, we used the computational thinking concept in order to identify the level of its integration in primary education based on teachers' perspective. We included pre-primary teachers due to the differences in the students' age in primary education and in order to examine preparation for informatics in primary education.

The online questionnaire, which took 15–20 minutes to answer, consisted of 14 questions that included eleven five-level Likert-scale questions (see the list of questions in Table 2) and three open-ended questions as follows: Describe what you understand by CT; If you already work on CT in class, describe how; Describe your needs to work on CT in class.

4. Results I. Study of Experts' Opinions

4.1. General Implementation

The vast majority of surveyed countries (83%) teach at least some elements of informatics in primary education. However, there are many differences in the level of informatics implementation. Out of the countries who teach at least some elements of informatics, 26% have either a non-compulsory (elective) informatics subject in primary education, or the level of introduction of informatics in primary education differs depending on the region, school type (e.g. private), choice done by school, and other factors.

It is important to know which age groups different primary education systems embrace. Most frequent lower and upper age boundaries are 6 and 11, and in general, the age of pupils in primary education in surveyed countries ranges from 3 to 16. According to experts' opinion, informatics education should start already at an early age, using playful approaches (more unplugged activities) and not forcing children to repeat formal tasks like steps or commands. In Lithuania, informatics is going to be introduced starting from pre-school (one year before grade 1).

Out of the countries who teach informatics in primary education (N = 37), 44% introduce it in the first year of primary school, 3% in the second year, 22% introduce informatics in grade 3, the same number (22%) in grade 5, and correspondingly 3% and 6% introduce it in grade 2 and grade 6.

Of the 17 countries who introduce informatics in grades 1 or 2 (47% out of countries with teaching of informatics in primary education; these 17 countries being Australia, Belarus, Bosnia and Herzegovina, Cuba, Denmark, Estonia, Greece, Indonesia, Norway, Poland, Romania, Russia, Sweden, Switzerland, Thailand, Ukraine and the UK), all (except Thailand) reported to have informatics curriculum for primary school or it is being developed at the moment of the survey.

4.2. Curriculum Issues

27 countries, that is 52% of the surveyed countries, have already introduced an informatics curriculum for primary education: 56% among them are countries of the European region (Fig. 1). There is no informatics curriculum in primary education in 27% of all surveyed countries, and the curriculum is being developed at the moment in 21% of all surveyed countries. Active development of new curriculum can be noticed in the surveyed countries from the European region (91% of all respondents who stated that curriculum is under development). However, when presenting generalized results, we should be aware of the factors described in the Limitations section.

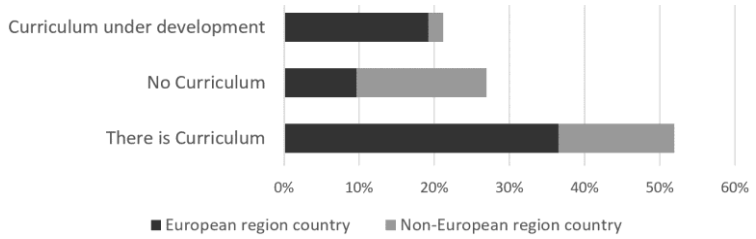


Fig. 1. Existence of informatics curriculum for primary education in the surveyed countries, N=52

During the first stage of the survey we asked experts whether six areas of primary informatics education are being developed in their primary education (each area had explanations on the content included into it). It is not a surprise that informatics content areas in primary education are similar from country to country, but that the naming and assigning to the particular category differs. For this reason, we selected a model to compare with. The model for the selected topics corresponds to the Lithuanian informatics curriculum for primary and pre-primary education, which is in the implementation process:

– *Digital content*. Essential skills of working with digital devices; managing textual, graphical, numeric, visual, audial information; information visualization and presentation; digital content creation.
– *Algorithms and programming*. Solving problems: algorithm, action control commands (sequencing, branching, looping), programming in a visual programming environment for children.
– *Problem solving*. Essential technical and technological skills of working with digital devices: solving technical problems, evaluating and identifying suitable technologies for the selected problem, creative use of technologies.
– *Data and information*. Working with data skills: problem analysis, data collection, sorting, search and data management, content quality evaluation.
– *Virtual communication*. Social skills in a virtual environment: continuous learning, e-learning, communication via email, chats, social networks, sharing, collaboration, reflection.
– *Safety*. Digital safety, safe work with digital devices; ethics and copyright issues of information processing and usage; safety, ethics and copyright issues in virtual communication.

The areas are being addressed in primary informatics education of up to 72% of countries. From 42% to 62% of the addressing countries are the countries who have implemented informatics curriculum or being developed it at the moment of the survey, and up to 10% are those countries with no curriculum (Fig. 2).

In the countries with existing informatics curriculum or curriculum under development, the following three content areas - Digital content (62% out of all 52 respondents), Algorithms and programming (54% of all surveyed countries), Safety (50% of all respondents) - are taught more often. Digital content and Algorithms and programming are even taught in some countries that have not introduced an informatics curriculum.

As it was mentioned before, 17 countries introduce informatics in grades 1 or 2. Almost all of them teach Digital content skills, and more than 70% of these countries introduce all other areas.

Further communication with experts from 15 countries that are either in a stage of active development of informatics curriculum in primary education, or already have introduced it, has shown that mostly just the naming of content topics differs between countries. For example, from the curriculum in Greece: 1. I know, create and express myself with ICT; 2. I communicate and collaborate using ICT; 3. I inquire-explore, discover, and solve problems using ICT; 4. ICT as a phenomenon in society. Bosnia and Herzegovina uses: Introduction to ICT; Components of computer systems; Treatment of data; Digital and virtual world; Algorithms and data structures. In Estonia: Key stage 1: Digital art, (playful) coding and digital safety. Key stage 2: Digital media, (visual) programming and digital hygiene. In the Netherlands the new curriculum has started to be implemented in 2020. The domains of digital literacy are discussed in these areas: information skills, media literacy, basic ICT skills and CT (solving issues or problems using digital technology). In primary education, pupils learn to use digital resources consciously in their own context and level of learning.

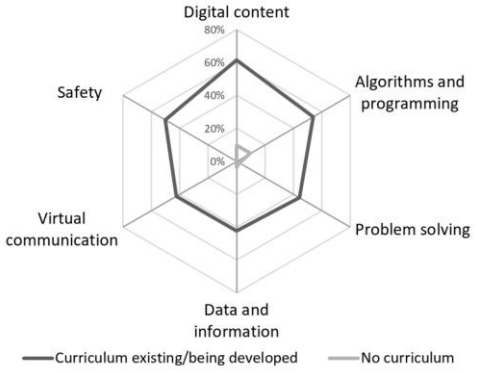


Fig. 2. Informatics content areas in primary education in surveyed countries (N=52)

In Switzerland the subject "Media education & informatics" consists of the competence areas defined by: 1. Pupils can present, structure and evaluate data from their environment; 2. Pupils can analyze simple problems, describe possible solution methods and implement them in programs; 3. Pupils understand the structure and operation of information processing systems and can apply concepts of secure computing.

The government of Italy has recently approved the introduction of CT and coding in compulsory school curricula by 2022, clarifying that this will not imply the introduction of a new subject neither in primary school nor in lower secondary school [19]. This suggests that CT will be probably handled as a transversal subject in primary school and within the existing subjects of mathematics or technology (including information technology) in lower secondary school. This situation is not unique in Europe – consider, for example, France, Finland, Sweden, and Norway, which have recently completed a curricula reform following this path [5].

4.3. Subject Integration

When comparing informatics integration in primary education, we face the effect of different age groups in primary education in different countries (e.g., in some countries primary education embraces grades 5, 6, and only for these grades, separate subject is used, while in other countries these grades are included in basic education level), phenomenon of changing integration through different grades (e.g., integrated in grades 1 to 3, while separate in grade 4), region-level and school-level differences in primary education (school/region level choice). These issues affected the data that we collected using quantitative questionnaires (with closed-question options available). When analyzing the quantitative data, out of the respondent countries who either have curriculum or undergo curriculum development process at the moment (N = 38), 50% of countries introduce informatics as a separate subject in primary education.

As it was discussed in the introductory section, the name of the subject varies across the countries, e.g., computing, computer science, computer modeling, information technologies, ICT, digital technologies, media education, technology and design. 21% of the countries include the basics of informatics in primary education in an integrated way. Some countries introduce it both as a separate subject and as integrated to other subjects either due to pilot study taking part at the moment (e.g., Denmark), due to differences in school years (e.g., in Switzerland, for grade 1–2 the subject is integrated, for grade 3–4 the subject is separated), or due to possibility to select on a school level (e. g., Czechia).

Further communication with experts via qualitative questionnaire (Phase II) enabled us to obtain more information and recommendation on the subject integration.

In the fifteen additionally surveyed countries, three ways of teaching informatics are used: as a separate subject; as a subject integrated into other disciplines' activities; mixed approach when there are both separate subject and natural integration into other subjects, or there is integrated informatics up to some grade, and then separated subject, or there is a selection possibility.

As reported by the experts, informatics as a separate subject is taught in Australia, Bosnia and Herzegovina, Greece, and Ukraine. Some experts from these countries provided opinion based on their practice that "interdisciplinary approach for the integration to other subjects should be developed" (Greece).

Estonia, the Netherlands, Poland, Russia, and Switzerland, and reported to use both approaches: integrated and separated subjects. In other surveyed countries, an integrated approach is used.

For those countries, where informatics curriculum is being developed, or it is being integrated into other subjects, experts state that some part of informatics subject should be learnt separately: "After learning the basics of informatics it can be integrated in other subjects" (Estonia). "Students should be prepared in other literacies, like coding" (Malta). "Integrated subject has pros and cons. In a short time <...> it is the best. In the long term a separate subject may be better" (Sweden).

In Australia, the subject of Digital Technologies is introduced in a separate way [3]. Teachers are encouraged to teach digital technologies knowledge and understanding as a separate topic, but to teach the processes and production skills integrated in other subjects in parallel, for example by creating a quiz game to show how the human body works and using robotic toys in literacy and numeracy activities. This has been a challenge to trained teachers. Digital literacy, which comes closest to informatics, is

introduced as both a separate subject and integrated into other fields. An expert proposed: “appointing a specialist for digital literacy in every school who can serve as an expert, helping out other teachers is recommended”.

One expert, representing the United Kingdom, provides an insight that “the cross-curricula element is a context in which an activity is set, e.g. Vikings as a context for making artwork, Math for a presentation. Whether we are really stretching pupils and differentiating effectively across subjects is very questionable. However, by setting computing in cross-curricular contexts then there is possibly more motivation for teachers to include the activities. English curricula are stated separately, and teachers find it hard to very effectively merge objectives in a single lesson. But they can do this over sequences and in themed topics. However, I have not seen research which proves this is effective”.

In primary schools of Poland, a stand-alone informatics subject is called Computer activities and runs through grades 1 to 6. In grades 1-3, computer activities are supposed to be fully integrated with other activities like reading, writing, calculating, drawing, playing etc. At the next stages of education, students are expected to use computers as tools supporting learning of various subjects and disciplines, in a formal, non-formal, and incidental manner in school and at home [31].

4.4. Teacher Training

A key question in the quality of informatics teaching in primary education is teacher training. Out of the 52 respondent countries, 77% have included elements of informatics into primary teacher education programs (data for one non-European country is not available) (Fig. 3). 27% of countries include all main aspects of computing in primary teacher training programs (answer “Yes”). However, almost half of the surveyed countries (46%) has teacher training mostly limited to digital literacy. In two countries (4%, Finland and Czechia), on the contrary, primary teacher training in informatics mostly includes programming.

It should be noticed that all countries who answered “Yes” to the question on teacher training have informatics curriculum in primary education.

Training in primary teacher education programs is limited mostly to digital literacy and dominates among all countries, even in those who have introduced informatics-related curriculum in primary education or where such curriculum is being developed.

Three experts, representing countries with informatics-related curriculum in primary education but without teacher training included into primary teacher training programs, commented that it is planned to be introduced soon (Denmark), some programs do (Poland), or informatics elements in primary education are taught by school teachers of informatics (Latvia).

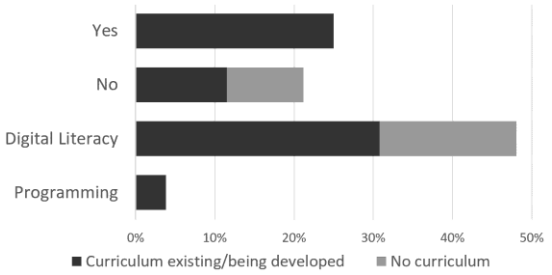


Fig. 3. Informatics inclusion in primary teacher education programs, N = 52

Further probing of the experts provided more generalized information on the topic. In all 15 countries of the Phase II study, pre-service and in-service teacher training activities are taking place. It is included in pre-service teachers' education programs in colleges and universities, but the time allocated for the subject differs from university to university even within the same country. The majority of experts' state that there should be more training. It is also stated, for instance, that "newly graduated students can serve as an example for more experienced teachers to learn from" (the Netherlands). In Greece and Bosnia and Herzegovina, informatics is usually taught by a computer science graduate, and therefore, more pedagogical training rather than subject training is needed. An expert from Malta stated the need to train university tutors in the basics of informatics and usage of information technologies in their work, so that student teachers can see good examples of information technology integration in educational processes.

Professional development courses are organized as in-service teacher training in informatics. Countries are developing online courses for teachers, organize face-to-face training and provide funding opportunities for competence development. Several cases are presented below.

In Australia, in-service teacher training has been "... a challenge to up-skill teachers. There have been many funding opportunities offered by both federal and state governments. Some schools and teachers are still resistant to change". The United Kingdom offers a new programme for in-service teachers funded by the government in England called the National Centre for Computing Education (NCCE), with £84m, which runs from November 2018 to July 2022. In Sweden, the National Agency for Education has developed online courses (about programming and how to program) and is also paying for university courses for teachers. There is more training for in-service teachers taking place than for pre-service. The Ministry of Education of Romania provides support and takes care of in-service teacher training. In Ukraine, in-service teachers are provided every year with online courses and a few weeks of learning at regional centers during summer. Poland states that many local and national activities are taking place. In Russia, it is defined by the Federal standard that each teacher must take a refresher course of at least 72 hours every 3 years. Courses for primary school teachers include all subjects, including computer science. Each publishing house, whose textbooks are included in the official Federal list of textbooks recommended by the Ministry of Education, provides methodological support to teachers on the basis of its methodological services. At the same time, primary school teachers are accustomed to the outdated model of teaching only the theoretical foundations of computer science.

Switzerland and Belarus state to have mandatory preparation of pre-service and in-service teachers. In Indonesia, teacher training is organized by the government and community services by the Bebras Indonesia community. Bebras Indonesia NBO collaborates with about 50 universities all over the country, to introduce the Indonesian K-12 informatics curriculum to neighboring schools.

4.5. Educational Resources and Methodological Aspects

The experts were asked questions about tools (hardware and software), educational resources and methods used in their countries for informatics education in primary school. The experts noted that there is an endless list of tools and they vary from school to school (they can choose themselves). A typical scenario for tools is described by an expert from Australia: "Some schools are creating smart gardens using micro:bits or Arduinos. Some schools run Coding Clubs using Code Club Australia resources. Others use Lego robotics and compete in the First Lego League. In primary schools, students use visual programming to implement solutions using a variety of tools and platforms at the discretion of the teacher, including but not limited to: Scratch, Blockly, Lego robotics, ScratchJr, Sphero, Ozobot, Dash robot, Swift playgrounds, Kodable, code.org, etc. Many schools use the Bebras challenge to teach and test CT skills."

Only experts from Australia, Greece, The Netherlands, Russia, and the United Kingdom mentioned that they have repositories for educational resources. Other countries lack modern and up-to-date literature and nation-wide resources for teachers. Also, the methods used for primary education in all 15 countries mostly depend on teachers. Teachers need a lot of support, starting from teacher training and resources.

Methodology used to teach informatics elements in primary education highly depends on a school or a teacher. But, among most usually used methodologies, experts mention: collaborative learning, group and pair work projects, problem solving, inquiry-based learning, role-playing, game-based learning, learning by doing, tinkering, interdisciplinary approach, PRIMM (Predict, Run, Investigate, Modify, Make), Use-Modify-Create, design approach, worked examples. Some schools (e. g. in Australia) are using the TPACK model and others are using the SAMR (Substitution, Augmentation, Modification, and Redefinition) model.

5. Results II. Survey on Teachers' Understanding of CT

5.1. Teachers' Attitudes on Computational Thinking

Teachers were asked the open-ended question *Q1: Describe what you understand by CT*. All respondents answered this question. The answers were grouped in 6 categories, based on the frequency, that show the same or similar understanding, and summarized in Fig. 4. The main categories are described below.

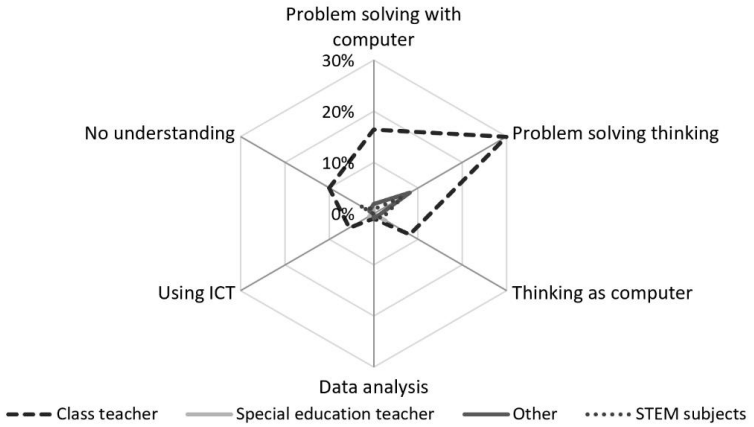


Fig. 4. Understanding of computational thinking by different types of teachers (N = 110)

The category "Problem solving thinking" includes logical and critical thinking and decision making, and according to the teachers' answers it can be defined as "an ability of a person to solve problems having interiorized the concepts in a computational language and having a critical sense". One more description that characterizes this category: "to think in logical, discrete steps and reduce, reformulate complex problems into well characterized models to which we can apply standard solution methods". The answers of 44.5% of all teachers fall into this category (83% from Portugal, 72% from Sweden, 60% from Spain, 46% from Finland, 35% from Lithuania and 33% from Belgium).

The category "Problem solving with computer" consists of answers related to problem solving and practical use. According to the answers, it can be defined as "ability to objectively analyze and evaluate a problem, develop possible solutions to the problem and then format these solutions in ways that a computer could execute". The answers of 20% of all teachers were assigned to this category (the category was based on the answers of teachers from Spain and Belgium).

The category "Thinking as computer" involves programming as well as using algorithms, and according to teachers' answers, it can be described as "it is learning to think like a computer and how you can use computers to do logical measurements". The answers of 13.6% of all teachers fall into this category (23% from Belgium, 17% from Portugal, 14% from Sweden).

6.4% of teachers showed misconceptions in understanding CT by providing answers related to the use of only CT, for example described as "computational thinking is using ICT in daily purposes". Answers that were categorized as "No understanding" were provided by 13.6% of teachers (36% from Finland and 35% from Lithuania).

We have analyzed teachers' understanding of CT according to different types of teachers. The results, presented in Fig. 5, show that 54% of STEM subject teachers and 42% of class teachers (i.e. primary school teachers) as well as 64% of other teachers understand CT as problem solving thinking. Unfortunately, 23% of STEM and 14% of class teachers provided answers showing no understanding.

5.2. How Teachers Teach Computational Thinking in Class

Teachers were asked an open-ended question *Q5: If you already work on computational thinking in class, describe how*. Only half of teachers answered this question. The answers were categorized according to the usage of tools: programming (coding), robotics, tangible devices, and unplugged activities. 43% of the teachers who answered this question named the tools they have used to develop CT skills of pupils. The tools, mentioned by the teachers, are the following: Scratch, ScratchJr, code.org, Minecraft Education, LEGO, WeDo, Bee-bots, Ozobot, mBot, micro:bit, Makey Makey, Scottie Go, Cubetto, Bebras cards.

Half of the teachers who responded to this question have described their work on CT in class through the methodology aspects. Some examples: "I try to induce pupils to "think on paper" including drawing the problem, finding patterns such as geometric shapes and to identify pieces of the problems to solve one by one. I also encourage pupils to reformulate problems into easier ones, i.e. if the numbers are large and hard to grasp, make a new problem with easy numbers where the pupil can anticipate the answer"; "Face situations through critical thinking with sequences and instructions to solve a problem in a simpler way".

In 7% of the answers, the teachers described the usage of ICT as enhancing their pupils' CT skills. It shows limited understanding of computational thinking in education.

5.3. Factor Analysis

Closed-type questions regarding CT education and understanding by the teachers (see Table 2) have been analyzed by conducting factor analysis. The factor analysis has been chosen for a twofold reason: to reduce the number of variables and to validate our questionnaire. This method aims at identifying clusters (components) of items for which responses for the questionnaire on understanding of CT had common patterns of variation. Each factor corresponds to a group of variables whose members correlate more highly among themselves than they do with variables not included in the factor. Table 2 shows the identified four factors using Varimax with Kaiser Normalization rotation method.

Four factors have been identified:

1. Active integration of computational thinking (questions Q2, Q3 and Q4).
2. Computational thinking as algorithmic thinking and/or as problem-solving (Q7, Q13, Q14).
3. Computational thinking has been associated with a tool (Q8, Q9 and Q10).

Computational thinking as digital literacy and thought processes (Q11 and Q12).

Table 2. Results of factor analysis

Variable	Factor			
	1	2	3	4
Q4. I work already on CT in my class.	.888			
Q3. I understand how I can integrate CT in my lessons.	.846			
Q2. CT is integrated in the curriculum of my school.	.846			
Q13. Identifying elements and their relationships within a system is a dimension of CT.		.816		
Q14. CT aims at working with algorithms.		.641		
Q7. CT is the human capability to solve complex problems.		.602		
Q8. CT can only be learned by applying digital tools.			.825	
Q10. CT can only be learned through programming.			.768	
Q9. CT incorporates thinking skills to use computers effectively.			.483	
Q11. CT is a basic skill comparable to reading, writing, calculating, …				.841
Q12. CT is an abstract and logical thought process.				.660

Factor 1 corresponds to practical CT skills development in an educational process. Those teachers who understand well what CT is, have integrated the CT development in their lessons. Factors 2 and 4 show two important aspects of CT: understanding of CT as algorithmic thinking and/or as problem solving, also understanding of CT as basic skills and thought processes. Factor 3 shows limited understanding of CT as depending on the usage of a particular tool (e.g.: CT can be learned only by using digital tools, only through programming, and using computers effectively).

Distribution of teacher opinions within each factor is shown in Figure 5. For the reasons of representation simplicity, we converted the 5-point Likert scale to the 3 groups of positive, neutral and negative answers (“agree”, “neutral”, “disagree”).

The data analysis revealed that 44% of respondents actively integrate CT in their lessons, 33% of teachers are of neutral position, and 24% do not integrate CT at all (Fig. 5, F1). The vast majority of respondents (78%) do agree that CT is understood as an algorithmic thinking and problem-solving ability (Fig. 5, F2). However, 24% of surveyed teachers think that CT is associated with a tool (digital tool, programming only or using a computer only) (Fig. 5, F3). In addition, more than half of the respondents (55%) have no strong opinion on this question, i.e., 79% of the teachers have some level of misunderstanding of CT education. At the same time, 73% of teachers associate CT with literacy and basic skill as reading, writing, and abstract and logical thinking (Fig. 5, F4).

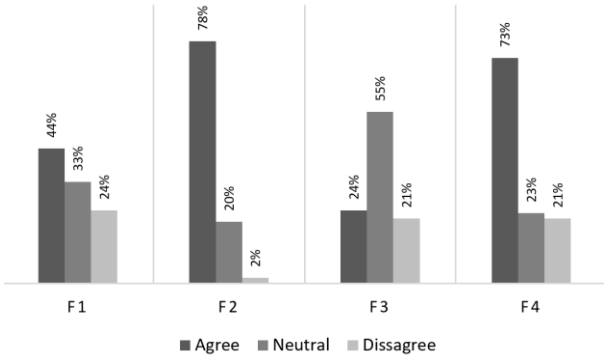


Fig. 5. Teachers' understanding of computational thinking, distributed among factors (N = 110)

Teachers indicated their needs in order to better integrate CT in their class work. This was an answer to the open-ended question of the questionnaire (Q6. Describe your needs to work on computational thinking in class). 72 respondents (65.4%) submitted their answers to this question. Out of qualitative answers, we defined 6 categories of the needs: educational resources, best practice examples, more time allocation, technological support, providing fundamental knowledge, and focus on teacher training. Some teachers indicated needs corresponding to several categories in the same answer (e.g. "More educational resources and technological support"). Major categories are related to methodological support and teacher training: they might be overlapping, but we leave them as separate categories, as they have been derived from the wordings used by the teachers. Analysis of teachers' needs by categories for each factor is presented in Fig. 6.

Respondents positively contributing to Factor 1 *Active integration of computational thinking in their classes* ("agree", 47%, N = 72) mostly need educational resources and technological support (14% out of all respondents who answered this question, or 29% out of those who actively develop CT skills in their lessons) (see Fig. 6, Factor 1). They also need more best practice examples on how to teach CT as well as more teacher training (10% out of all respondents who answered these questions, or 21% out of those who actively develop CT skills in their lessons). Respondents who negatively contribute to Factor 1, i.e., do not apply CT to education in practice ("disagree", 32%, N = 72), indicated their need for fundamental knowledge (10% of all respondents who submitted answer to this question, or 41% out of those who do not develop CT skills in their lessons). They also expressed a need for more educational resources (35%), more best practice examples (29%) and more teacher training (20%). We notice that those teachers, who do not actively develop CT skills of their students, did not indicate that they needed more time allocation or technological support in order to change their practice.

The vast majority of teachers who submitted an answer about their needs are positively contributing to Factor 2 *Computational thinking as algorithmic thinking and problem-solving* (86%). Most of them indicate that more educational resources (28% out of all respondents who answered this question, or 32% out of those who agree on this factor question group), best practice examples (correspondingly, 22% and 26%) and

focus on teacher training (22% and 26%) are needed. Respondents who do not see CT as algorithmic thinking and problem-solving skills, mention time allocation and best practice examples as their need (50% in "disagree" group).

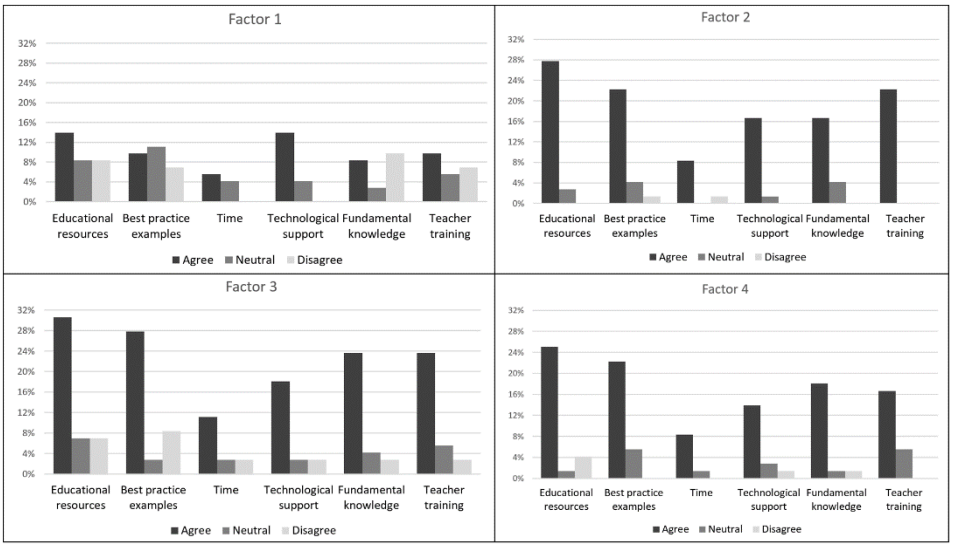


Fig. 6. Teachers' needs, represented by Factors 1–4 (N = 72)

Factor 3 shows limited understanding of CT, i.e., CT is here assumed associated with particular tools. 31% out of those teachers who expressed their needs, do associate CT with particular tools. The pattern of needs is similar to that of Factor 2 (Figure 6, Factor 3). Despite understanding of computational thinking, teachers mostly indicate the need of educational resources, best practice examples, providing fundamental knowledge and focus on teacher training. Time allocation and technological support are not among top level needs. Teachers who do not associate CT with a tool (7%, N = 72), mostly indicate the need of best practice examples and educational resources. Factor 4 shows a similar pattern as the one of Factor 3.

5.4. Limitations of the Research

The limitations of this research are driven by a challenging task to compare implementation of informatics in different countries due to the difference in the education systems. For instance, only 17 out of 52 surveyed countries (33%) introduce informatics in grades 1 or 2. 19% of countries start teaching informatics in grade 5 or 6 while in some countries, grade 5 is already the starting grade for secondary school. There are also differences in school implementation and regional implementation within a single country, as generalized by the experts. It should also be mentioned that the data analyzed here reflects the situation for the time of the survey, i.e. 2019. One of the future research directions is studying further developments of informatics and CT

implementation in primary school comparing to the study results presented in this article.

The next limitation is due to the dynamics of the informatics (CT) integration into primary education and the number of teacher respondents, the size of subgroups (e.g. teacher profession) do not allow to study differences between subgroups. The topic of misunderstanding of CT by the primary school teachers needs further investigation and is also one of the directions for future work.

6. Conclusion and Discussion

Introduction of informatics education in primary schools is a difficult and challenging task. According to the ACM Europe and Informatics Europe strategy [7] the most important challenges are: 1) curriculum development; 2) teacher preparation (education) and training; 3) research of the implementation process and what should be taught. We addressed them in this paper.

Active participation of experts representing 52 countries for quantitative study and 15 countries for qualitative study indicates the importance of the problem. 21% of surveyed countries of quantitative study undergo active development of informatics curriculum for primary education (91% of these countries belong to the European region).

The current tendencies and trends we learnt by answering our first research question are the following.

The results of the quantitative study have shown that CT and concepts related to informatics subject are introduced in the majority of surveyed countries (83%) in primary education. However, there are a lot of differences in the level of its implementation.

Informatics is introduced in grades 1 or 2 in 33% of surveyed countries. In these countries, the most implemented are Digital content skills, and more than 70% of these countries introduce other areas, related to algorithms and programming, problem solving, data and information, virtual communication and safety. The titles of the content areas differ among the countries, but include similar concepts.

At the time of the research, many countries pay priority to have informatics as a separate subject in primary education rather than integrating into other subjects. However, this result cannot be strictly interpreted (see Limitations section). Due to the interdisciplinary nature of primary education, it seems natural to have integrated informatics education. Additional insights from the qualitative research and experts' opinions and recommendations tell us that integration relies to a large degree on teachers' competence and curriculum flexibility. Experts expressed the wish for natural integration into other subjects' topics together with a separated subject.

Still more attention should be paid to primary teacher education and training. The results from the quantitative study show that training in primary teacher education programs is mostly limited to digital literacy in the majority of surveyed countries, even in those who have introduced an informatics-related curriculum in primary education or where such a curriculum is being developed. Additional experts' comments revealed that at the time of this research, more attention is paid to in-service teacher professional development and teacher training than to pre-service teacher education. Furthermore,

informatics, CT and digital literacy courses are included in professional development programs. There are national activities and financial support from the state to take informatics-related courses. Some countries introduce the requirements for primary teachers to accomplish some minimal number of CT and/or informatics courses during a defined period of time. Countries that have an official curriculum of informatics, or where it is being developed, include informatics/CT related modules in the future teachers' study programs. However, the level of training of pre-service teachers differs from university to university even within the same country.

The results of the teachers' study on their understanding or misconceptions related to CT (as posed by the second research question) show that:

- Almost half (44%) of the respondents actively integrate CT in their lessons, but one third (33%) is of a neutral position, and 24% do not integrate CT at all.
- Four factors have been identified, related to integration of CT into the primary school lessons and CT understanding by the teachers: 1) active integration of CT; 2) CT as algorithmic thinking and/or as problem-solving; 3) association of CT with a tool; 4) CT as literacy and thought processes.
- The vast majority of teachers (79%) have some level of misunderstanding of CT education as related to the tool usage. This requires further investigation and confirms again the need of teacher training on CT and its integration in their lessons.
- Time allocation and technological support are not essential for teachers in order to start introducing CT in their lessons. Instead, additional fundamental knowledge and methodological support in the form of educational resources, more practical examples and focus on teacher training is needed. This result confirms the essential need for teacher training and methodological support.

Acknowledgment. We acknowledge all international experts who took part in the survey on informatics in primary education for active participation and collaboration. Part of this research (Phase III) was conducted under the STEAM CT project funded by the Erasmus+ Programme (agreement No 2019-1-BE02-KA201-060222) and TeaEdu4CT (agreement No 2019-1-LT01-KA203-060767). We thank Nicklas Anttu from Turku University, Finland, for proofreading the paper and making suggestions.

References

1. Armoni, M., Gal-Ezer, J.: Early computing education: why? what? when? who? ACM Inroads. Vol. 5, No. 4, 54–59 (2014). DOI: https://doi.org/10.1145/2684721.26847344
2. Australian Computing Academy: Coding and Computational Thinking. What is the evidence? (2016) Retrieved May 12, 2021 from https://education.nsw.gov.au
3. Australian Curriculum: Digital Technologies (2020). Retrieved May 12, 2021 from https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/
4. Balanskat, A., Engelhardt, K.: Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe. European Schoolnet (2015) Retrieved May 12, 2021 from http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf

5. Bocconi, S., Chioccariello, A., Ear, J.: The Nordic approach to introducing Computational Thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group (2018). DOI: https://doi.org/10.17471/54007
6. CAS: Computing in the National Curriculum. A Guide for Primary Teachers (2013). Retrieved May 12, 2021 from https://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf
7. Caspersen, M.E., Gal-Ezer, J., McGettrick, A., Nardelli, E.: Informatics for all: The strategy. ACM Europe & Informatics Europe (2018). Retrieved May 12, 2021 from https://europe.acm.org/binaries/content/assets/public-policy/acm-europe-ie-i4all-strategy-2018.pdf
8. CSTA: K-12 CS standards (2020). Retrieved May 12, 2021 from https://www.csteachers.org/page/standards
9. The Committee on European Computing Education (CECE): Informatics Education in Europe: Are We All in the Same Boat? 2017 Technical Report. ACM, New York, NY, United States (2017). DOI: https://doi.org/10.1145/3106077
10. Dagienė, V.: Resurgence of informatics education in schools: a way to a deeper understanding of informatics concepts. In: Adventures Between Lower Bounds and Higher Altitudes, Lecture Notes in Computer Science, Springer, Cham, Vol. 11011, 522–537 (2018). DOI: https://doi.org/10.1007/978-3-319-98355-4_30
11. Dagienė, V.: Teaching Information Technology in General Education: Challenges and Perspectives. In: From Computer Literacy to Informatics Fundamentals. ISSEP 2005. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Vol. 3422, 53–64 (2005). DOI: https://doi.org/10.1007/978-3-540-31958-0_7
12. Dagienė, V., Jevsikova, T., Stupurienė, G.: Introducing informatics in primary education: curriculum and teachers' perspectives. In: Informatics in Schools. New Ideas in School Informatics. ISSEP 2019, Lecture notes in computer science, Springer, Cham, Vol. 11913, 83–94 (2019). DOI: https://doi.org/10.1007/978-3-030-33759-9_7
13. Denning, P.J.: Remaining trouble spots with computational thinking. Commun. of the ACM Vol. 60, No. 6, 33–39 (2017), DOI: https://doi.org/10.1145/2998438
14. Denning, P.J., Tedre, M.: Computational Thinking. The MIT Press (2019).
15. Duncan, C., Bell, T.: A Pilot Computer Science and Programming Course for Primary School Students. In Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE '15). Association for Computing Machinery, New York, NY, USA, 39–48 (2015). DOI: https://doi.org/10.1145/2818314.2818328
16. European Commission: DigComp into Action. Get inspired, make it happen. A user guide to the European digital competence framework. Joint Research Centre (European Commission) (2018), DOI: http://dx.doi.org/10.2760/112945
17. Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., Liebe, C., Maiorana, F., McGill, M.M., Quille, K.:. An International Study Piloting the MEasuring TeacheR Enacted Computing Curriculum (METRECC) Instrument. In Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR'19). Association for Computing Machinery, New York, NY, USA, 111–142 (2019), DOI: https://doi.org/10.1145/3344429.3372505
18. Fessakis, G., Prantsoudi, S.: Computer Science Teachers' Perceptions, Beliefs and Attitudes on Computational Thinking in Greece. Informatics in Education, Vol. 18, No. 2, 227–258 (2019). DOI: https://doi.org/10.15388/infedu.2019.11
19. Freina, L., Bottino, R., Ferlino, L.: Fostering Computational Thinking skills in the Last Years of Primary School. International Journal of Serious Games, Vol. 6, No. 3, 101–115 (2019).
20. Grgurina, N.: Computational thinking in Dutch secondary education. In Informatics in Schools: Local Proceedings of the 6th International Conference ISSEP 2013 – Selected Papers, 119 (2013). DOI: https://doi.org/10.17083/ijsg.v6i3.304

21. Heintz, F., Mannila, L., Färnqvist, T. A.: Review of Models for Introducing Computational Thinking, Computer Science and Computing in K-12 Education. In: 2016 IEEE Frontiers in Education Conference (FIE). Erie, PA, USA, 1–9 (2016). DOI: 10.1109/FIE.2016.7757410.
22. Heintz, F., and Mannila, L.: Computational thinking for all: an experience report on scaling up teaching computational thinking to all students in a major city in Sweden. ACM Inroads, Vol. 9, No. 2, 65–71 (2018). DOI: https://doi.org/10.1145/3210553.
23. Hubwieser, P., Armoni, M., Giannakos, M., Mittermeir, R. T.: Perspectives and Visions of Computer Science Education in Primary and Secondary (K-12) Schools. ACM Trans. Comput. Educ. Vol. 14, No. 2, Article 7, 9 pages (2014). DOI:https://doi.org/10.1145/2602482
24. Juškevičienė, A., Passey, D.: Outcomes of Computing Education. In: Tatnall A. (eds) Encyclopedia of Education and Information Technologies. Springer, Cham (2019).
25. Mannila L., Dagiene V., Demo B., Grgurina N., Mirolo C., Rolandsson L., Settle A.: Computational Thinking in K-9 Education. In Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference (ITiCSE-WGR '14). Association for Computing Machinery, New York, NY, USA, 1–29 (2014). DOI: https://doi.org/10.1145/2713609.2713610
26. Mannila L., Nordén L. Å., Pears A.: Digital Competence, Teacher Self-Efficacy and Training Needs. In Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18). Association for Computing Machinery, New York, NY, USA, 78–85 (2018). DOI: https://doi.org/10.1145/3230977.3230993
27. Manches, A., Plowman, L.: Computing education in children's early years: A call for debate. British Journal of Educational Technology, Vol. 48, No. 1, 191–201 (2017).
28. Palts, T., Pedaste, M.: A Model for Developing Computational Thinking Skills. Informatics in Education, Vol. 19, No. 1, 113–128 (2020).
29. Papert, S. A.: Mindstorms: Children, computers, and powerful ideas. Basic Books, New York (1980).
30. Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., Belikov, O. M.: Coding in K-8: International Trends in Teaching Elementary/Primary Computing. TechTrends, Vol. 63, No. 3, 311–329 (2019). DOI: https://doi.org/10.1007/s11528-018-0295-4
31. Sysło M. M., Kwiatkowska A. B.: Introducing a New Computer Science Curriculum for All School Levels in Poland. In Informatics in Schools. Curricula, Competences, and Competitions. ISSEP 2015, Vol. 9378, 141–154. Springer, Cham (2015). DOI: https://doi.org/10.1007/978-3-319-25396-1_13
32. Vezis, V., Krumins, O.: New Competencies-Based Curriculum of Computing in General Basic Education Schools in Latvia and its Testing. Baltic Journal on Modern Computing, Vol. 7, No. 3, 364–379 (2019). DOI: https://doi.org/10.22364/bjmc.2019.7.3.0
33. Wing, J.M.: Computational thinking. Commun. ACM, Vol. 49, No. 3, 33–35 (2006). DOI: https://doi.org/10.1145/1118178.1118215
34. Wing, J.M.: Computational Thinking: What and Why (2011). Retrieved May 12, 2021 from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Valentina Dagienė is professor and principal researcher at Vilnius university, Lithuania. She has published over 300 scientific papers and 60 books on computer science education. She established and is Editor of two international journals "Informatics in Education" and "Olympiads in Informatics". She has coordinated over 30 national and international projects on Informatics and STEM education. She is acknowledged by Ada Lovelace Computing Excellence Award by the European Commission's in 2016.

Tatjana Jevsikova holds PhD in computer science and is a senior researcher and associate professor at Vilnius University Institute of Data Science and Digital Technologies. Her main research interests include e-learning, computer science and computational thinking education, teacher training, and cultural aspects of human-computer interaction. She authors more than 35 research papers, a number of methodological papers and educational books.

Gabrielė Stupurienė holds PhD in technological sciences (informatics engineering). She is a research assistant at Vilnius University. Her main research interest is Informatics/Computer science education, computational thinking.

Anita Juškevičienė, PhD in technological sciences (informatics engineering), is a researcher at Vilnius University Institute of Data Science and Digital Technologies. The areas of her scientific interest focus on technology enhanced learning, computational thinking, hands-on activities. She has published a number of scientific papers and publications in popular magazines, participated in a number of large scale EU-funded R&D projects.

Received: December 15, 2020; Accepted: June 01, 2021.

Computer Science and Information Systems 19(1):25–45 https://doi.org/10.2298/CSIS201218041L

Link quality estimation based on over-sampling and weighted random forest

Linlan Liu[1], Yi Feng[3, 1], Shengrong Gao[1], and Jian Shu[2]

[1] School of Information Engineering, Nanchang Hangkong University,
330063 Nanchang, China
765693987@qq.com
1322415547@qq.com
[2] School of Software, Nanchang Hangkong University,
330063 Nanchang, China
shujian@nchu.edu.cn
[3] School of Engineering, Zhejiang Normal University Xingzhi College,
321000 Jinhua, China
458018002@qq.com

Abstract. Aiming at the imbalance problem of wireless link samples, we propose the link quality estimation method which combines the K-means synthetic minority over-sampling technique (K-means SMOTE) and weighted random forest. The method adopts the mean, variance and asymmetry metrics of the physical layer parameters as the link quality parameters. The link quality is measured by link quality level which is determined by the packet receiving rate. K-means is used to cluster link quality samples. SMOTE is employed to synthesize samples for minority link quality samples, so as to make link quality samples of different link quality levels reach balance. Based on the weighted random forest, the link quality estimation model is constructed. In the link quality estimation model, the decision trees with worse classification performance are assigned smaller weight, and the decision trees with better classification performance are assigned bigger weight. The experimental results show that the proposed link quality estimation method has better performance with samples processed by K-means SMOTE. Furthermore, it has better estimation performance than the ones of Naive Bayesian, Logistic Regression and K-nearest Neighbour estimation methods.

Keywords: Wireless Sensor Network, Link Quality Estimation, Weighted Random Forest, Oversampling.

1. Introduction

The Wireless Sensor Network [1] (WSN) is a self-organizing network formed by wireless communication through various inexpensive micro sensor nodes with sensing capabilities, computing power, and communication capabilities. In recent years, WSN has been widely used in different fields, such as military target tracking, natural disaster rescue, biomedical health monitoring, hazardous environment detection, and so on.

WSN is different from traditional network because of its design and limited resources. Sensor nodes are usually deployed in the wild environment with few people. The network is built by self-organization between nodes, and the information in monitoring areas is sent to the sink node through multi hops. The ability of processing, storage and communication of sink nodes are relatively great. They process the received information such as fusion, calculation and storage, and transmit the data to the remote terminal equipment through the external networks.

Wireless communication technology is used to transmit sensing data between sensor nodes. Due to the small communication range and low bandwidth of sensor nodes, a large number of sensor nodes are needed to monitor one area together, so as to obtain relevant data.

Due to the different deployment environments of sensor nodes, there are great differences in the interference received by nodes [2]. In some harsh field environments, interference will seriously affect the communication, resulting in unstable communication links, and data loss in the process of transmission. In the existing routing protocols [3], although there is a packet loss retransmission mechanism, which allows sending nodes to retransmit data packets, for worse quality links, this will cause nodes to repeatedly send data packets, result in seriously consuming node energy. Selecting high quality links for communication through link quality estimation will improve the efficiency of data transmission and reduce the number of retransmissions. For nodes with limited energy storage, this will further reduce the energy consumption of the nodes, thus extending the life cycle of the entire network.

The existing link quality estimation methods mainly use physical layer parameters and link layer parameters to construct a link quality estimation model. However, in the existing methods, the distribution problem of sample data is not considered, so that it will affect the accuracy of link quality estimation, especially when imbalanced samples occur. The existing estimation models tend to be more inclined to the majority of sample data, thus affecting the classification performance of the model on the minority of sample data. Therefore, this paper employs K-means SMOTE to deal with imbalanced sample data. K-means is used to cluster samples. In order to balance samples, SMOTE is employed to increase minority class samples. Make the distribution of link quality samples achieve balance through K-means SMOTE. Benefit from the great effect on the imbalanced data processing of weighted random forest, weighted random forest is adopted to construct the estimation model, which lean the interest towards to the correct classification of rare class, and improve the prediction performance.

Considering the fluctuation and asymmetry existing in the links are fully considered. The mean, variance and asymmetry metrics of the physical layer parameters are selected as the link quality parameters, and a link quality estimation model based on weighted random forest is constructed.

The main contributions of this paper are as follows:

(1) A method of processing imbalanced samples is proposed during the link quality estimation process. Aiming at the problem of link quality imbalanced samples, this paper uses synthetic minority over-sampling technique to deal with imbalanced samples. K-means clustering is used to divide the samples into different clusters, and minority samples is increased by stochastic linear interpolation in each cluster, so that samples of each link quality level is balanced.

(2) A link quality estimation model based on weighted random forest is proposed, which set decision trees with poor classification performance smaller weight and good classification performance bigger weight.

(3) The evaluation metrics suitable for imbalanced samples are employed to propose evaluate models. This paper comprehensively evaluates the performance of the proposed evaluate model by accuracy, recall and F1 value.

The rest of the paper is organized as follows. The state-of-the–art is discussed in Section 2. The selection of link quality parameters is addressed in Section 3. The division of link quality level is described in Section 4. The processing of imbalanced data is presented in Section 5. The link quality estimation model is constructed in Section 6. The experiments and analysis are presented in Section 7. The conclusion is made in section 8.

2. Related Work

Link quality estimation has attracted many scholars to carry out in-depth research, the existing methods fall mainly into the following categories: link quality estimation method based on link characteristics [4], link quality estimation method based on probability estimation theory, and link quality estimation method based on intelligent learning [5].

2.1. Link quality estimation method based on link characteristics

Such methods mainly use the received signal strength indicator (RSSI), link quality indicator (LQI), and signal to noise ratio (SNR) to estimate link quality. In order to solve the link quality problem of the upper communication network in the transmission detection system, the literature [6] analyzes the network characteristics of WSN and selects the optimal next hop node in the routing establishment stage according to the hop count and network environment. Literature [7] proposes a simple, accurate and low-cost link quality estimation technique, which is suitable for WSN scenarios with limited resources. Kalman filtering and fuzzy logic are used to optimize the influence of RSSI and LQI on link quality at low cost. Experimental results show that the method realizes error-free transmission at the cost of less delay.

Literature [8] employs several link property indicators in the fuzzy algorithm, without specific calculation methods and formulas. Literature [9] proposes a new link delay aware energy efficient routing metric, namely, predicted remaining deliveries (PRD), for routing path selection of wireless sensor networks deployed in harsh environments. Literature [10] proposes a link quality metric method based on triangle metric. The link quality can be evaluated quickly and reliably with fewer link detection packets by using geometric methods combined with packet reception rate (PRR), LQI and SNR information.

Literature [11] establishes a generalized model that connects PLR to link quality indicator, a physical layer link quality measure, and packet length under diverse environmental conditions. By rich observations on the spatio-temporal characteristics of

the dependency of packet loss rate (PLR) on LQI and packet length, propose a packet loss rate model as a function of LQI and packet length, that is applicable in all experimented scenarios. A comparison with a literature LQI-only based PLR model shows that the proposed model has higher accuracy for various packet lengths.

2.2. Link quality estimation method based on probability estimation theory

Such methods focus on estimating the packet reception rate at the receiving end in communication. In literature [12], based on the analysis of common lognormal path loss models, the wireless link quality characterized by SNR can be decomposed into two parts with different characteristics, a time-varying nonlinear part and a non-stationary random part. By processing the two parts separately, a new link quality estimation method WNN-LQE is proposed to obtain the confidence interval of link quality. In literature [13], the mathematical model of exponentially weighted moving average (EWMA) and link quality prediction are used to solve the problem of unstable packet transmission rate. The experimental results show that the correlation is established in the EWMA model. Literature [14] proposes a three-layer impulse response framework to analyze the time fading effect of fixed wireless links in industrial environment. The original observation of link quality is mapped to the distributed parameter space. In the new space, the measurement noise has a constant covariance, meeting the requirements of linear Kalman filtering. Based on this, an enhanced Kalman filter is designed, which can obtain better link quality prediction effect in industrial environment.

Estimating the quality of a link is a key primitive in WSNs, as upper layers use this piece of information in making performance-critical decisions. Literature [15] proposed Rep, a novel sampling scheme able to extract the link quality from the packet repetitions of low-power preamble sampling MACs. The experiments show that Rep reduces the energy and traffic used for link estimation by one order of magnitude, and increases the speed of the process by one order of magnitude, while maintaining state-of-the-art accuracy.

Considering the selection of high quality path in mobile ad-hoc network is a critical issue due to mobility of nodes and variable channel conditions during data transmission, Literature [16] proposed routing protocol named as modified expected transmission count enabled ad-hoc on-demand distance vector (MXAODV). Select the path with the smallest number of hops as the next hop of the mobile ad hoc network, and modify the expected transmission number at the same time. Extensive simulation has been done to analyze the performance of MXADOV. Significant improvements found in terms of throughput, packet delivery fraction, normalized routing load and end-to-end delay.

2.3. Link quality estimation method based on intelligent learning

Such methods are mainly modeled by intelligent learning methods such as machine learning and pattern matching.

Literature [17] proposed a lightweight, fluctuation insensitive multi-parameter fusion link quality estimator, which have characteristics of high flexibility and low overhead.

Signal-to-Noise Ratio and Link quality indicator are preprocessed by exponential weighted Kalman filtering. These two parameters are fused using lightweight weighted Euclidean distance. Link quality is estimated quantitatively with the mapping model of the fused parameter and packet reception ratio, which is constructed by logistic regression. Experimental results show that the proposed estimator could reflect link quality more realistically. Compared with some similar estimators, estimate error of the proposed one is reduced by 18.32% to 60.11%.

Literature [18] uses naive Bayes (NB), logistic regression (LR), artificial neural networks (ANN) to build a prediction model. By combining link layer parameter PRR and physical layer parameters RSSI, LQI and SNR, link quality is predicted. Literature [19] uses RSSI and LQI as estimation parameters. It divides link quality into five grades according to PRR, and establishes a multi-classification link quality estimation mechanism based on support vector machine. Literature [20] proposes a missing data estimation algorithm based on k-nearest neighbor (KNN), which uses a linear regression model to describe the spatial correlation of data from different sensor nodes and uses data information from multiple neighbor nodes to estimation missing data. Literature [21] proposes a link quality estimation algorithm based on stacked autoencoder. The zero-filling method is developed to process the original missing link information. The SAE model is used to extract the asymmetric characteristics of the uplink and downlink.

The estimation method based on the physical layer and link layer parameters only estimates the link quality from a single perspective and cannot fully reflect the link characteristics. The estimation method based on machine learning adopts a data-driven approach, which constructs a learning model to mine the relationship between link data and link quality by collecting a large amount of link quality data.

Considering the fluctuation and asymmetry of the link, this paper selects the mean, variance and asymmetry indicators of the physical layer parameters as the link quality parameters and determines the link quality level according to the PRR so as to estimate the link quality. Since sensor nodes are often deployed in harsh environments, they are subject to environmental noise during communication, which makes the link quality worse. Samples of good link quality and bad link quality account for a small proportion, and samples of medial link quality account for a big proportion. So, the samples show imbalance. If the samples are directly used for training, the model will produce deviation. Therefore, the samples need to be processed before training. In this paper, K-means SMOTE is used to preprocess the samples to reduce the imbalance. And a link quality estimation model based on weighted random forest classification (WRF) algorithm is constructed to estimate the link quality.

3. Selection of Link Quality Parameters

In the process of link quality estimation in this paper, the link quality level is estimated according to the physical layer parameters. The relationship between the physical layer parameters and link quality levels can be mined through the link quality estimation model. Reasonable link quality parameters can better characterize the status of links, so as to improve the effect of the estimation model.

WSN is often deployed in harsh environment. It is easy to be affected by the environment in the communication, causing instability of communication links and making link quality volatility and asymmetry [22]. The physical layer parameters RSSI, LQI, and SNR can quickly detect link changes, which can reflect the sensitivity of the link. So, we choose physical layer parameters as link quality parameters. The asymmetry level (ASL) of the link is reflected by difference between physical layer parameters of uplink and downlink, and the stability of the link is reflected by the variance of the physical layer parameters. There are defined as follows:

$$Input = [\overline{PHY}, \sigma^2(PHY), ASL(PHY)] \tag{1}$$

Where

$$\begin{aligned} ASL &= \left| PHY_{up} - PHY_{down} \right| \\ PHY &\subset (RSSI, LQI, SNR) \end{aligned} \tag{2}$$

4. Division of Link Quality Level

We take link quality level to measure link quality. Literature [23] divides the link quality into three levels according to the PRR value, which are good link, middle link and bad link. Experiments show that the average number of consecutive lost packets in good links is 1.6, the number of consecutive lost packets in middle links and poor links is 5.3 and 56.8, respectively. The link can be well distinguished by the PRR value. In this paper, the link quality level is divided by the same standard, and the link quality is divided into three levels by the PRR value. The specific division criteria are shown in Table 1.

Table 1. Divide Standard.

Link quality level	Link status	PRR
Level 1	Good Link	[80,100]
Level 2	Middle Link	[20,80)
Level 3	Bad Link	[0,20)

5. Processing of Imbalanced Data

The experiments are conducted in parking, indoor and forest scenarios. Link quality samples in different scenarios with different interferes have different distribution characteristics. In some real-world scenarios, the link quality samples of a certain level account for a small proportion of the total sample set is small, which leads to the imbalance of link quality samples distribution.

Due to the imbalance of link quality samples, if the samples are directly used for training, there will be a certain deviation to the classification model, which will make the model more inclined to the majority class samples and eventually lead to the decline of the learning performance of the classification model. For the processing of imbalanced

samples [24], the distribution of imbalanced samples can be changed through data sampling to reduce the degree of data imbalance. Commonly used methods mainly include over-sampling [25] and under-sampling [26]. Over-sampling improves the classification performance of the model for minority class by adding the number of minority class samples, while under-sampling reduces the imbalance of data by reducing the number of classes of majority class samples. Since under-sampling will delete data, it may cause some important information of majority class samples to be lost. When there are few minority class samples, the distribution of samples will be balanced by under-sampling, which will cause a too small data set and further limit the performance of the classifier. The commonly used over-sampling method is random over-sampling, which replicates a small number of minority class samples randomly, thereby increasing the number of minority class samples and changing the degree of data imbalance. However, it does not add additional information to minority class samples, which increases the training time and easily leads to over-fitting of the model.

Chawla [27] et al. proposed SMOTE in 2002. This method not only replicates existing minority class samples, but also creates synthesizes samples by interpolation, which can avoid the over-fitting risk of random over-sampling. During the execution of SMOTE algorithm, a sample a is randomly selected from minority class samples, and a sample b is randomly selected from its nearest neighbor. And then, random linear interpolation is performed between samples a and b, namely new sample $x=a+w*(b - a)$, where is the random weight between (0, 1).

SMOTE algorithm still has some deficiencies in dealing with within-class imbalance and noise. There are mainly two problems. First, when randomly selecting minority class samples for uniform over-sampling, the problem of between-class imbalance can be effectively solved, but the problem of within-class imbalance is ignored. In areas where minority class samples are dense, the number of samples will further increase, while in areas where minority class samples are sparse, the samples are still sparse. Second, SMOTE algorithm may further amplify the noise existing in the data. When the selected sample a is noise in majority class samples, the linear interpolation is performed by selecting the nearest neighbor, and the resulting synthetic sample is likely to be noise, which will reduce the classification performance of the trained model.

K-means SMOTE algorithm [28] combines K-means clustering and SMOTE algorithm to avoid noise generated by carrying out over-sampling in the clustering area and solves the problem of between-class imbalance and within-class imbalance. Due to the adoption of K-means clustering, when the samples synthesis is performed in the cluster region, the sparse samples distribution region synthesizes more samples than the dense samples region, thus solving the within-class imbalance problem.

K-means SMOTE algorithm includes three steps: clustering, filtering and over-sampling. In the clustering step, the entire data is clustered into clusters by the K-means algorithm. The filtering step selects clusters with a high proportion of minority class samples, determines the number of synthetic samples assigned to each cluster, and assigns more synthetic samples to clusters with sparse sample distribution. In the over-sampling step, SMOTE is applied to the selected cluster to achieve a balance between the majority and minority samples.

K-means is a commonly used unsupervised clustering algorithm in data mining. For a given sample set $D=\{x_1, x_2, \mathrm{L}\ , x_m\}$, randomly selecting k samples from D as initial

mean vectors $\{\mu_1, \mu_2, \mathrm{L}\ , \mu_k\}$, calculating the distance $d_{ji} = \left\| x_j - \mu_i \right\|_2$ between the sample $x_j (j = 1, 2, \mathrm{L}\ , m)$ and each mean vector $\mu_i (1 \le i \le k)$, and dividing the sample x_j into clusters with the smallest d_{ji} according to the size of d_{ji}. In each cluster, calculate new mean vector μ_i' and replace the original mean vector until current mean vector is not updated or the maximum number of iterations is reached. Finally, the clustered $C = \{c_1, c_2, \mathrm{L}\ , c_k\}$ is obtained.

The filtering step selects clusters to be over-sampling and determines the number of samples to be generated in each cluster. The selection of clusters is determined according to the proportion of minority class samples and majority class samples in each cluster, and clusters with at least 50% minority class samples are selected. The cluster with a higher proportion of minority class samples can also be selected through the hyperparameter imbalance rate threshold of K-means SMOTE algorithm. The default value is 1, and the imbalance rate threshold of cluster c_i is defined as:

$$irt = \frac{majority\ count(c_i) + 1}{minority\ count(c_i) + 1} \tag{3}$$

In order to determine the generating sample number of each cluster, it is necessary to assign sampling weights to the selected clusters and assign bigger weights to the clusters with sparse samples to generate more samples. The calculation steps of sampling weights are as follows:

1) For each filtered cluster f, the Euclidean distance matrix between minority class samples is calculated, ignoring majority samples.

2) The average distance of each cluster is obtained by calculating the average value of non-diagonal elements in the distance matrix.

3) Calculate the density of minority class samples in the cluster.

$$density(f) = \frac{minority\ count(f)}{average\ minority\ distance(f)^m} \tag{4}$$

Where m is the number of features.

4) Calculate the sparsity of minority class samples in a cluster.

$$sparsity(f) = \frac{1}{density(f)} \tag{5}$$

5) The sampling weight of each cluster is defined as the sparsity of the cluster divided by the sum of the sparsity of all clusters, and the sum of the sampling weights of all clusters is 1.

In the over-sampling step, SMOTE is used for over-sampling in the selected clusters, and the number of samples to be generated for each cluster is $\left\| sampling\ weight(f) \times n \right\|$, where n is the total number of samples to be generated. The process of imbalanced link quality samples processing based on k-means SMOTE is shown as Algorithm 1.

Algorithm 1 Imbalanced sample processing algorithm

Input: input space $Z = \{(\overline{PHY}_i, \sigma^2(PHY_i), ASL(PHY_i)), lv_i\}$, number of clusters k, imbalance rate threshold irt, nearest neighbor $knn = 20$ **Error! Reference source not found.**.

Output: The synthesized link quality sample after the over-sampling.

begin

clusters $\leftarrow$ $K\text{-}means(\overline{PHY}_i, \sigma^2(PHY_i), ASL(PHY_i))$;

filtered cluster $\leftarrow \varnothing$;

for C $\in$ clusters **do**

Calculate imbalance ratio of link quality samples by eq(3);

If imbalance ratio < irt **then**

filtered cluster $\leftarrow$ filtered cluster $\cup$ {C};

end

end

for $f \in$ filtered cluster **do**

average minority distance $\leftarrow$ $mean(euclidean\ distance(f))$;

Calculate $density(f)$ by eq(4);

Calculate $sparsity(f)$ by eq(5);

end

sparsity sum $\leftarrow$ $\sum_{f \in filtered\ clusters} sparsity\ (f)$;

sampling weight $\leftarrow$ $\frac{sparsity(f)}{sparsity\ sum}$;

generated link quality samples $\leftarrow \varnothing$;

for $f \in$ filtered cluster **do**

number of samples $\leftarrow$ $\|sampling\ weight(f) \times n\|$;

generated link quality samples $\leftarrow$ generated samples $\cup$ $SMOTE(f, number\ of\ samples, knn)$;

end

return generated link quality samples

end

6. Link Quality Estimation

In different experimental scenarios, the link quality level distribution of the collected sample data is imbalanced. The K-means SMOTE method is used to randomly linearly interpolation on the original samples to increase the number of minority class samples, and improves the classification accuracy of the estimation model for minority class samples.

6.1. Construct estimation model

Link quality estimation is a process of estimating link quality level based on selected link quality parameters, which is essentially a multi-classification problem. Random forest algorithm [29] is not prone to over-fitting in the training process, and can identify overlapping samples between classes, which is suitable for multi-classification problems. Two random processes are introduced into the random forest. One is to randomly extract samples by Bootstrap resampling method when constructing training subsets to increase the differences among the subsets. The second is to randomly select features from the total number of features to construct the decision tree in order to ensure the diversity of the decision tree and reduce the similarity between the trees. Since the randomness is guaranteed by the two random processes, the decision tree may not be pruned during construction process, and it can also ensure that the random forest is not prone to over-fitting.

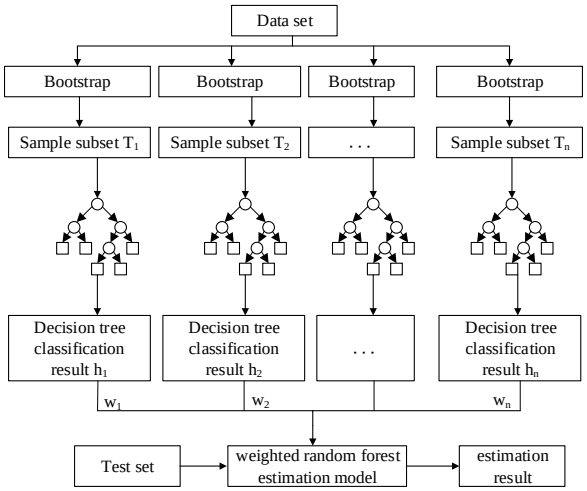


Fig. 1. Link quality estimation model based on weighted random forest.

The traditional random forest algorithm produces the final result through combined voting and gives the same weight to each decision tree. This method gives a too big weight to the decision tree with low classification performance, leading to reducing the overall classification performance. In this paper, a WRF [30] algorithm is applied to construct a link quality estimation model. The model structure is shown in Figure 1. WRF is an improved algorithm for random forests. It assigns smaller weights to decision trees with low classification performance and bigger weights to decision trees with high classification performance.

Let the training sample set processed by K-means SMOTE be $T = \{(x_i, y_i)\}$,where $x_i = \{\overline{PHY}_i, \sigma^2(PHY_i), ASL(PHY_i)\}$, $y_i = \{lv_i\}$, $i = 1, 2, ..., N$, N is the total number of samples, x_i is the vector composed of the selected link quality parameters, and y_i is the link quality level value. In the training process of WRF, the training sample set is split

into different training subsets $T_1, T_2, \mathrm{L}\ , T_n$ by Bootstrap resampling method, and the training process for each subset is independent of each other and does not affect each other. The link quality decision trees are constructed for the training subsets, shown in Fig.1. The key to decision tree learning is to select the optimal partitioning attribute. ID3 decision tree learning algorithm selects attributes with large information gain when dividing nodes, but the information gain criterion will bring certain deviation. For attributes with a large number of values, the corresponding information gain is larger. In order to reduce this influence, C4.5 decision tree algorithm selects attributes with large gain rate when dividing nodes. In this paper, C4.5 algorithm is used in the process of decision tree construction, that is, gain rate is used to select features when dividing attributes. The calculation process of gain rate is as follows:

1) Calculating information entropy

$$Ent(T) = \sum_{k=1}^{|y|} p_k \log_2 p_k \tag{6}$$

Where T is the current sample set, p_k is the proportion of the k th sample in the sample set, and $|y|$ is the number of categories of the sample set, which is the number of link quality levels in this paper.

2) Calculating information gain

$$Gain(T,a) = Ent(T) - \sum_{v=1}^{V} \frac{|T^v|}{|T|} Ent(T^v) \tag{7}$$

Where T^v is the subsets according to attribute a, V is the number of divided subsets, $|T^v|$ is the number of samples in the subsets, $|T|$ is the total number of samples, and $Ent(T^v)$ is the information entropy calculated from subsets $|T^v|$.

3) Calculating gain rate

$$Gain\_ratio(T,a) = \frac{Gain(T,a)}{IV(a)} \tag{8}$$

Where,

$$IV(\mathrm{a}) = -\sum_{v=1}^{V} \frac{|T^v|}{|T|} \log_2 \frac{|T^v|}{|T|} \tag{9}$$

A decision tree is constructed for each training subset separately, and when selecting the optimal partition attribute, the attribute with the largest gain rate is selected. Since the training and classification processes of each decision tree are independent of each other, the construction and classification processes of the decision tree can be parallelized so as to save program running time.

In order to determine the weight of each decision tree, WRF divides the samples into training sets and test sets at a ratio of 3:1. The out-of-bag data is employed to estimate the accuracy of the decision tree, and to calculate the voting weight w_j of the decision tree. In the implementation process of WRF, the training set includes 75% of the initial samples. For each decision tree, about 50% of the in-bag data is used to train the decision tree, and 25% of the data is used to evaluate the classification performance of the decision tree and to calculate the voting weight.

$$w_j = \frac{X_j^{correct}}{X_n}, j = 1,2,\mathrm{L}\ ,n \tag{10}$$

Where $X_j^{correct}$ is the number of out-of-bag samples with the correct classification, and X_n is the total number of out-of-bag samples.

After the weight of decision tree is calculated on the training set, the estimation model is applied on the test set. For each test sample, the output sequence $\{h_1(X), h_2(X), \mathrm{L}\ , h_n(X)\}$ of n decision tree classifiers can be obtained, and the weight w_j of decision tree is used to construct a combined classification model.

$$H(x) = \arg\max_Y \sum_{i=1}^{n} w_j \cdot I(h_i(x) = Y) \tag{11}$$

Where $H(x)$ is the combined classification model, $h_i(x)$ is the i-th decision tree classification model, Y is the output variable, which is the value of link quality level, and $I(\cdot)$ is the indicative function.

6.2. Evaluation Metrics of models

Accuracy is often used as the evaluation metric of link quality estimation models. But for models with imbalanced samples, accuracy does not well reflect the performance of them. For example, there is a test set of 1000 samples including 100 negative samples. If a model classifies all samples into positive, then the accuracy of the model is 90%. From the view of accuracy, it can be seen that the estimation effect of the model is very good, but the model does not identify even one negative sample. So such model has no meaning. Therefore, this paper uses the precision, recall and F1 values to evaluate the performance of proposed model.

For the two-class classification problem, the combination of the real class of the sample and the predicted class of the classifier has four cases: true positive, false positive, true negative and false negative. The confusion matrix formed by the classification results is shown in Table 2.

Table 2. Confusion Matrix**.**

Real class	Predict result	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Confusion matrix can be used to evaluate the precision and recall of classifiers.

$$precision = \frac{TP}{TP + FP} \tag{12}$$

$$recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{14}$$

7. Experiments and Analysis

In the experiments, TelosB nodes of Crossbow Company and the wireless sensor network link quality test platform were used to collect the required data, and K-means SMOTE algorithm and WRF estimation model were implemented through python platform.

7.1. Experimental Scene Setting

Three experimental scenarios are set up, namely, forest, indoor and campus parking. In each experimental scenario, a small star link quality testing network is deployed to collect corresponding link quality data. The experimental parameter settings are shown in Table 3.

Table 3. Experimental Parameter Setting.

Parameter attribute	Parameter value
Transmit power /dBm	31
Channel	26
Number of probe packets	30
Packet transmission rate	5
Transmission period/ms	200
Test period /s	10

7.2. Analysis of experimental results

In the experiments, the training set and the test set are randomly distributed at a ratio of 3:1. For the data in the training set, K-means SMOTE is used to balance samples. WRF is used to construct a link quality estimation model for the original samples and the samples processed by K-means SMOTE respectively. The evaluation confusion matrix diagrams under different scenarios are shown in Figure 2 to Figure 7.

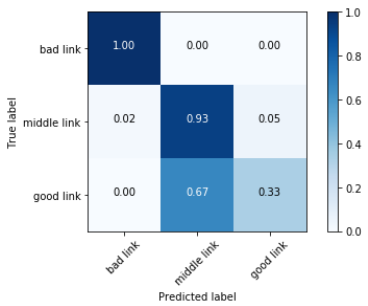


Fig. 2. Confusion matrix of original samples in parking scenes.

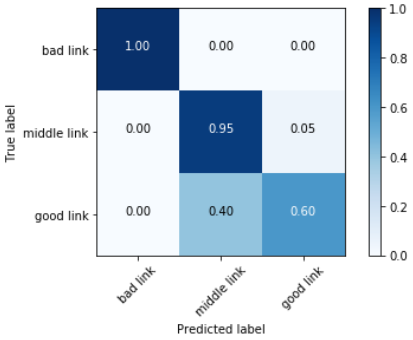


Fig. 3. Confusion matrix of K-means SMOTE processing samples in parking scenes.

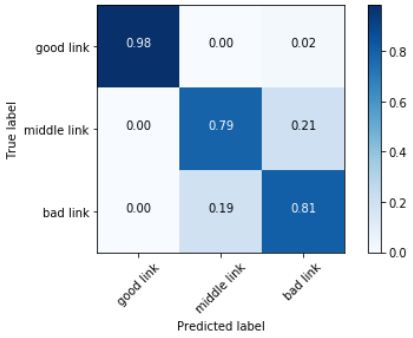


Fig. 4. Confusion matrix of original samples in indoor scenes.

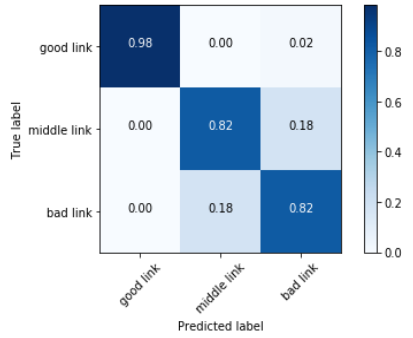


Fig. 5. Confusion matrix of K-means SMOTE processing samples in indoor scenes.

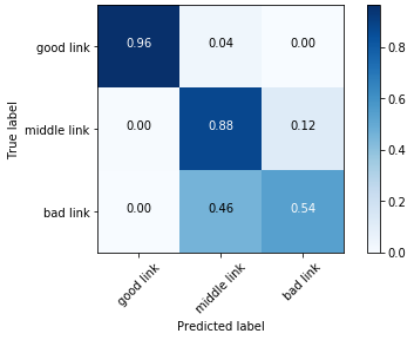


Fig. 6. Confusion matrix of original samples in forest scenes.

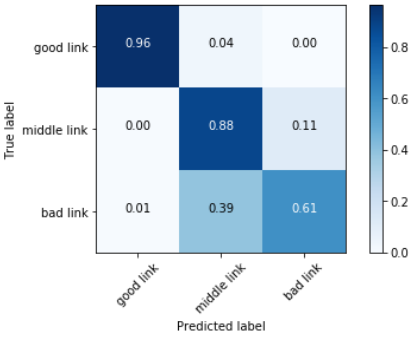


Fig. 7. Confusion matrix of K-means SMOTE processing samples in forest scenes.

It can be seen from Figure 2 and Figure 3 that the estimation accuracy of good link in the parking scene is 33% for the original samples, 60% after K-means SMOTE processing, and the estimation accuracy is increased by 27%. In the indoor scene, the estimation accuracy for middle link has increased by 3%, and in the forest scene, the estimation accuracy for bad link has increased by 7%. The estimation results of three experimental scenes show that the data processed by K-means SMOTE can obviously improve the estimation effect of the model on minority class samples.

In order to further reflect the estimation effect of the model, we calculate the precision, recall and FI value of the original samples and the samples processed by K-means SMOTE under different scenes respectively.

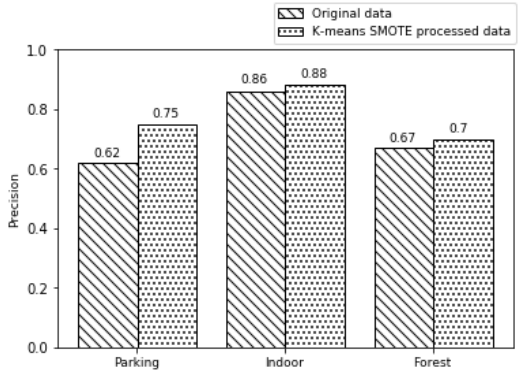


Fig. 8. Comparison of precision.

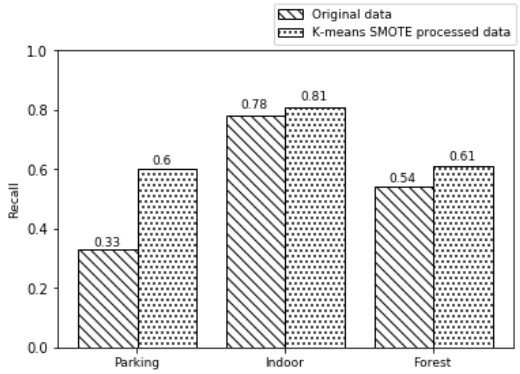


Fig. 9. Comparison of recall.

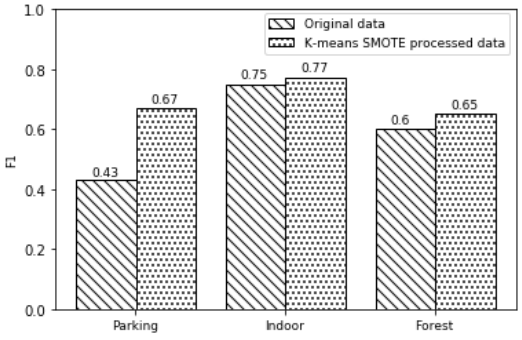


Fig. 10. Comparison of F1.

Figure 8 to Figure 10 show the comparison of precision, recall and F1 of the original data and K-means SMOTE processed data under different scenes respectively. As can be seen from the Figures, in the parking scene, the precision, recall and F1 value of the data processed by K-means SMOTE have increased by 13%, 27% and 24%, respectively, with the most obvious performance improvement. In other scenes, the evaluation metrics of K-means SMOTE with the data processed are improved compared with ones of the model with the original data, indicating that K-means SMOTE has good performance in dealing with imbalanced data.

In order to further verify the estimation performance of WRF, this paper uses the data, processed by K-means SMOTE to train WRF model, and compares the trained WRF model with NB, LR and KNN models in three experimental scenarios.

The comparison results of precision, recall and F1 in each scenario are shown in Figure 11 to Figure 13.

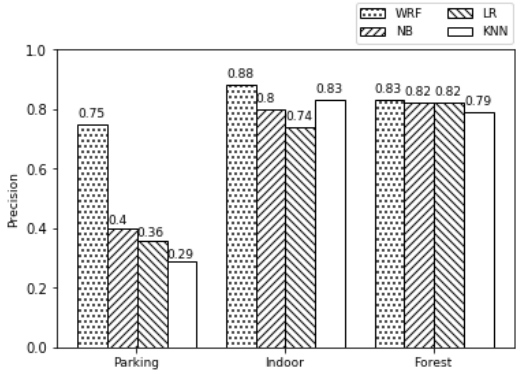


Fig. 11. Precision comparison result 1.

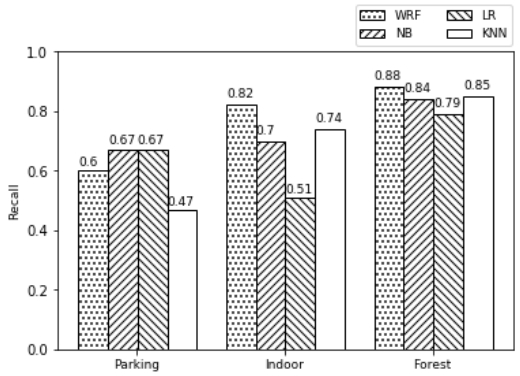


Fig. 12. Recall comparison result 1.

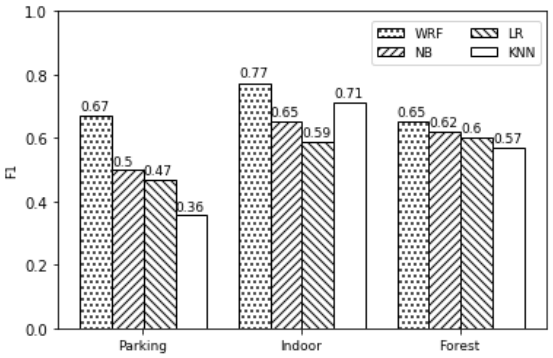


Fig. 13. F1 comparison result 1.

As can be seen from Figure 11 to Figure 13, the precision of the WRF model is 35% bigger than that of the LR model in the parking scene, and the precision and F1 of the WRF model in the three scenes are bigger than those of the other three models. The recall values in the indoor scene and the forest scene are also bigger than that of the NB, LR and KNN models. The experimental results show that the WRF has good estimation performance in different experimental scenes.

8. Conclusion

The main work of this paper is as follows: The average, variance and asymmetry metric of physical layer parameters are selected as link quality parameters, and link quality level are divided according to PRR values to estimate link quality. K-means SMOTE is applied to generate minority samples, and the samples are divided into different clusters by K-means, thus noise data can be avoided. For different clusters, minority class samples are generated by random linear interpolation in each cluster, so that the number of minority class samples is increased, thus solving the imbalance of link data. This paper constructs a link quality estimation model based on WRF, assigns smaller weights to decision trees with low classification performance and bigger weights to decision trees with high classification performance in the combined model, and uses out-of-bag data to evaluate the accuracy of the decision trees. Precision, recall and F1 are used to evaluate the performance of proposed model. The experimental results in three scenarios show that the proposed model with samples processed by K-means SMOTE presents better performance. Compared with NB, LR and KNN estimation models, WRF model has better estimation performance.

Acknowledgements: This work was supported by the National Natural Science Foundation of China (Grant No. 61962037, 61762065, 61363015, 61501218), the Natural Science Foundation of Jiangxi Province (Grant No. 20171ACB20018, 20171BAB202009, 20181BAB202015).

References

1. Buşoniu, L., Babuška, R., Schutter, B. D.: Multi-agent Reinforcement Learning: An Overview. Innovations in Multi-Agent Systems and Applications, Vol. 38, No. 2, 156-172. (2010)
2. Babuška, R., buşoniu, L., and De Schutter, B.: Reinforcement learning for multi-agent systems. IEEE International Conference on Emerging Technologies and Factory Automation. IEEE, 1-7. (2006)
3. Liu, Z., Wang, F.: Scale-free topology for wireless sensor networks with energy efficient characteristics, Journal of Beijing University of Posts and Telecommunications, Vol. 38, No. 1, 87-91. (2015)
4. Cao, N., Liu, P., Li, G., Zhang, C.: Evaluation models for the nearest closer routing protocol in wireless sensor networks, IEEE Access, Vol. 6, No. 1, 77043-77054. (2018)
5. Gao, D., Zhang, S., Zhang, F.: RowBee: A routing protocol based on cross-technology communication for energy-harvesting wireless sensor networks, IEEE Access, Vol. 7, No. 1, 40663-40673. (2019)
6. Lowrance, C. J., Lauf, A. P.: Link quality estimation in ad hoc and mesh networks: A survey and future directions, Wireless Personal Communications, Vol. 96, No. 1, 475-508. (2017)
7. Bote-Lorenzo, M. L., Gómez-Sánchez, E., Mediavilla-Pastor, C.: Online machine learning algorithms to predict link quality in community wireless mesh networks, Computer Networks, Vol. 132, No. 1, 68-80. (2018)
8. Lu, J., Zhu, Y., Xu, Z.: A reliable wireless sensor network routing method for power transmission line monitoring, Power System Technology, Vol. 41, No. 2, 644-650. (2017).
9. Jayasri, T., Hemalatha, M.: Link quality estimation for adaptive data streaming in WSN, Wireless Personal Communications, Vol. 94, No. 3, 1543-1562. (2017)
10. Baccour, N., Koubâa, A., Youssef, H.: F-lqe: A fuzzy link quality estimator for wireless sensor networks, in Proc. 2010 European Conference on Wireless Sensor Networks, Coimbra, Portugal, 240-255. (2010)
11. Lai, X., Ji, X., Zhou, X.: Energy efficient link-delay aware routing in wireless sensor networks, IEEE Sensors Journal, Vol. 18, No. 2, 837-848. (2018)
12. Boano, C. A., Zúniga, M. A., Voigt, T.: The triangle metric: Fast link quality estimation for mobile wireless sensor networks, in Proc. 19th International Conference on Computer Communications and Networks, Zurich, Switzerland, 1-7.(2010)
13. Zhang, Y., Fu, S., Jiang, Y.: An LQI-based packet loss rate model for IEEE 802.15.4 links, in Proc. IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications(PIMRC), Bologna, Italy, 1-7.(2018)
14. Sun, W., Lu, W., Li, Q.: WNN-LQE: Wavelet-neural-network-based link quality estimation for smart grid WSNs, IEEE Access, Vol. 5, No. 1, 12788-12797. (2017)
15. Mi, X., Zhao, H., Zhu, J.: Research on EWMA based link quality evaluation algorithm for WSN, in Proc. 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, Harbin, China, 757-759.(2011)
16. Qin, F., Zhang, Q., Zhang, W.: Link quality estimation in industrial temporal fading channel with augmented Kalman filter, IEEE Transactions on Industrial Informatics, Vol. 15, No. 4, 1936-1946. (2019)
17. Rojas, C., Decotignie, J.: Leveraging MAC preambles for an efficient link estimation, in Proc. International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Limassol, Cyprus, 1-10.(2018)
18. Sharma, A., Bansal, A., Rishiwal, V.: Selection of high quality path through MXAODV in mobile ad-hoc network, International Journal of Systems Control and Communications, Vol. 10, No. 1, 1-17. (2019)

19. Liu, W., Xia, Y., Luo, R.: Lightweight, fluctuation insensitive multi-parameter fusion link quality estimation for wireless sensor networks, IEEE ACCESS, Vol. 8, No. 1, 28496-28511. (2020)
20. Liu, T., Cerpa, A. E.: Data-driven link quality prediction using link features, ACM Trans on Sensor Networks, Vol. 10, No. 2, 1-35. (2014)
21. Shu, J., Liu, S., Liu, L.: Research on link quality estimation mechanism for wireless sensor networks based on support vector machine, Chinese Journal of Electronics, Vol. 26, No. 2, 377-384. (2017)
22. Pan, L., Li, J.: K-nearest neighbor based missing data estimation algorithm in wireless sensor networks, Wireless Sensor Network, Vol. 2, No. 2, 115-122. (2010)
23. Luo, X., Liu, L., Shu, J., AL-KALI, M.: Link quality estimation method for wireless sensor networks based on stacked autoencoder, IEEE Access, Vol. 7, No. 1, 21572-21583. (2019)
24. Baccour, N., Koubâa, A., Youssef, H.: Reliable link quality estimation in low-power wireless networks and its impact on tree-routing, Ad Hoc Networks, Vol. 27, No. 1, 1-25. (2015)
25. Bildea, A., Alphand, O., Rousseau, F., Duda, A.: Link quality estimation with Gilbert-Elliot model for wireless sensor networks, in Proc. IEEE 26th Annual Int. Symp. Personal, Indoor, and Mobile Radio Communications (PIMRC), Hong Kong, China, 2049-2054.(2015)
26. Zhu, M., Xia, J., Jin, X., Yan, M., Cai, G., Yan, J., Ning, G.: Class weights random forest algorithm for processing class imbalanced medical data, IEEE Access, Vol. 6, No. 1, 4641-4652. (2018)
27. Galar, M., Fernandez, A., Barrenechea, E.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 42, No. 4, 463-484. (2012)
28. Batista, G. E., Prati, R. C., Monard, M. C.: A study of the behavior of several methods for balancing machine learning training data , ACM SIGKDD explorations newsletter, Vol. 6, No. 1, 20-29. (2004)
29. Chawla, N. V., Bowyer, K. W., Hall, L. O.: SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, Vol. 16, No. 1, 321-357. (2002)
30. Douzas, G., Bacao, F., Last, F.: Improving imbalanced learning through a heuristic oversampling method based on K-means and SMOTE, Information Sciences, Vol. 465, No. 1, 1-20. (2018)
31. Liaw, A., Wiener, M.: Classification and regression by random forest, R News, Vol. 2, No. 3, 18-22. (2002)
32. Winham, S. J., Freimuth, R. R., Biernacka, J. M.: A weighted random forests approach to improve predictive performance, Statistical Analysis and Data Mining: The ASA Data Science Journal, Vol. 6, No. 6, 496-505. (2013)

Linlan Liu born in 1968, and received the Bachelor degree in computer science from the National University of Defense Technology, Changsha, China, in 1988. Currently she is a full Professor, Internet of Things Technology Institute, Nanchang Hangkong University, Nanchang, China. She was a Visiting Scholar at Wilfrid Laurier University, Waterloo, Ontario, Canada. She has authored/coauthored more than 70 papers. Her research interests include wireless sensor networks and embedded system (765693987@qq.com).

Yi Feng born in 1995. She received the Master degree from Nanchang Hangkong University, Nanchang, China, in 2020. She is now with the Department of School of

Engineering, Zhejiang Normal University Xingzhi College, Jinhua, China. Her research interests include wireless sensor networks. (458018002@qq.com).

Shengrong Gao born in 1994. He received the Master degree from Nanchang Hangkong University, Nanchang, China, in 2019. His research interests include wireless sensor networks. (1322415547@qq.com).

Jian Shu born in 1964. He received the M.Sc.degree in computer networks from Northwestern Polytechnical University. He is currently a Professor with Nanchang Hangkong University. His research interests include wireless sensor networks, embedded systems, and software engineering. (shujian@nchu.edu.cn).

Received: December 18, 2020; Accepted: May 22, 2021.

Computer Science and Information Systems 19(1):47–63 https://doi.org/10.2298/CSIS201221043N

Comparative Analysis of HAR Datasets Using Classification Algorithms

Suvra Nayak[1], Chhabi Rani Panigrahi[1], Bibudhendu Pati[1], Sarmistha Nanda[1], and Meng-Yen Hsieh[2]

[1] Department of Computer Science, Rama Devi Women's University, Bhubaneswar, India
{suvra.nayak24, panigrahichhabi, patibibudhendu, sarmisthananda}@gmail.com
[2] Department of Computer Science & Information Engineering, Providence University, Taiwan
mengyen@gm.pu.edu.tw

Abstract. In the current research and development era, Human Activity Recognition (HAR) plays a vital role in analyzing the movements and activities of a human being. The main objective of HAR is to infer the current behaviour by extracting previous information. Now-a-days, the continuous improvement of living condition of human beings changes human society dramatically. To detect the activities of human beings, various devices, such as smartphones and smart watches, use different types of sensors, such as multi modal sensors, non-video based and video-based sensors, and so on. Among the entire machine learning approaches, tasks in different applications adopt extensively classification techniques, in terms of smart homes by active and assisted living, healthcare, security and surveillance, making decisions, tele-immersion, forecasting weather, official tasks, and prediction of risk analysis in society. In this paper, we perform three classification algorithms, Sequential Minimal Optimization (SMO), Random Forest (RF), and Simple Logistic (SL) with the two HAR datasets, UCI HAR and WISDM, downloaded from the UCI repository. The experiment described in this paper uses the WEKA tool to evaluate performance with the matrices, Kappa statistics, relative absolute error, mean absolute error, ROC Area, and PRC Area by 10-fold cross validation technique. We also provide a comparative analysis of the classification algorithms with the two determined datasets by calculating the accuracy with precision, recall, and F-measure metrics. In the experimental results, all the three algorithms with the UCI HAR datasets achieve nearly the same accuracy of 98%.The RF algorithm with the WISDM dataset has the accuracy of 90.69%,better than the others.

Keywords: Machine Learning, Human Activity Recognition, WEKA, Classifier, Classification Algorithms.

1. Introduction

Due to recent scientific research efforts, Machine Learning (ML) [1] as an essential branch of computer science has emerged out of Artificial Intelligence (AI). Based on the importance of the logical and knowledge-based approaches, there is a rift between ML and AI. ML mechanism relying on database technology is the extensive use of data

technological items [12]. Various problems such as HAR, air quality prediction[21], key sentence extraction using the comments in the blogs[22], detection of emergency situation[23], face recognition for security purpose [24],[30], film review analysis for maximizing the profit of investors and recommendation for viewers [25], intrusion detection[26], [31] can be solved using the ML techniques[27]. Human Activity Recognition (HAR) plays an essential role in different fields, such as human-computer interaction, health care, and security surveillance [2], as one of the prominent research branches. Due to specific challenges like optimal sensor placement, sensor motion, inherent variability, and cluttered background, HAR remains a very intricate task [3], [4]. HAR systems can replace human operators to intensify the proficiency and fruitfulness of the analysis and observation processes. For example, one of the HAR systems inform users about an emergent situation by tracking their health conditions with the help of specific sensor devices. Disaster management is a research area which attracts researchers of different communities like health care, computer science, business, and disaster management etc. The disaster recovery systems need to be designed using effective fault-tolerant techniques [41]. HAR plays a very important role during any kind of emergency. For collecting information on human behaviour, the steps with raw sensor data are concluded [10], [13], [15]: (a) pre-processing, (b) preliminary classification, (c) select classifier, and (d) performance evaluation. Classification is a widely used way of ML techniques, while the datasets consisting of training data and testing data are required. The training dataset always comprises a set of characteristics, and the primary role of classification divides the dataset to determine the classes [14]. There are several classification algorithms in the literature for resolving multiple challenges such as Multilayer Perceptron (MLP), SMO, decision tree, J48, RF, SL, Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Support Vector Machine (SVM) [28-29], [32], [42] and others. The essential components of classification techniques [15] are shown in Figure 1.

The remaining of the article is organized as follows: Section 2 summarizes the related works proposed in the literature. The methodology adopted in our experiments is presented in Section 3. Section 4 summarizes and analyzes the experimental results. Section 5 presents a detailed comparison of the adopted classification algorithms with the chosen HAR datasets and finally, we provide concluding remarks and future directions of this research in Section 6.

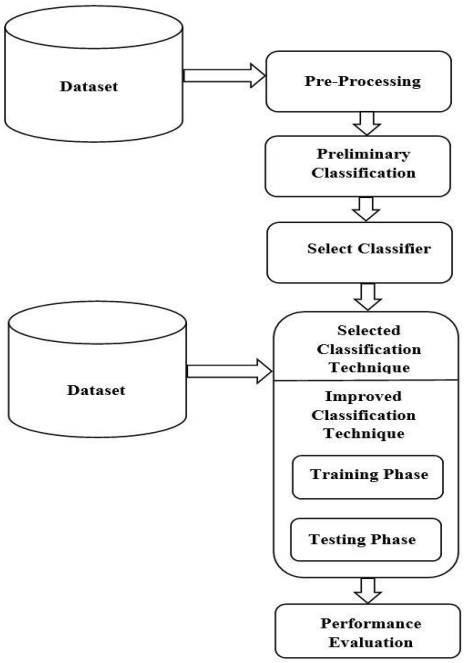


Fig. 1. Components of Classification Techniques

2. Related Work

In this section, a brief study of HAR datasets used for various applications is presented. Various researchers used different datasets for HAR in the literature [40]. The details of the datasets used are presented in Table 1.

Authors in [41] used different classifiers such as RF, IBK, J48, Bagging, and MLP for HAR. From the experimental results of the authors, it is indicated that RF performs well as compared to other considered classifiers and they achieve the 87.19 % accuracy. In [33], different classifiers like PEF, FNN, PTN, and PDF were used by the authors on various HAR datasets i.e. WISDM, MHEALTH, and SPAR. From the experimental results obtained, FNN was found to be the best classifier by the authors. The authors used categorical cross-entropy loss, the embedding and triplet loss for training. Also the authors observed that the training can be improved by using subject triplet selection. In [37], Yang *et al.* used DPCRCN for classification which uses end-to-end learning. During experimentation, they used Adam optimizer with the ReLU activation function. In [38], authors used HMDB, UCF101 and Kinetic datasets in their work and used a supervised approach where the weights of the branch were learned with standard back-propagation. The relation schema was integrated with an appearance branch and a Smart

Block was created to capture the spatiotemporal information. Then multiple Smart Blocks were stacked up to construct ARTNet.

Table 1. Details of HAR Datasets used for Various Applications

Author [Year]	Dataset Used	Tools/ Framework Used	Classifiers Used	Best Classifier	Accuracy in %
Nanda *et al.* [2021] [41]	WISDM Smartphone and Smartwatch Activity and Biometric Dataset	WEKA	RF, IBK, Bagging, J48 and MLP	RF	87.19
Burns *et al.* [2020] [33]	WISDM, MHEALTH, and SPAR	Seglearn, Keras, and Python Scikit-learn library	FCN, PEF, PDF, PTN	PTN	WISDM: 91. 3 MHEALTH: 99.9 SPAR: 99.0
Yang *et al.* [2019] [37]	AReM	LSTM, Fully connected Layer, and Softmax	LR, RF, SVM, DPCN, LSTM, XgBoost, LISEN, IDNNs, Dual Path Convolutional Neural Network (DPCRCN)	DPCRCN	99.97
Wang *et al.* [2018] [38]	Kinetics, HMDB51, UCF101	SMART Blocks, Two stream CNN, 3D CNNs, and ARTNets	C3D and ARTNet	ARTNet	Kinetics: 78. 7 HMDB51: 70.9 UCF101: 94.3
Min-Cheol Kwon *et al.* [2018] [39]	HAR dataset	Python Scikit-learn library	DT, RF, and SVM, ANN	ANN	95
Jain *et al.* [2017] [34]	Physical Activity Sensor data, UCI HAR	Score level fusion and Feature level fusion	k-NN and Multiclass SVM	Multiclass SVM	Physical Activity Sensor data: 96.83, UCI HAR: 97.12
Walse *et al.* [2016] [35]	WISDM	WEKA and Adaboost.M1	Decision Stump, Random Tree, RF, Hoeffding Tree, REP Tree, J48	J48	97.83
Kutlay *et al.* [2015] [36]	MHEALTH	WEKA	SVM and MLP	MLP	91.7

In [39], authors developed a HAR system where data from an off-the-shelf smartwatch was collected and ANN was used for human activity classification. The proposed system was improved by the authors by considering location information. From the experimental results of authors it was observed that an accuracy of 95% was achieved using ANN. Jain *et al.* [34] used UCI-HAR dataset and Physical Activity Sensor data for activity recognition and are publicly available sensor-based datasets. Here SVM and kNN classifiers were used by the authors. The simulation results indicate that SVM performs best for both of the datasets. Authors used a histogram of centroid and for feature extraction, gradient signature-based Fourier descriptor was utilized and then for information fusion, score and feature level fusion were combined. In [35], Walse *et al.* used different classifiers like Random Tree, J48, Hoeffding tree, RF, Decision Stump, and REP tree a log with MetaAdaboost.M1 for classification. All considered classification models were experimented with 10-fold cross-validation technique and J48 with MetaAdaboost.M1 was found to give improved results as compared to other algorithms. In [36], Kutlay *et al.* applied SVM and MLP classifiers on the MHEALTH dataset for classification. From the experimental results by the authors, it was found that MLP with 10 fold cross-validations gave 91.70% accuracy.

3. Methodology

In this section, we choose and discuss two HAR datasets downloaded from the UCI ML repository [16] and three classification algorithms. In this paper, we use the WEKA software as an open-source tool to demonstrate the classification algorithms.

Table 2. Description of UCI HAR and WISDM Datasets

Description/Dataset	**UCI-HAR**	**WISDM**
Instances	10299	15630426
Attributes	561	6
No. of Subjects	30	51
Activities	6	18
Characteristics	Multivariate/ Time-Series	Multivariate/ Time-Series
Associated Tasks	Classification/ Clustering	Classification
Sampling Rate	50 Hz	20 Hz
Device used	Smartphone: Samsung Galaxy 2	Smartphone: Google Nexus 5/5x or Samsung Galaxy S5 Smart watch: LG G Watch
Type	Filtered (Butterworth low pass filter)	Raw data as collected
Sensors used	Accelerometer, Gyroscope	Accelerometer, Gyroscope
File Type	Text	CSV

3.1. Datasets

We detail each of the datasets in Table 2. The UCI-HAR dataset is one of the chosen HAR datasets constructed from the recordings of 30 subjects performing activities of daily living while carrying a waist-mounted smartphone with embedded inertial sensors [5]. The dataset is characterized by multivariate and time series. The other dataset, WISDM, contains time-series sensor data of accelerometer and gyroscope, collected from 51 test subjects with 18 activities using smartphones and smart watches [6]. The characteristic of this dataset is multivariate on actual time-series and attribute characteristics.

3.2. Classification algorithms

The three classification algorithms are as follows: SMO, RF, and SL. We used WEKA tool to evaluate the algorithms with the two HAR datasets, provide a comparison of these algorithms, and suggest which algorithms may perform best. The specifics of the classification algorithms are described in this section below.

- **Sequential Minimal Optimization Algorithm (SMO):** This algorithm is developed by John C. Platt, Microsoft researcher in the year 1998 [18]. During the training of SVM, SMO is proposed to solve various quadratic problems. The worst-case time complexity of SMO as one of supervised classification algorithms is $O(n^3)$. Generally, SMO breaks down a significant problem into the sub-problems using the divide and conquer method, and solves them through analysis. SMO performs the functions of polynomial or RBF kernels to solve the classification problems, implemented in the popular LIBSVM tool [9] and widely used for training SVM. Using Support Vector Machine with sparse datasets, SMO is found to be the fastest algorithm.
- **Random Forest Algorithm (RF):**RF is developed by Breiman [19].This algorithm is designed for regression, classification, and other tasks by building a multitude of decision trees during training time and output the class, while the class represents the mode of classification or mean prediction of the trees [7],[8]. RF is one of the supervised learning algorithms, and mostly used for classification. By originating decision trees on the training data, the algorithm can make the prediction from each of the data samples. Generally, RF selects the best solution using the voting technique, and reduces the over fitting by averaging the results of calculated classification. Consequently, RF consisting of multiple single trees is an ensemble method, better than a single decision tree.
- **Simple Logistic Algorithm (SL):** This algorithm proposed by Sumner *et al.* in the year 2005 [20] is used for modelling the possibility of a particular event or certain class such as fail/pass, or lose/win. The algorithm for supervised learning tasks applies the simple logistic function on the data to predict the probability of a target variable. Besides, the algorithm is used to model a binary dependent variable, while every event would be assigned a probability value between 0 and 1. In logistic regression, estimating the parameters of a logistic model is applied in various fields such as ML, medical fields, and social sciences. Logistic regression is a statistical model used to model a binary dependent variable using a logistic function, although

many variations exist. In regression analysis, logistic regression estimates a logistic model with given parameters, as a form of binary regression [11].

The different features of these three classification algorithms are given in Table 3.

Table 3. Features of SMO, RF, and SL Classification Algorithms

Algorithm	SMO	RF	SL
Primary Problem	Classification	Classification and Regression	Classification can be done but good for Regression
Class Type	Binary and Multiclass	Binary and Multiclass	Good for Binary but Multiclass is also possible
Solution Approach	Quadratic Programming	Uncorrelated forest of trees	Statistical Learning
Dataset Type	Large	Large	Small
Time Complexity	$O(n^3)$	O(v * n (log (n))	O(n)
Data Normalization	Required	Not Required	Not Required
Raw Implementation	Difficult	Difficult	Easy
Predictors	Categorical or Numeric	Categorical or Numeric	Numeric

4. Results and Analysis

We have presented the results obtained from our experimentation, along with the accuracy analysis of the classifiers in this section. We considered two HAR datasets, UCI HAR and WISDM, to evaluate the quality of each of the classifiers. Three particular classifiers corresponding to SMO, RF, and SL algorithms were generated with the two datasets. We measured a number of parameters such as the accuracy percentage; the number of correctly classified instances, and the error percentage, while as the three metrics of the accuracy are adopted, corresponding to the values of precision, recall, and F-measure. Using the confusion matrix and weighted average computation, we calculated all these measures for each of the classifiers. The sum of diagonals in the matrix denotes the number of correctly classified instances. True Positive (TP) and False Positive (FP) denote the true positive and false-positive rates. Suppose that TP_A, TP_B, and TP_C denote the TP number of class A, class B, and class C, individually, the accuracy value is computed in Equation (1) as follows:

$$\text{Accuracy} = \frac{TP_A + TP_B + TP_C}{Total\ number\ of\ classification} \quad (1)$$

The other metrics related to accuracy are used for evaluating the performance results of each of the classifiers, as follows:

Precision: This metric is called positive predictive value as the fraction of appropriate instances among the retrieved instances. The formula of Precision is defined in Equation (2), as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall: Recall, also called sensitivity, is the fraction of the number of appropriate instances retrieved [17], defined in Equation (3):

$$\text{Recall}=\frac{TP}{TP+FN} \tag{3}$$

F-Measure: Precision and accuracy are combined into the calculation of F-Measure. Keeping in view the weighted average of both values, F-measure in Equation (4) is calculated as follows:

$$F=\frac{2*precision*recall}{precision+recall} \tag{4}$$

MCC: The Matthews Correlation Coefficient (MCC) correlates with the actual and predictive series. The formula of MCC always returns a value between -1 and +1 [17], defined as Equation (5).

$$MCC=\sqrt{\frac{X^2}{n}}, \text{ where } n \text{ is the total number of observations.} \tag{5}$$

Kappa statistic: The kappa statistic, κ measures the inter-related reliability for the qualitative items. Since κ considers the agreement possibility that occur by chance [17], the formula of κ is defined as follows.

$$\kappa \equiv \frac{P_0-P_e}{1-P_e}=1-\frac{1-P_0}{1-P_e} \tag{6}$$

Where, P_0 is the relative observed agreement among raters and is identical to accuracy. P_e represents the hypothetical probability of chance agreement. The observed data is used to calculate the probabilities of each observer randomly by considering each category [17].

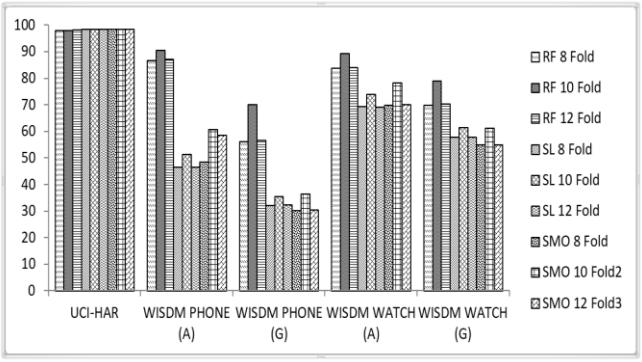


Fig. 2. Accuracy % Graph for 8, 10, and 12 Fold Cross Validation

In this work, the experimental results of the classifiers were operated on the considered datasets, while the classifiers were implemented by the SMO, RF, and Simple Logistics algorithms. We have used 8-fold, 10-fold, and 12-fold cross-validation to estimate the performance of all of the three considered algorithms during our experimentation. Based on the results, it was found that the cases of 10-fold cross-validation have a good accuracy performance on all the three algorithms. The experimental results of all the considered algorithms with different cross validations are presented in Figure 2.

The results obtained for each of the algorithms in terms of the different parameters such as TP rate, FP rate, Precision, Recall, F-Measure, MCC, ROC area, and PRC area are given in Tables 4-12 based on the datasets. The WISDM dataset contains the sensor data, collected using mobile devices such as phone and watch. Accelerometer and Gyroscope as sensors are embedded to the considered devices. The classification algorithms identify their activities with the data received from the phone and watch devices.

Table 4. Weighted Average of Accuracy of the SMO Algorithm with 8 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.99**	0.00	0.99	0.99	0.99	0.98	0.99	0.98
WISDM	Phone A	0.49	0.03	0.48	0.49	0.48	0.45	0.89	0.39
	Phone G	0.30	0.04	0.29	0.30	0.29	0.25	0.81	0.23
	Watch A	**0.70**	0.02	0.70	0.70	0.70	0.68	0.95	0.61
	Watch G	0.55	0.03	0.55	0.55	0.55	0.52	0.90	0.43

Table 5. Weighted Average of Accuracy of the RF Algorithm with 8 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.98**	0.00	0.98	0.98	0.98	0.78	1.00	1.00
ISDM	Phone A	**0.87**	0.01	0.87	0.87	0.87	0.86	0.99	0.93
	Phone G	0.56	0.03	0.56	0.56	0.56	0.53	0.92	0.60
	Watch A	0.84	0.01	0.84	0.84	0.84	0.83	0.99	0.90
	Watch G	0.70	0.02	0.70	0.70	0.69	0.68	0.96	0.75

Table 6. Weighted Average of Accuracy of the SL Algorithm with 8 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.99**	0.00	0.99	0.99	0.99	0.98	0.99	0.99
WISDM	Phone A	0.47	0.03	0.45	0.47	0.45	0.43	0.89	0.45
	Phone G	0.32	0.04	0.30	0.32	0.30	0.27	0.84	0.30
	Watch A	**0.69**	0.02	0.69	0.69	0.69	0.67	0.97	0.75
	Watch G	0.58	0.03	0.57	0.58	0.57	0.55	0.93	0.59

Tables 4-6 present the classification results in terms of accuracy measures such as TP, FP, Recall, F-Measure, Precision, MCC, PRC area, and ROC area for the considered datasets with 8 fold cross validation. The dataset is partitioned into 8 different sets randomly. Among them one set behaves as validation set whereas remaining sets act as training set. This was repeated for 8 times by considering each partition as validation set and the results are averaged to get the prediction. In UCI-HAR dataset, SMO, RF, and SL algorithms produce nearly the same results. Also, in UCI HAR dataset, SMO, RF as well as SL algorithm results in higher accuracy percentage whereas in WISDOM dataset the accuracy percentage was found to be less in all types of sensors data and Phone with

Accelerometer sensor performs better among them in RF algorithm with an accuracy of 87%.

Table 7. Weighted Average of Accuracy of the SMO Algorithm with 10 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.98**	0.00	0.98	0.98	0.98	0.98	0.99	0.97
WISDM	Phone A	0.60	0.02	0.60	0.60	0.60	0.58	0.93	0.53
	Phone G	0.36	0.03	0.35	0.36	0.35	0.31	0.84	0.29
	Watch A	**0.78**	0.01	0.78	0.78	0.78	0.77	0.97	0.72
	Watch G	0.61	0.02	0.60	0.61	0.60	0.58	0.93	0.51

Table 8. Weighted Average of Accuracy of the RF Algorithm with 10 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.98**	0.00	0.98	0.98	0.98	0.97	0.99	0.99
WISDM	Phone A	**0.90**	0.00	0.90	0.90	0.90	0.90	0.99	0.95
	Phone G	0.70	0.01	0.69	0.70	0.69	0.68	0.95	0.76
	Watch A	0.89	0.00	0.89	0.89	0.89	0.88	0.99	0.94
	Watch G	0.79	0.01	0.78	0.79	0.78	0.77	0.97	0.84

Table 9. Weighted Average of Accuracy of the SL Algorithm with 10 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.98**	0.00	0.98	0.98	0.98	0.98	0.99	0.99
ISDM	Phone A	0.51	0.02	0.50	0.51	0.50	0.48	0.91	0.52
	Phone G	0.35	0.03	0.34	0.35	0.34	0.30	0.85	0.34
	Watch A	**0.74**	0.01	0.73	0.74	0.73	0.72	0.97	0.78
	Watch G	0.61	0.02	0.60	0.61	0.60	0.58	0.94	0.63

Table 10. Weighted Average of Accuracy of the SMO Algorithm with 12 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.99**	0.00	0.99	0.99	0.99	0.98	0.99	0.98
WISDM	Phone A	0.49	0.03	0.48	0.49	0.48	0.45	0.89	0.39
	Phone G	0.31	0.04	0.29	0.30	0.29	0.26	0.81	0.23
	Watch A	**0.70**	0.02	0.70	0.70	0.70	0.68	0.95	0.61
	Watch G	0.55	0.03	0.55	0.55	0.55	0.52	0.90	0.44

Like 8 fold cross validation, the result of 10 fold cross validation was also estimated and are provided in the Tables 7-9 using all the considered algorithms and datasets. From the simulation

results, it is found that the prediction accuracy is improved in 10 fold cross validation as compared to the 8 fold cross validation. Using the UCI-HAR dataset, SMO, RF, and SL algorithms produce the same results and the accuracy percentage is about 98%. Using the WISDM dataset, we computed the accuracy of the classifiers and is found to give better accuracy as compared to 8 fold cross validation in all types of sensors data.

Table 11. Weighted Average of Accuracy of the RF Algorithm with 12 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.98**	0.00	0.98	0.98	0.98	0.78	0.99	0.99
WISDM	Phone A	**0.87**	0.01	0.88	0.87	0.87	0.87	0.99	0.94
	Phone G	0.57	0.03	0.56	0.57	0.56	0.54	0.92	0.60
	Watch A	0.84	0.01	0.84	0.84	0.84	0.83	0.99	0.90
	Watch G	0.70	0.02	0.70	0.70	0.70	0.68	0.96	0.75

Table 12. Weighted Average of Accuracy of the SL Algorithm with 12 Fold Cross Validation

Datasets	Device	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
UCI HAR	Phone	**0.99**	0.00	0.99	0.99	0.99	0.98	0.99	0.99
WISDM	Phone A	0.47	0.03	0.45	0.47	0.46	0.42	0.89	0.46
	Phone G	0.33	0.04	0.31	0.33	0.31	0.28	0.84	0.31
	Watch A	**0.69**	0.02	0.69	0.69	0.69	0.67	0.97	0.75
	Watch G	0.58	0.03	0.57	0.58	0.57	0.55	0.93	0.59

Table 13. Comparative Analysis Results of SMO, RF, and SL on UCI-HAR and WISDM Datasets

Datasets	Device	Classifiers	Accuracy in %	Kappa statistics	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error in %	Root Relative Squared Error in %
UCI-HAR	Phone	SMO	**98.52**	0.98	0.22	0.31	80.32	83.47
		RF	98	0.97	0.04	0.10	17.03	28.47
		SL	98.47	0.98	0.00	0.06	2.94	16.62
WISDM	Phone A	SMO	60.94	0.58	0.09	0.21	94.99	95.91
		RF	**90.69**	0.90	0.02	0.09	26.11	42.55
		SL	51.48	0.48	0.07	0.18	67.56	80.69
	Phone G	SMO	36.65	0.32	0.10	0.22	95.88	96.88
		RF	**70.26**	0.68	0.68	0.16	59.87	70.91
		SL	35.65	0.31	0.08	0.20	80.80	89.19
	Watch A	SMO	78.47	0.77	0.09	0.21	94.48	95.38
		RF	**89.51**	0.88	0.03	0.10	31.43	46.62
		SL	74.10	0.72	0.04	0.14	41.54	62.55
	Watch G	SMO	61.27	0.58	0.09	0.21	94.95	95.90
		RF	**79.01**	0.77	0.05	0.14	49.35	62.24
		SL	61.44	0.59	0.06	0.17	58.46	74.57

Table 10-12 present the results with 12 fold cross validation. The obtained results indicate that the accuracy decreases as compared to those with 10 fold cross validation. So, from this experimental study we can conclude that for both the datasets 10-fold cross validation works well as compared 8 fold and 12 fold cross validation.

The accuracy percentages of all of the algorithms for both the datasets are presented in Table 13. From the comparative analysis results, it showed that the RF algorithm in all of the cases performs well in predicting human activity as compared to the SMO and SL algorithms in WISDM dataset and SMO showed higher accuracy of 98.52% in UCI-HAR dataset as compared to RF and SL.

5. Comparative Analysis

A comparative analysis results in terms of accuracy is shown in Figure 3 for the SMO, RF, and SL algorithms with both UCI and WISDM datasets. SL algorithm is a good binary classifier which performs better than the others while adopting small datasets. In the current work, we have dealt with the multiclass data to the algorithms, and the experimental results showed that the RF and SMO performed better than the SL. In this paper, WISDM denotes a large dataset and UCI HAR denotes a small dataset. All of the three considered algorithms give good results in UCI HAR. On the contrary, the results obtained with the WISDM dataset were found to have less accuracy in the algorithms. Another observation from our experiment is that compared with using the gyroscope sensor, no matter what kind of device's accelerator sensor is used, it can provide better results.

From the obtained results, it is also found that the weighted average of TP rate is high in SMO and SL classifiers with the UCI HAR dataset. The weighted average of TP rate is high in RF with WISDM by the phone accelerometer. Consequently, SMO and SL with the UCI dataset provide better performance in terms of precision, recall, and F-measure values. However, RF with WISDM gives the best performance in the same conditions.

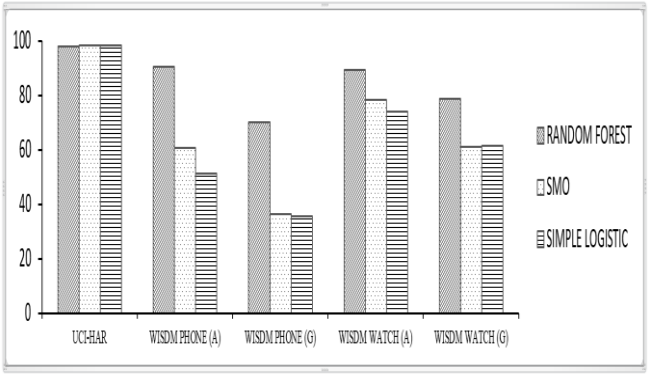


Fig. 3. A comparative analysis in terms of accuracy for SMO, RF, and Simple Logistic algorithms with UCI and WISDM datasets

Using the UCI HAR dataset, the kappa values of the three classifiers are almost same. Nevertheless, the kappa values for the classifiers with the WISDM dataset are different widely. By performing the RF with the WISDM dataset in Phone A, the kappa value is 0.90, higher than the others. For UCI-HAR, although SMO results in a slightly higher accuracy as compared to RF and SL algorithms but the different error percentages such as Mean Absolute Error, Relative Absolute Error, Root Mean Squared Error, and Root Relative Squared Error was found to be less for SL than RF and SMO and is presented in Table 13. However, for WISDM dataset, in all types of sensors RF was found to give better results along with less error percentages as compared to SMO and SL.

A MCC value of 0.98 was obtained in SMO and SL classifiers with the UCI dataset. While the three classifiers were applied in different devices, RF with the WISDM dataset obtained the largest value of MCC in Phone A, but a low value of MCC in the other devices. In Table 5, the three algorithms with the UCI dataset returned the ROC area value of 0.99as the maximum value from the range between 0 and 1 when being compared with these with the WISDM dataset.

In addition, Watch A returns the ROC value of 0.97 as the maximum value in all the cases of WISDM dataset in Table 5. RF and SL with the UCI dataset have the maximum value of the PRC area, 0.99.In addition, RF with the WISDM dataset in Phone A, returned the PRC value of 0.95 in Table 4. According to the experimental results, it is evident that when the number of activities is increasing, the accuracy is decreasing, and when the number of activities is decreasing, the values of accuracy and the other metrics are increasing.

Based on the above reason, the accuracy value is inversely proportional to the number of activities. That is why the classifiers with the UCI dataset consisting of six activities achieve better results than them with the WISDM dataset consisting of eighteen activities.

6. Conclusion

In this work, we have evaluated the performance of the three classification algorithms, namely SMO, RF, and SL in terms of the metrics related to accuracy such as TP rate, FP rate, F-measure, Precision, Recall, ROC area, and PRC area. On the UCI HAR dataset, SMO is a better algorithm than the others; however all of the algorithms provide nearly equal results. On the WISDM dataset, we found that RF is the best algorithm. According to the experimental results, we infer that the adopted classification algorithms are suitable to classify human activities in the domain of HAR. Recognizing human activities is very important and is useful for various applications like elderly care service, healthcare, assisted living, smart home, etc. This study can provide a reference to help researchers make decisions on applying classification algorithms into human activity recognition.

Acknowledgment. This work is funded by CURIE Grant, Department of Science and Technology, Govt. of India under Grant no. DST/CURIE/01/2019(G) dt. 20.05.2019.

References

1. Sarle, W. S. Neural networks and statistical models, *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute, USA, pp. 1538-1550. (1994).
2. Ann, O. C., and Theng, L. B. Human activity recognition: a review. In *2014 IEEE international conference on the control system, computing, and engineering (ICCSCE 2014)* (pp. 389-393). IEEE. (2014).
3. Oikonomopoulos, A., and Pantic, M. Human activity recognition using hierarchically-mined feature constellations. In *International Symposium on Visual Computing*, Springer, Berlin, Heidelberg, pp. 150-159. (2013).
4. Roshtkhari, M. J., and Levine, M. D. Human activity recognition in videos using a single example. *Image and Vision Computing*, *31*(11), pp. 864-876. (2013).
5. Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. A public domain dataset for human activity recognition using smartphones. *In ESANN proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges (Belgium), 24-26 April 2013, ISBN 978-2-87419-081-0. (2013).
6. Weiss, G. M., Yoneda, K., and Hayajneh, T. Smartphone and Smart watch-Based Biometrics Using Activities of Daily Living. *IEEE Access*, *7*, pp. 133190-133202. (2019).
7. Ho, T. K. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1, pp. 278-282. IEEE. (1995).
8. Ho, T. K. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 832-844. (1998).
9. Chang, C. C., and Lin, C. J. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, *2*(3), pp. 1-27. (2011).
10. Zanni, L., Serafini, T., and Zanghirati, G. Parallel software for training large scale support vector machines on multiprocessor systems. *Journal of Machine Learning Research*, *7*(Jul), pp. 1467-1492. (2006).
11. Tolles, J.,andMeurer, W. J. Logistic regression: relating patient characteristics to outcomes. *Jama*, *316*(5), pp. 533-534. (2016).
12. Singaravelan, S., Arun, R., Arun Shunmugam, D., Ruba Soundar, K., Mayakrishnan, R., and Murugan, D. Analysis of classification algorithms on different datasets. *Review of Innovation and Competitiveness: A Journal of Economic and Social Research*, *4*(2), pp. 41-54. (2018).
13. Ranasinghe, S., Al Machot, F., and Mayr, H. C. A review of applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks*, *12*(8), 1550147716665520. (2016).
14. Krishnan, N. C., Juillard, C., Colbry, D., and Panchanathan, S. Recognition of hand movements using wearable accelerometers. *Journal of Ambient Intelligence and Smart Environments*, *1*(2), pp. 143-155. (2009).

15. Florence, A. M., and Savithri, R. Talent knowledge acquisition using the C4.5 classification algorithm. *International Journal of Emerging Technologies in Computational and Applied Sciences*, pp. 406-410. (2013).
16. Dua, D., and Graff, C. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, Accessed: 2020-05-01 (2019).
17. Alghobiri, M. A Comparative Analysis of Classification Algorithms on Diverse Datasets, Engineering, Technology & Applied Science Research, 8(2), pp. 2790-2795. (2018).
18. Platt, J. Sequential minimal optimization: A fast algorithm for training support vector machines, Technical Report MSR-TR-98-14(1998).
19. Breiman, L. Random forests. Machine learning, 45(1), 5-32. (2001).
20. Sumner, M., Frank, E., & Hall, M. Speeding up logistic model tree induction. In European conference on principles of data mining and knowledge discovery (pp. 675-683). Springer, Berlin, Heidelberg. (2005).
21. Chang, F., Ge, L., Li, S., Wu, K., & Wang, Y. Self-adaptive spatial-temporal network based on heterogeneous data for air quality prediction. Connection Science, 1-20. (2020).
22. 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. (2020).
23. Nanda, S., Panigrahi, C. R., Pati, B., & Mishra, A. A Novel Approach to Detect Emergency Using Machine Learning. In Progress in Advanced Computing and Intelligent Engineering (pp. 185-192). Springer, Singapore.
24. Verma, R. K., Singh, P., Panigrahi, C. R., & Pati, B. ISS: Intelligent Security System Using Facial Recognition. In Progress in Advanced Computing and Intelligent Engineering (pp. 96-101). Springer, Singapore. (2019).
25. Wang, Q., Zhu, G., Zhang, S., Li, K. C., Chen, X., & Xu, H. Extending emotional lexicon for improving the classification accuracy of Chinese film reviews. Connection Science, 1-20. (2020).
26. Tian, Q., Han, D., Li, K. C., Liu, X., Duan, L., & Castiglione, A. An intrusion detection approach based on improved deep belief network. Applied Intelligence. (2020).
27. Ramasamy, V., Gomathy, B., Obulesu, O., Sarkar, J. L., Panigrahi, C. R., Pati, B., & Majumder, A. Machine Learning Techniques and Tools: Merits and Demerits. New Age Analytics, 23. (2020).
28. Qin, P., Chen, J., Zhang, K., & Chai, R. Convolutional neural networks and hash learning for feature extraction and of fast retrieval of pulmonary nodules. Computer Science and Information Systems, 15(3), 517-531. (2018).
29. Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., & Atkinson, P. M. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS Journal of Photogrammetry and Remote Sensing, 140, 133-144. (2018).
30. Cao, L., &Shen, H. Imbalanced data classification based on hybrid resampling and twin support vector machine. Computer Science and Information Systems, 14(3), 579-595. (2017).

31. Albawi, S., Mohammed, T. A., & Al-Zawi, S. Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). IEEE. (2017).
32 Bhargava, N., Sharma, G., Bhargava, R., & Mathuria, M. Decision tree analysis on j48 algorithm for data mining. Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering, 3(6). (2013).
33 Burns, D. M., and Cari M. W. Personalized Activity Recognition with Deep Triplet Embeddings. arXiv preprint arXiv: 2001.05517 (2020).
34 Ain, A., and Vivek K. Human activity classification in smartphones using accelerometer and gyroscope sensors. IEEE Sensors Journal 18, no. 3 (2017): 1169-1177.
35 Walse, K. H., Rajiv, V. D., and Vilas, M. T. A study of human activity recognition using Ada Boost classifiers on WISDM dataset. The Institute of Integrative Omics and Applied Biotechnology Journal 7, no. 2: 68-76. (2016)
36 Kutlay, M. A., and Sadina, G. Application of machine learning in healthcare: Analysis on mhealth dataset. Southeast Europe Journal of Soft Computing 4, no. 2. (2016)
37 Yang, C., Wenxiang J., and Zhongwen G. Time Series Data Classification Based on Dual Path CNN-RNN Cascade Network. IEEE Access 7: 155304-155312. (2019)
38 Wang, L., Wei, L., Wen, L., and Luc Van Gool. Appearance-and-relation networks for video classification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1430-1439. (2018)
39 Kwon, M. and Choi, S. Recognition of Daily Human Activity Using an Artificial Neural Network and Smartwatch. Wireless Communications and Mobile Computing, Hindawi, https://doi.org/10.1155/2018/2618045. (2018)
40 Biswal, A., Nanda, S., Panigrahi C. R., Cowlessur, S. K., Pati, B. Human Activity Recognition Using Machine Learning: A Review. Progress in Advanced Computing and Intelligent Engineering, Advances in Intelligent Systems and Computing 1299, pp. 323-333. (2021)
41 Mendonça, J., Andrade, E., Endo, P.T., Lima, R. Disaster recovery solutions for IT systems: A Systematic mapping study, Journal of Systems and Software, Volume 149, pp. 511-530. (2019)
42 Feng, F., Li, K-C, Shen, J., Zhou, Q., Yang, X. Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification. IEEE Access, vol. 8, pp. 69979-69996. (2020)

Suvra Nayak received her M.Sc and M. Phil degree in Computer Science from Rama Devi Women's University, Bhubaneswar, India in 2021. Her research interests include Machine Learning and Computer Networks.

Chhabi Rani Panigrahi received her Ph.D. in Computer Science and Engineering from IIT Kharagpur, India. She is currently an Assistant Professor in the Department of Computer Science at Rama Devi Women's University, Bhubaneswar, India. Prior to this,

she was working as Assistant Professor in Central University of Rajasthan, India. Her research interests include Software Testing, Mobile Cloud Computing, and Machine Learning. She holds 20 years of teaching and research experience. She has published several international journals, conference papers, and books. She served as chairs and technical program committee member in several conferences of international repute.

Bibudhendu Pati completed his Ph.D. degree from IIT Kharagpur, India. He is currently working as Associate Professor in the Department of Computer Science at Rama Devi Women's University, Bhubaneswar, India. He has around 23 years of experience in teaching and research. His areas of research interests include Wireless Sensor Networks, Cloud Computing, Big Data, Internet of Things, and Advanced Network Technologies. He has got several papers published in reputed journals, conference proceedings, and books of international repute. He has been involved in many professional and editorial activities. He is a Life Member of Indian Society for Technical Education, Computer Society of India and Senior Member of IEEE.

Sarmistha Nanda is a Ph.D. research scholar in the Department of Computer Science at Rama Devi Women's University, Odisha, India. She received her M.Tech. degree in Computer Science from Sambalpur University, Odisha, India. She worked as a JRF in the National Institute of Science Education and Research, Bhubaneswar, India, and Research Associate at Central Rice Research Institute, Cuttack. She also served as a Senior Software Developer in a software firm. Her area of interest is Machine Learning, Cloud Computing, and Programming.

Meng-Yen Hsieh received his MS degree in Computer Science & Information Engineering from National Central University, Taiwan, R.O.C. in 2001, and PhD degree in Engineering Science from National Cheng Kung University, Taiwan, R.O.C. in 2007. He is currently a professor of the Department of Computer Science & Information Engineering, Providence University, Taiwan, R.O.C. His research interests include computer security, machine learning, block chain application, and software engineering. Dr. Hsieh has served on symposium chairs and technical program committees for several international conferences. **http://www1.pu.edu.tw/~mengyen**

Received: December 21, 2020; Accepted: June 17, 2021.

Computer Science and Information Systems 19(1):65–85 https://doi.org/10.2298/CSIS210215037G

Distributed Ledger Technology: State-of-the-Art and Current Challenges

Maria Gorbunova[1,2], Pavel Masek[3], Mikhail Komarov[2], and Aleksandr Ometov[4]

[1] Rostelecom, Moscow, Russia mariya.v.gorbunova@rt.ru
[2] National Research University Higher School of Economics, Russia mkomarov@hse.ru
[3] Brno University of Technology, Brno, Czech Republic masekpavel@vutbr.cz
[4] Tampere University, Tampere, Finland aleksandr.ometov@tuni.fi

Abstract. Distributed Ledger Technology (DLT) is making the first steps toward becoming a solution for the growing number of various decentralized systems worldwide. Unlike pure Blockchain, DLT finds many uses across different industries, including eHealth, finance, supply chain monitoring, and the Internet of Things (IoT). One of the vital DLT features is the ability to provide an immutable and commonly verifiable ledger for larger-scale and highly complex systems. Today's centralized systems can no longer guarantee the required level of availability and reliability due to the growing number of the involved nodes, complicated heterogeneous architectures, and task load, while the publicly available distributed systems are still in their infancy. This paper aims to provide an exhaustive topical review of the state-of-the-art of Distributed Ledger Technology applicability in various sectors. It outlines the importance of the practical integration of technology-related challenges, as well as potential solutions.

Keywords: Distributed management, Distributed information systems, Data storage systems, Information exchange, Information security

1. Introduction and Motivation

The number of new services and devices on the market is growing at a tremendous pace on a yearly basis [59]. More and more functions previously performed by humans are transferred to various smart devices to make life easier. Simultaneously, the devices evolve, intending to become more computationally powerful in addition to more efficient and independent data processing. On the other hand, the centralized coordination of an ever-growing number of devices is already becoming a significant problem for conventional systems designed for a smaller number of active nodes [47]. The Distributed Ledger Technology (DLT) has caused quite a stir over the last years as many experts now consider that it has the potential for facilitating multiple bursts of creativity and catalyzing an exceptional level of digital innovation not seen since the advent of the Internet [80]. DLT has been proposed to ensure the effective interaction of various complex-scalable systems [7].

The original concept of DLT existed before Bitcoin and blockchain concepts. The Byzantine Generals Problem theorized by Lamport et al. as early as in the 1980th described how § 'computer systems must handle [...] conflicting information' in an adversarial environment [40]. Subsequent research led to the emergence of the first algorithm for 'highly available systems that tolerate Byzantine faults' with little increase in latency [16].

The earliest identified occurrences of the concept of a 'Blockchain' can be traced back to the 1990^{th} with Haber et al. that introduced the notion of a chain of cryptographically-linked data blocks to efficiently and securely timestamp digital data in distributed systems using cryptographic hashing functions and Merkle trees [62], a mathematical construct known for more than a third of a century [51].

Today, DLT aims at enabling the operation of a highly available, append-only database, which is maintained by physically distributed storage and computing devices (referred to as nodes), in an untrustworthy environment [67]. DLT promises to increase efficiency and transparency of collaborations between individuals and/or organizations based on inherent qualities such as tamper- and censorship resistance, and democratization of data [36]. Using a variety of methods, DLT provides an opportunity to solve several challenges in various industries providing an additional level of secure abstraction for direct interaction between heterogeneous systems [39].

DLT has received growing attention as an innovative method of storing and updating data within and between organizations in recent years. A distributed ledger is a digital ledger that is different from centralized networks and ledger systems in two significant aspects. First, information is stored in a network of machines, with changes to the ledger reflected simultaneously for all ledger holders. Second, the data is authenticated by a cryptographic signature. Together, it provides a transparent and verifiable record of transactions. Blockchain technology is one of the most well-known underlays of DLT, in which the ledger comprises 'blocks' of transactions, and it is the technology that underlies the cryptocurrency Bitcoin [20].

However, the possible uses of DLT go far beyond the financial sector, e.g., its use has also been explored in education [18,61,70], e-Governance [33], agriculture [29], supply chain [57] and many other industries. This paper is a critical review aiming to provide a comparative analysis of existing DLT applications and to answer the following research question:

> What are the main challenges of the distributed ledger technology integration and its further operation?

The rest of the paper is organized as follows. The next section provides an overview of the method applied to the topical literature review. Section 3 describes the use of technology in various industries and the corresponding motivation to move toward DLT. The review covers sectors such as eHealth, education, supply chain, intellectual property, Internet of Things (IoT), finance, energetics, as well as a vision from the horizontal domain perspective. Next, Section 4 outlines the main challenges that can be solved by applying the DLT to the identified industries, thus, outlining the future perspectives of the DLT. The last section concludes the paper.

2. Methodology

This section outlines the research methodology adopted to carry out this systematic literature review based on the PRISMA guidelines proposed in [44].

In order to identify key publications on the analysis of distributed ledgers through modeling or simulation, we performed a literature search in scientific databases that cover leading computer science journals and conferences: *IEEE Xplore*, *ACM Digital Library*, *ScienceDirect*, *SAGE Journals Online*, and *Springer Link*.

To find relevant articles and papers for our research, we applied the following search string: *(DLT OR "Distributed Ledger") AND (Applications OR Challenges OR "Future Perspective" OR State-of-the-Art)*

In total, we gathered a set of 963 potentially relevant publications, excluding grey literature and pre-prints.

We then analyzed the titles, keywords, and abstracts of the publications to identify papers and articles that described at least one modeling or simulation approach for distributed ledgers. In doing so, we selected a total of 45 publications. To further extend our literature sample, we analyzed the selected publications' references for additional papers or articles relevant to our research. Following this process resulted in a total of 61 publications.

Once the literature selection process was completed, we carefully read the selected publications and used an open coding approach to identify the described DLT applications and challenges. Next, we classified the extracted applications into six general groups. The results of our analysis form the core of the topical literature review and are presented in the next section.

3. DLT Applications State-of-the-Art

Today, the potential for using DLT technology can be already seen in almost all areas of society, from healthcare to complex information systems, see Fig. 1. This section discusses the most promising DLT application areas, according to the literature review.

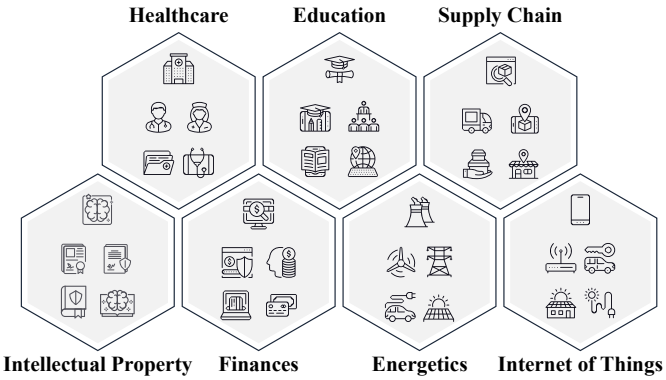


Fig. 1. Main DLT Applications Classification

3.1. Healthcare sector

The use of Information and Communication Technologies (ICT) for health is commonly defined as eHealth or "an emerging field of medical informatics, referring to the organization and delivery of health services and information using the Internet and related technologies. In a broader sense, the term characterizes not only a technical development but also a new way of working, an attitude, and a commitment for networked, global thinking, to improve health care locally, regionally, and worldwide by using information and communication technology" according to to [56]. eHealth has recently been added to the list of sectors affected by DLT in several ways. DLT's technological advances in eHealth have been documented, among other things, by using data to record and analyze human behavior. The adoption of wearables and IoT devices is expected only to accelerate this expansion [46,55]. It has been partially addressed by General Data Protection Regulation (European Union) 2016/679 (GDPR), which requires transparency in data use as well as covers other aspects related to health-related data [35,63].

One of the main challenges of the modern medical sector is the lack of unified patient data storage [5]. To date, proprietary cloud centralized storages are utilized to solve this problem being individual for different clinics, even city-wise. As a result, the complete medical history or a list of diagnoses per patient can hardly be accessed. Moreover, it is not possible to track the patient's indices and produce appropriate analysis.

In 2015, 78.8 million patients, nearly a quarter of the US population, had their information stolen after a hack occurred on the insurance corporation Anthem [23]. At the end of 2019, the health insurance company started rolling out blockchain-powered features that allow patients to securely access and share their medical data. In the next two to three years, all 40 million members will gain access to it.

In this context, DLT has emerged as a path to application development that enables interoperability between systems by providing secure and immutable information storage and exchange [13,53]. Examples of extant use cases are defined for the following areas of healthcare: pharma, biotechnology, medicine, insurance, genomics [2,68].

The authors of [77] propose a distributed system for patient health data sharing. The content is generated by two types of IoT nodes: wearable devices and static sensors. The data-sharing mechanism is operated through a distributed registry based on a system called Tangle and based on Directed Acyclic Graph (DAG). The Masked Authenticated Messaging (MAM) protocol is used to securely transmit encrypted data streams, thus, ensuring reliable authentication. Merkle tree is applied to ensure data integrity [12].

The next tremendous challenge of the healthcare sector is the need to ensure data confidentiality [27]. In most cases, the patient cannot have a complete view of the patient's medical data processing and access, which should be improved in the next-generation data exchange systems.

Medical data storage is essential in eHealth being extremely sensitive and, therefore, a primary target for various attacks [10]. Since the patients themselves could govern access to their health records, there would no longer be a central point of attack that could be compromised to release large numbers of patient records. Therefore, DLT has the potential to provide a flexible framework for the management of health data. DLTs provide the infrastructure which may enable users in the future to have more control over their health histories and medical records, allowing for better decision making and preventative measures to be applied [76].

Next, the work [25] provides a deep discussion on the main components of blockchain required for developing e-Health targeted distributed architectures. The studied roadmap outlines how DLT can fit into existing Electronic Health Records (EHR) systems or Insurance Content Management (ICM) systems. It also covers important topics of data exchange and privacy in such a heterogeneous environment.

The authors of [22] also presents a system allowing for flexible third-party access to electronic medical records of patients, which makes it possible to confirm the fact of such interaction unequivocally. The authors developed a DLT prototype, which records the validity of information processing using Smart contracts based on Solidity [72].

Pharmaceutical companies often receive government funding to produce specific drugs, such as vaccines and autoimmune diseases. DLT is useful and presumably suitable in areas such as transferring funds from the state treasury to the company and the transfer of medical supplies along the supply chain. In 2020, counterfeit medicines will cost the world economy more than 75 million USD. Tracking the provenance of drugs using DLT is one approach to reduce this trade in counterfeit medicines [43].

3.2. Education sector

From the education perspective, one of the DLT applications is related to professional competencies [54]. The authors propose a procedure for developing a register of the population of professional competencies. The proposed algorithm for the Education Index's critical components evaluation is based on the Ethereum blockchain platform. Moreover, a scoring model for calculating these parameters is developed, and a Solidity smart contract scheme is presented being based on the proposed algorithm.

DLT, based on Blockchain technology, also allows creating a decentralized environment where any third party does not control transactions and data. Based on this technology and the European Credit Transfer and Accumulation System (ECTS) is discussed in [70]. The work proposes to create a global credit platform for higher education based on Ark Blockchain called EduCTX, which ensures the aspects of user anonymity, data privacy & confidentiality due to potential legal reasons, depending on a country's policy. It is achieved by employing sophisticated, flexible multi-signature blockchain address generation based on the home university and dynamically generated student identifiers. The system itself is expected to further process, manage, and control ECTX tokens in a Peer-to-Peer (P2P) manner. The tokens could be ECTS received by students for courses taken. In this case, universities will act as network partners. Interestingly, the designed schema may face a (multi-)single-point-of-failure attack since taking over one university's key generator may create a flood of newly-produced wallets since students are not involved in their private/public keys production (the university delivers those).

Nonetheless, DLT could be utilized in conjunction with other environments [61]. Here, the TEduChain platform is considered for fundraising in higher education and its' control by students. The users of the platform would be students, fundraisers, and sponsors. The system comprises two different operating environments: a traditional unit managed by a relational database and a blockchain-based DLT.

3.3. Supply chain monitoring

Another popular niche for the application of DLT is the food industry with respect to supply chain monitoring [57]. The proposed DLT-based systems allow analyzing the record of transactions and related metadata. It is designed to consider data confidentiality and integrity, i.e., origin, contracts, process steps, environmental changes, microbiological records, and many others. Today, the ability to track the food movement route is legally required for all participants in this chain. International standards for food traceability are established through a joint program of Food and Agriculture Organization (FAO) and World Health Organization (WHO), and the principles of food traceability are outlined in CAC/GL 60-2006 [34].

From an agricultural perspective, challenges in the supply chain include disconnected stakeholders, limited financing resources, lack of transparency, costly intermediaries, and more. DLT can be employed to trace and track the farming, food processing, and production, distribution, and retail process and provide an unaltered record of food provenance. The IBM Food Trust initiative started with its collaboration with Walmart and has grown into a global consortium that includes big-name companies such as Dole, Driscoll's, Kroger, Nestle, Tyson, and Unilever. The improved data traceability provided by the IBM platform reduced the time it took to trace a mango from the store back to its source from 7 days to 2.2 seconds [29]. That reduction in time enables companies to identify contaminated supply chains and recall affected products before consuming and cause illness.

As an extension of Amazon's web services, the e-commerce giant offers blockchain tools for companies that do not want to create their own. In Australia, Nestle used the Amazon blockchain product to help launch its new coffee brand, Chain of Origin. The consumers can peer into the coffee supply chain by scanning the QR code and checking where those were planted. Other Amazon blockchain customers include Sony Music Japan, BMW, Accenture, and South Korean brewery Jinju Beer [32].

Biometric Blockchain (BBC) includes individuals' biometric characteristics to uniquely identify users of the system, which can satisfy the growing needs for the logistics of food products [73]. Generally, it is essential after the recent incident with mislabeled food products, which led to the death of a passenger in an airplane [6]. The advantages of using the BBC in food logistics are inevitable: the system aims not only to determine whether the data or labels are genuine but also to indicate the responsible party in case of an accident clearly.

3.4. Intellectual property

The next DLT application sector is related to licensing [64]. The authors consider modern blockchain-based applications that support the licensing and distribution of intellectual property. The paper compares both non-technical and technical application criteria. Non-technical criteria are used to evaluate the application functionality range, while technical criteria aim to study the technology used to implement the applications. Finally, eight different platforms were analyzed and classified according to selected criteria.

Authors of [8] analyze Technical Protection Measures, Rights Management Information (RMI), and Digital Rights Management (DRM) while developing the blockchain architecture. Besides, the blockchain-related copyright aspects are highlighted, taking into

consideration the specifics of private orders in terms of copyright are examined. Finally, an interface for the DRM legal protection is developed.

The work [76] examines the use of the blockchain to create limited editions of digital art with a particular focus on the business models of two companies: Monograph and Ascribe. For some, the development of blockchain technologies and smart contracts suggests an opportunity for artists to protect their work from misuse and expropriation. For others, it indicates the possibility of more robust forms of digital rights management going forward, which may negatively impact digital culture. However, this article argues that the aim of limited editions on the blockchain is usually not to institute more substantial restrictions over the use or a new form of digital rights management but rather to create new kinds of tradable digital assets. In turn, this trend implies a different operation of intellectual property rights concerning digital culture, one where alienation rather than exclusion is significant, and a separate operation of scarcity concerning digital cultural goods, where their free circulation is not necessarily antithetical to profit.

3.5. Financial sector

In 2018, an unprecedented shift towards the potential assimilation of DLT infrastructure and the quasi-recognition of cryptocurrencies took place as a new asset class by wealth managers and investors. Many reputable financial intermediaries, including Fidelity Investments, Ameritrade, JPMorgan Chase, and the Intercontinental Exchange, have decided not to remain indifferent to the 21st-century DLT revolution and its consequences for the entire economy [50].

For example, partnering with blockchain leader Guardtime and multiple industry participants, Ernst & Young Global Limited created a blockchain program that connects clients, brokers, insurers, and third parties to distributed common ledgers. These capture data about identities, risks, and exposures and integrate this information with insurance contracts. This program is the first to apply blockchain's transparency, security, and standardization to marine insurance [31].

The work [50] outlines that DLT affects the traditional financial industry and identifies several possible ways for transforming financial services with respect to the DLT ecosystem. The adoption of DLT is expected to take place in three main directions, firstly, through servicing the existing and potential client base both at the retail and institutional levels, secondly, through the improvement of internal and intra-industry processes that remain slow, expensive, and error-prone, and, finally, by tokenizing both liquid and illiquid assets, creating new financial products, and expanding the market. In general, the above changes will open up new opportunities for creating wealth in the financial industry.

The authors of [26] consider the use of DLT to maintain the infrastructure of the securities market, which promises to solve serious challenges of fragmentation and violation of property rights, which also impede market transparency.

At the same time, at least 40 central banks worldwide are currently, or soon will be, researching and experimenting with Central Bank Digital Currency (CBDC) [41]. CBDC, as a commonly proposed application of blockchain and DLT, has attracted much interest within the central banking community for its potential to address long-standing challenges such as financial inclusion, payment efficiency, and both payment system operational and cyber resilience.

Recently, DP Morgan launched its blockchain-based product “Interbank Information Network”, which speeds up cross-border payments between banks by using a shared ledger to resolve delays that arise when, for example, one bank thinks a transfer might violate an international sanction [74].

The subcategories for the classification of the use cases in the finance sector are listed in Table 1 below. Tapscott [60] initially inspired the categories.

Table 1. Financial sector DLT applications.

Subcategories	Application of DLT
ID verification (KYC/AML)	DLTs can provide a trusted way to do customer verification to satisfy Know Your Customer (KYC) and Anti-Money Laundering (AML) obligations, e.g., through past immutable data in the DLT.
Tokenization and stable coins	The digitization of regulated financial products and services such as security/asset tokens and utility tokens and create new ones, e.g., cryptocurrency/payment tokens through tokenization.
Financial management (accounting and auditing)	Smart contracts can automate some accounting processes. Auditing costs can be reduced through cheaper verification of transactions in DLT [17].
Reduction in the risk of fraud	Real-time data is decentralized, and this can increase the trust of the shared information, e.g., management of cash or financial controls, data of maritime industry for insurance purposes, etc.
Funding	DLT creates new revenue opportunities such as new funding models and new types of markets such as equity crowdfunding, secondary market, or new kinds of exchanges.
Investments	Tokenised assets can support the transformation of the regular investments model and promote accessibility to new asset investments.
Regulatory compliance and audit	DLTs can provide accurate and tamper-proof financial, audit, and regulatory reports, improving speed and quality.
Clearing and settlement	Automation and improvement of the centralized clearing and settlement processes using DLT can increase efficiency and reduce costs, time, and agents involved.
Payments and P2P transactions	DLTs can bring new models and arrangements to make payments and transfers faster with lower costs and fewer/no intermediaries. e.g., remodeling correspondent banking, cross-border payments, etc.
New product models	New Peer-to-Peer insurance models can be secured with DLT.

3.6. Energy Supply

In 2019, green renewable energy sources, i.e., biomass, geothermal, hydropower, solar, wind, accounted for 18.49 % of net domestic electrical generation in US [9]. Existing systems can automatically adjust energy absorption, intelligently responding to external on-demand signals from the network. In this case, DLT can be used to transmit real-time data between multiple network participants for more efficient use of resources. The proposed system is based on two key concepts: Transitive Energy Management and P2P communication via DLT. The work [66] provides an assessment of the DLT potential for the energy transition in local markets. The authors propose a new management infrastruc-

ture based on DLT in addition to a novel consensus protocol avoiding additional energy costs.

Authors of [14] also analyze various concepts and technologies for the DLT-based energy transition. Given that traditional centralized energy systems are no longer viable, DLT-based P2P transactional controllers in local energy markets represent the most likely evolutionary step for Smart Grids, as confirmed by various pilot projects. The authors have developed and implemented the infrastructure of transactional management based on blockchain and smart contracts.

DLT shows increasing promise for securing Supervisory Control and Data Acquisition (SCADA) systems in traditional energy grids while enabling distributed energy generators such as rooftop solar panels and electric-vehicle charging stations to access cheaper P2P energy transfers [3].

3.7. Internet of Things

The most broadly developing niche for the DLT application is the IoT. Many projects focus on this combination to solve the Smart City tasks, decentralized applications, cryptocurrency, spectrum sharing, user incentivization, etc. [4] examines the interaction of the IoT and DLT technologies. It focuses on new and broader technical issues related to the security of solutions based on DLT specifically designed for IoT applications. Authors of [24] also analyze how the IoT and DLT interact in terms of connectivity aspects, introducing DLT-based distributed trust network architecture. Finally, they propose a new classification to simplify the DLT synchronization in classic communication networks. The study showed that wireless systems might become a severe challenge for the synchronization protocols' solid functionality.

DLTs based on DAG could also be utilized in several IoT scenarios [21]. This paper analyzes a commonly discussed attack scenario known as a parasite chain attack for the IOTA Foundation's DAG-based ledger. DLT should serve as an invariable and irreversible record of transactions. However, the DAG structure is a more complex mathematical object than its counterparts in the blockchain, namely, it allows branching of the hash tree.

More and more often, IoT comes hand in hand with Artificial Intelligence (AI). The question of the ability to make AI operation safe for humans in the context of distributed systems is discussed in [15]. This study proposes a number of necessary components to enable various AI operation scenarios without harm to people. DLT is an integral part of this proposal, e.g., smart contracts are essential to solve the problem of AI development, which may happen too quickly for prompt human intervention.

The Web of Things (WoT) paradigm is another segment of the IoT concept to overcome the barriers between heterogeneous network environments. WoT aims at solving the compatibility issue by establishing interoperability at the application level. Work [49] expands WoT's vision of decentralized Smart Grids to create a seamless, autonomous, and interoperable environment of technically and economically interconnected systems presented on HEILA's integrated distributed energy resource business platform [48]. Specifically, this work proposes a new network management system and demonstrates its architecture in the context of WoT design patterns.

Finally, the limited spectrum becomes one of the drivers for applying underlying Blockchain technology to incentivize mobile users and operators to share underused li-

censed spectrum [58] and/or their computational resources to other nodes in need [78]. Those concepts are expected to find their niche on the Edge and Fog paradigms.

To sum up this section, we overview the leading sectors and related existing solutions in Table 2.

4. Main integration challenges

Any technological revolution brings the problems of adapting already formed systems to new processes. Based on the critical review, the primary identified issues of using and implementing DLT are system scalability, DLT design aspects, trust management, identity, security, and data management. Let us consider in more detail each of these challenges.

Scalability issues can create bottlenecks in throughput and processing speed, affected by the consensus mechanism, a number of nodes, and network performance. Many applications must be able to process transactions at a certain rate, and the ability to provide such performance remains a massive obstacle to adoption [1]. Over the past decade, many types of distributed services have become increasingly widespread. This trend is primarily due to the number of users accessing such applications, which are not necessarily humans but preferably various autonomous nodes. Accordingly, such distributed systems require a high level of scalability. While scalability of algorithms has always been a focus for research for a long time, even a carefully designed system is often limited in practice when scaling to such large scale applications, often requiring the developing company to optimize the existing software for higher scalability, despite careful previous design [11].

Since the blockchain's advent, we have seen many kaleidoscopic applications based on the DLT, including applications for financial services, healthcare, or the Internet of Things. In addition to scalability, the DLT design also plays an essential role in running viable applications on DLT. For each sector, different approaches to system design can be applied. Each application has specific DLT performance requirements, e.g., high throughput, scalability, etc. However, an important issue is to find a balance between DLT systems and the prevention of high load [38].

At the same time, end-user smart devices have great potential to be available to external entities as a service for various applications or processes. Along with these, the question of security and, in particular, trust between devices arises [65]. Considering the actual use of DLT and a set of use cases, the following subcategories were defined, which are further described in the following subclauses: identity management, security, data management, management, and Decentralized Autonomous Organizations (DAOs) as well as general crypto infrastructure [1]. DAOs, in a broad sense, are digital entities that manage assets and operate autonomously in a decentralized system and rely on individuals tasked to perform certain functions that the automaton itself cannot. While a comprehensive DLT-based digital identity solution can focus on three essential tasks: security, privacy, and portability. DLT technology can offer a way to solve this problem with or without a reliable central authority. In particular, individuals and legal entities can store and authenticate their identity in DLT, giving them greater control over who has their personal information and how they get access to it [28]. Security management identifies an organization's assets (including people, buildings, cars, systems, and information assets), followed by developing, documentation, and implementing policies and procedures to protect these assets. The organization uses security management procedures such as

Table 2. Main challenges and potential solutions.

Challenges	SGM.	REF.	Brief description	Example	Potential solutions
Lack of an ability to determine data ownership	H, IR, GA	[46], [55]	A challenge is the inability to determine which data belongs to whom in real-time reliably. It emanates from the fact that healthcare facilities have numerous users who own multiple devices, thereby creating an N x M data source heterogeneity and complexities for the streaming process.	Petri Net	An enhanced Petri Nets service model aids with a transparent data trace route generation, tracking, and the possible detection of medical data compromises.
Lack of a unified system for patients' data storage	H, DA	[5], [23], [77]	Currently used cloud storage facilities for clinic networks but does not provide access to the healthcare system as a whole. In addition, the patient cannot track his/her personal data access rights.	DAG, IOTA, Tangle, MAM, Merkle tree, Solidity, Ethereum, smart contracts	A blockchain-powered feature that allows patients to securely access and shares their medical data.
The need for high confidentiality	H, GA, IR	[53], [27]			System with entries about treatment offers, participants who submitted their data or rejected the offer
Lack of the unified and international competency system	E	[54]	Employers and educational institutions cannot receive complete information about the competencies, place of study, and educational opportunities of applicants.	Solidity, Ethereum, blockchain	Decentralized Higher Education Lending System, Certification System, Register of professional competencies of the population
Lack of a unified certification system	E, DA, GA	[70]			
Lack of manageable sponsorship for students	E	[61]			
Lack of active supply tracking ability	SC, DA	[57]	Lack of a unified system for tracking products, their origin, and movement	Blockchain, BBC	Tracking and Identification Systems
Lack of supply chain information immutability	SC, DA	[73]	At the moment, there is no way to determine who is responsible for incorrect labeling		
Inability to license digital assets	SC	[64], [8]	Licensing and ongoing tracking of digital assets that can be easily copied	Blockchain, RMI, DRM	Licensing application and digital protection interface
Interaction between IoT and DLT	IoT, DA, GA	[4], [24]	The need for distributed information security enablers	DAG, IOTA, Tangle, MCMC IOTA, blockchain, smart contracts	Special algorithms and protocols
Artificial Intelligence Security	IoT, DA	[15]	Artificial Intelligence Development Process Management		Negative impact analysis and sce- nario correction
Trust aspects	IoT, DA	[65], [33]	The need for trust in the systems interaction	GAIA, UML, blockchain with PBFT	DLT-based technical concept
User incentivization	IoT, DA	[58], [78]	Attracting new users to share limited resources	Consensus protocols, tokenization	Deep adoption by the operators and big market players
Violation of property rights	IR	[26]	Market transparency	All DLT methods, tokenization	Transformation of financial services infrastructure
Regulation of loads on electric networks	DA, ES	[66]	Previously, to prevent overload, they used to dump underused electricity	P2P, consensus protocols	Distributed power systems, Development of a transactional management infrastructure, Development of an integrated business platform for distributed energy resources
Energy efficiency	DA, ES	[14]		Smart contracts, HEILA, WoT	
Compatibility issue between heterogeneous services	IoT, ES	[48]	Overcoming barriers between network environments		
Scalability	IoT, DA, GA	[11], [38], [79]	An increase in the load on the centralized node occurs with an increase in traffic or the number of users	All DLT methods, DAG, vDLT, Sharding, Sidechain, and cross-chain	Creation of applications, where data exchange is based on the distributed registry principles
Identity, security, and data management	IoT, IR, GA	[28], [45]	The identification of an organization's assets (including people, buildings, cars, systems, and information assets)		
DLT design in system functionality	GA	[52], [30]	Depending on the chosen design, the system may have certain functions.	All DLT methods, R3 Corda	Comparison of DLT characteristics by isolating them and compiling groups

H – Healthcare E – Education SC – Supply chains IR – Intellectual rights
F – Finance ES – Energy Supply GA – General industries DA – Distributed architectures

asset and information classification, threat assessment, risk assessment, and risk analysis to identify threats, asset categories, and assess system vulnerabilities so that they can be effectively controlled [45].

The number of challenges related to DLT utilization is indeed vast. The remaining of this section attempts to highlight the most significant ones in more detail.

4.1. System scalability

The study [11] examines the practical consequences of discovering services in systems with a large number of them. Example application areas for this type of system are as follows: Increased automation in homes ("smart home") and urban scenarios ("smart cities") result in large sets of smart components dealing with automation aspects. The study revealed that services could appear and disappear dynamically, negatively affecting the system's general condition.

One of the solutions to the scalability problem is proposed in [38]. Developers highlight the need for a compromise between DLT non-functional properties (e.g., availability and consistency), so following one feature's requirements may hamper the others. Thus, if one DLT architecture can be ideally suited for a specific use case, its application for other scenarios can be detrimental, which stimulates the appearance of various DLT constructs (for example, Ethereum, IOTA, or Tezos). The authors have identified and characterized existing inter-blockchain integration features (from the English Cross-Chain Technology (CCT)). Then, highlighted characteristics systematized, making it more comprehensive for future comparisons. CCT can potentially extend the functionality of DLT design-based applications (such as Hyperledger Fabric), allowing payments to be made, for example, through Ethereum.

Authors of [36] analyzed the problematics of compromise between DLT characteristics from a universal DLT creating perspective. This paper presents a comprehensive set of 49 DLT characteristics synthesized from the DLT literature that were considered relevant for consideration in viable DLT applications. Besides, an in-depth analysis of the dependencies and tradeoffs between DLT characteristics was performed. Finally, the authors identified and combined 26 compromises in 6 archetypes.

In work [79], the authors describe the blockchain performance problem, mainly paying attention to scalability, and then classify the existing mainstream solutions (Sharding, Sidechain, and cross-chain) into several representative layers.

The use of DLT can also be an effective way to solve the problems of highly loaded systems. For example, the government is already facing issues handling the data generated by its' internal subdivisions [71] while moving to dynamic and distributed systems may maintain a register to store information about enterprises, including their identification number and name. However, a centralized registry-supporting service is the one point of failure for the entire system. Work [69] presents an online tool for generating and deploying registries based on smart contracts with access by Representational State Transfer (REST)-ful Application Programming Interface (API).

Another way to address the scalability issue is to virtualize system elements [75]. In particular, the authors propose a virtualization layer for DLT (vDLT), which abstracts the primary resources, such as, for example, hardware, computing, storage, network, etc. By providing a logical layer of resources, vDLT can significantly improve performance, facilitate system changes, and simplify the management and configuration of DLT. This

paper also describes several vDLT options, including DAG-based vDLTs and blockchain-based vDLTs, side-channel mechanisms in vDLTs, and separation of control from traffic.

4.2. DLT design aspects

Authors of [52] suggest using an architecture based on the proprietary blockchain of the R3 Corda solution. It was designed with a different network architecture compared to existing blockchain systems. Here, the node can operate on the user's personal mobile device. It allows users to store and manage their data directly and exchange data with authorized network participants. According to the authors, the main challenge in the DLT design is the need for structural comparison of designs.

The article [30] outlines DLT characteristics that are selected according to an in-depth comparison of DLT elements. In addition, a benchmarking process is proposed to structure of the DLT projects according to the application requirements.

Despite the lack of practical examples of the comparative characteristics of DLT, work [37] states that the DLT design must be tailored to specific contextual requirements. A successful DLT configuration requires a deep understanding of DLT features and their interdependencies. This study examines 37 DLT characteristics divided into six archetypes (maximum utilization, maximum development flexibility, maximum productivity, maximum anonymity, maximum security, maximum institutionalization), aiming to define the appropriate way of selection.

4.3. Trust management

DLT and IoT always come along with the concept of trust management. The authors of [65] propose a fully decentralized architecture of the Machine-to-Machine (M2M) communication system, where the services can be quickly and flexibly adapted to the specific application requirements. Service delivery's decentralized nature eliminates a single point of failure and improves operational efficiency by intellectually allocating the resources among participants.

Work [33] also examines the use of blockchain technology as a tool for trust management. The authors propose applying club governance to the technical design and development of a number of DLT systems, including cryptocurrencies and corporate applications. It is expected to lead to the emergence of cyber-physical systems. Due to the use of stand-alone systems, new problems arise, associated, on the one hand, with the need for consistency, flexibility, and interoperability, as well as to the reliability of the systems interoperation.

Authors of [42] propose a technical concept for new production systems based on DLT. The proposed system is based on the Gaia method. The Gaia is an agent-based development method closely related to object-oriented development methods. Gaia is based on presenting a multi-agent system as a computing system consisting of nodes with different roles. It focuses on the safety and reliability of cyber-physical environments. The authors recommend a blockchain structure with a coordination mechanism based on the Byzantine generals' task as a basis for their DLT system.

4.4. Identity, security and data management

The emergence of DLT has given rise to new approaches to identity management aiming to upend dominant approaches to providing and consuming digital identities. These new approaches to Identity Management (IdM) propose to enhance decentralization, transparency, and user control in transactions that involve identity information. This paper introduces the emerging landscape of DLT-based IdM and evaluates three representative proposals – uPort, ShoCard, and Sovrin – using the analytic lens of a seminal framework that characterizes the nature of successful IdM schemes [28].

In [45], a cryptographic membership authentication scheme, i.e., authenticating graph data. was proposed to support Blockchain-based Identity Management Systems (BIMS). The system is designed to bind a digital identity object to its real-world entity. Specifically introduced a new Transitively Closed Undirected Graph Authentication (TCUGA) scheme, which only needs to use node signatures, e.g., certificates for identifying nodes. The trapdoor hash function used in the scheme allows the signer to update the certificates without re-signing the nodes efficiently. This scheme is efficient even though the graph dynamically adds or deletes vertices and edges.

DLT can be used to create new tools to realize governance for a decentralized global public utility for self-sovereign identity on the Internet, for which a new term has been put forward as DAOs. A DAO is similar to a regular corporation in that it is a separate entity and has its bank account (here it is cryptocurrency wallet) and ID number (the contact address). The main difference is that a DAO is autonomous. In contrast to regular corporations, a DAO is managed by itself (its code) rather than by humans (in the form of executive management, i.e., the CEO). The benefit here is that it is a public offering and open for everyone so that it gives equal rights to all token holders, brings in more liquidity, and makes it easier to buy and sell stocks. DAOs provide the organization a high level of transparency, which means less opportunity for corruption and less administrative costs [19].

Thus, the use of the DLT allows not only to improve the processes in the sectors of health care, education, finance, and others presented above, but also allows to solve problems that arise as in modern systems when they are scaled and developed, but also when DLT itself is applied.

Based on the critical review, we quantitatively analyzed the complexity to overcome the main DLT integration challenge and presented the results in Table 3.

5. Discussion

The evolution of centralized systems towards the distributed ones is a complicated step with numerous challenges to be solved. The primary outcomes of this topical review are as follows. First, it surveys the main representative applications of the DLT operation. Next, it outlines the main related challenges and, finally, highlights recommended solutions provided by other researchers in a critical review manner.

The results identify that the major problem for most DLT-based systems at the moment is still scalability. It is mainly due to an increase in autonomous users in contrast to conventional human-oriented system design, which significantly affects the network and overall system load negatively. Still, many researchers foresee the future of DLT systems

Table 3. Main DLT integration challenges.

Parameter	System scalability	DLT design aspects	Trust management	Identity, security and data management
Migration from centralized system				
Development-related costs				
Integration-related costs				
Maintenance-related costs				
Complexity of implementation				
Main challenge	Dynamically appearing services [11]	A lot of characteristics and services [37]	Requests from untrusted sources [42]	Handling various data sources [28]

Characteristic: ■ – low, ■ – moderate, ■ – high, ■ – very high

as avoidance of the centralized systems' overload and additional protection of the private data. Finally, this review provides a comparative analysis of the challenges that could become a baseline for potential future research activities in the DLT field.

While there have been many diverse efforts in different research directions, we outlined that there are still many open questions, no universal solutions, and significant space for future research and experimentation. We conclude that DLT has great potential to support the economies, while many problems are still to be solved and carefully considered.

Acknowledgments. Project group was funded by the Graduate School of Business National Research University Higher School of Economics (HSE). The research was financed by the Technology Agency of the Czech Republic (TACR) under grant no. TK02030013.

List of Acronyms

DLT	Distributed Ledger Technology
AI	Artificial Intelligence
AML	Anti-Money Laundering
API	Application Programming Interface
BBC	Biometric Blockchain
BIMS	Blockchain-based Identity Management Systems
CBDC	Central Bank Digital Currency
CCT	Cross-Chain Technology
DAG	Directed Acyclic Graph
DAO	Decentralized Autonomous Organization
DRM	Digital Rights Management
ECTS	European Credit Transfer and Accumulation System
EHR	Electronic Health Records
FAO	Food and Agriculture Organization
GDPR	General Data Protection Regulation
ICM	Insurance Content Management
ICT	Information and Communication Technologies
IdM	Identity Management
IoT	Internet of Things
KYC	Know Your Customer
M2M	Machine-to-Machine
MAM	Masked Authenticated Messaging
P2P	Peer-to-Peer
REST	Representational State Transfer
RMI	Rights Management Information
SCADA	Supervisory Control and Data Acquisition
TCUGA	Transitively Closed Undirected Graph Authentication
TPM	Technical Protection Measures
vDLT	Virtual Distributed Ledger Technology
WHO	World Health Organization
WoT	Web of Things

References

1. Technical Report FG DLT D2.1 Distributed Ledger Technology Use Cases. Tech. rep., ITU-T Focus Group on Application of Distributed Ledger Technology (FG DLT) (August 2019)
2. Agbo, C.C., Mahmoud, Q.H., Eklund, J.M.: Blockchain Technology in Healthcare: A Systematic Review. In: Healthcare. vol. 7, p. 56. Multidisciplinary Digital Publishing Institute (2019)
3. Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., McCallum, P., Peacock, A.: Blockchain Technology in the Energy Sector: A Systematic Review of Challenges and Opportunities. Renewable and Sustainable Energy Reviews 100, 143–174 (2019)
4. Arslan, S.S., Jurdak, R., Jelitto, J., Krishnamachari, B.: Advancements in Distributed Ledger Technology for Internet of Things (2019)

5. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: Using Blockchain for Medical Data Access and Permission Management. In: Proc. of 2nd International Conference on Open and Big Data (OBD). pp. 25–30. IEEE (2016)
6. BBC: Pret Inquest: Baguette Allergy Alerts Before Girl's Death. BBC News (2018)
7. Benedict, G.: Challenges of DLT-enabled Scalable Governance and the Role of Standards. Journal of ICT Standardization (2019)
8. Bodó, B., Gervais, D., Quintais, J.P.: Blockchain and Smart Contracts: The Missing Link in Copyright Licensing? International Journal of Law and Information Technology 26(4), 311–336 (2018)
9. Bossong, K.: Solar and Wind Energy Provide Almost 10 Percent of Total Generation in the US in 2019. Renewable Energy World (Oct 2019)
10. Bouras, M.A., Lu, Q., Zhang, F., Wan, Y., Zhang, T., Ning, H.: Distributed Ledger Technology for eHealth Identity Privacy: State of The Art and Future Perspective. Sensors 20(2), 483 (2020)
11. Braubach, L., Jander, K., Pokahr, A.: A Novel Distributed Registry Approach for Efficient and Resilient Service Discovery in Megascale Distributed Systems. Computer Science & Information Systems 15(3) (2018)
12. Brogan, J., Baskaran, I., Ramachandran, N.: Authenticating Health Activity Data Using Distributed Ledger Technologies. Computational and Structural Biotechnology Journal 16, 257–266 (2018)
13. Burns, J.: EHR Interoperability's Uncertain Future. Medical Economics (2016)
14. Cali, U., Çakir, O.: Energy Policy Instruments for Distributed Ledger Technology Empowered Peer-to-Peer Local Energy Markets. IEEE Access 7, 82888–82900 (2019)
15. Carlson, K.W.: Safe Artificial General Intelligence via Distributed Ledger Technology. arXiv preprint arXiv:1902.03689 (2019)
16. Castro, M., Liskov, B.: Practical Byzantine Fault Tolerance and Proactive Recovery. ACM Transactions on Computer Systems (TOCS) 20(4), 398–461 (2002)
17. Catalini, C., Jagadeesan, R., Kominers, S.D.: Market Design for a Blockchain-Based Financial System. Available at SSRN 3396834 (2019)
18. Chen, G., Xu, B., Lu, M., Chen, N.S.: Exploring Blockchain Technology and its Potential Applications for Education. Smart Learning Environments 5(1), 1 (2018)
19. Coita, D.C., Abrudan, M.M., Matei, M.C.: Effects of the Blockchain Technology on Human Resources and Marketing: An Exploratory Study. In: Strategic Innovative Marketing and Tourism, pp. 683–691. Springer (2019)
20. Collomb, A., Sok, K.: Blockchain/Distributed Ledger Technology (DLT): What Impact on the Financial Sector? Digiworld Economic Journal (103) (2016)
21. Cullen, A., Ferraro, P., King, C., Shorten, R.: Distributed Ledger Technology for IoT: Parasite Chain Attacks. arXiv preprint arXiv:1904.00996 (2019)
22. Cunningham, J., Ainsworth, J.: Enabling Patient Control of Personal Electronic Health Records Through Distributed Ledger Technology. Stud Health Technol Inform 245, 45–48 (2018)
23. Dagher, G.G., Mohler, J., Milojkovic, M., Marella, P.B.: Ancile: Privacy-preserving Framework for Access Control and Interoperability of Electronic Health Records Using Blockchain Technology. Sustainable Cities and Society 39, 283–297 (2018)
24. Danzi, P., Kalør, A.E., Sørensen, R.B., Hagelskjær, A.K., Nguyen, L.D., Stefanović, Č., Popovski, P.: Communication Aspects of the Integration of Wireless IoT Devices with Distributed Ledger Technology. arXiv preprint arXiv:1903.01758 (2019)
25. Dodevski, Z., Filiposka, S., Mishev, A., Trajkovik, V.: Real Time Availability and Consistency of Health-related Information Across Multiple Stakeholders: A Blockchain Based Approach. Computer Science and Information Systems (00), 17–17 (2021)
26. Donald, D.C., Miraz, M.H.: Multilateral Transparency for Securities Markets through DLT. The Chinese University of Hong Kong Faculty of Law Research Paper (2019-05) (2019)

27. Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F.: Secure and Trustable Electronic Medical Records Sharing Using Blockchain. In: Proc. of AMIA Annual Symposium. vol. 2017, p. 650. American Medical Informatics Association (2017)
28. Dunphy, P., Petitcolas, F.A.: A First Look at Identity Management Schemes on the Blockchain. IEEE Security & Privacy 16(4), 20–29 (2018)
29. Galvin, D.: Blockchain for Food Safety. IBM and Walmart (2017)
30. Gräbe, F., Kannengießer, N., Lins, S., Sunyaev, A.: Do Not Be Fooled: Toward a Holistic Comparison of Distributed Ledger Technology Designs. In: Proc. of 53rd Hawaii International Conference on System Sciences (Forthcoming) (2020)
31. Hamish, T.: How Are You Using Blockchain to Reimagine Your Industry? (2020)
32. Holbrook, J.: Architecting Enterprise Blockchain Solutions. John Wiley & Sons (2020)
33. Howell, B.E., Potgieter, P.H., Sadowski, B.M.: Governance of Blockchain and Distributed Ledger Technology Projects. Available at SSRN 3365519 (2019)
34. Joint FAO/WHO Codex Alimentarius Commission, F., Agriculture Organization of the United Nations, J.F.F.S.P.: Codex Alimentarius: Food Import and Export Inspection and Certification Systems. World Health Organ (2007)
35. Kalloniatis, C., Lambrinoudakis, C., Musahl, M., Kanatas, A., Gritzalis, S.: Incorporating Privacy by Design in Body Sensor Networks for Medical Applications: A Privacy and Data Protection Framework. Computer Science and Information Systems 18(1), 323–350 (2021)
36. Kannengießer, N., Lins, S., Dehling, T., Sunyaev, A.: Mind the Gap: Trade-Offs between Distributed Ledger Technology Characteristics. arXiv preprint arXiv:1906.00861 (2019)
37. Kannengießer, N., Lins, S., Dehling, T., Sunyaev, A.: What Does Not Fit Can be Made to Fit! Trade-Offs in Distributed Ledger Technology Designs. In: Proc. of the 52nd Hawaii International Conference on System Sciences (2019)
38. Kannengießer, N., Pfister, M., Greulich, M., Lins, S., Sunyaev, A.: Bridges Between Islands: Cross-Chain Technology for Distributed Ledger Technology. In: Proc. of 53rd Hawaii International Conference on System Sciences (2020)
39. Lamberti, R., Fries, C., Lücking, M., Manke, R., Kannengießer, N., Sturm, B., Komarov, M.M., Stork, W., Sunyaev, A.: An Open Multimodal Mobility Platform Based on Distributed Ledger Technology. In: Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pp. 41–52. Springer (2019)
40. Lamport, L.: The Weak Byzantine Generals Problem. Journal of the ACM (JACM) 30(3), 668–676 (1983)
41. Lannquist, A.: Central Banks and Distributed Ledger Technology: How are Central Banks Exploring Blockchain Today. Report of World Economic Forum (2019)
42. Lebioda, A., Lachenmaier, J., Burkhardt, D.: Control of Cyber-Physical Production Systems: A Concept to Increase the Trustworthiness within Multi-Agent Systems with Distributed Ledger Technology. In: Proc. of 23rd Pacific Asia Conference on Information Systems (PACIS) (2019)
43. Li, P., Nelson, S.D., Malin, B.A., Chen, Y.: DMMS: A Decentralized Blockchain Ledger for the Management of Medication Histories. Blockchain in Healthcare Today (2019)
44. Liberati, A., Altman, D.G., Tetzlaff, J., Mulrow, C., Gøtzsche, P.C., Ioannidis, J.P., Clarke, M., Devereaux, P.J., Kleijnen, J., Moher, D.: The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies that Evaluate Health Care Interventions: Explanation and Elaboration. Journal of clinical epidemiology 62(10), e1–e34 (2009)
45. Lin, C., He, D., Huang, X., Khan, M.K., Choo, K.K.R.: A New Transitively Closed Undirected Graph Authentication Scheme for Blockchain-based Identity Management Systems. IEEE Access 6, 28203–28212 (2018)
46. Lomotey, R.K., Pry, J., Sriramoju, S.: Wearable IoT Data Stream Traceability in a Distributed Health Information System. Pervasive and Mobile Computing 40, 692–707 (2017)
47. Mäkitalo, N., Ometov, A., Kannisto, J., Andreev, S., Koucheryavy, Y., Mikkonen, T.: Safe, Secure Executions at the Network Edge: Coordinating Cloud, Edge, and Fog Computing. IEEE Software 35(1), 30–37 (2017)

48. Mashlakov, A., Keski-Koukkari, A., Romanenko, A., Tikka, V., Jafary, P., Supponen, A., Markkula, J., Aro, M., Abdurafikov, R., Kulmala, A., et al.: Integrated Business Platform of Distributed Energy Resources–HEILA (2019)
49. Mashlakov, A., Keski-Koukkari, A., Tikka, V., Kulmala, A., Repo, S., Honkapuro, S., Aro, M., Jafary, P.: Uniform Web of Things Based Access to Distributed Energy Resources via Metadata Registry (2019)
50. Mazur, M.: Brawl at the Gates: How Distributed Ledger Technology is Transforming the Financial Services Sector. Available at SSRN 3400649 (2019)
51. Merkle, R.C.: Protocols for Public Key Cryptosystems. In: Proc. of IEEE Symposium on Security and Privacy. pp. 122–122. IEEE (1980)
52. Moon, J., Kim, D.: Design of a Personal-Led Health Data Management Framework Based on Distributed Ledger. The Journal of Society for e-Business Studies 24(3), 73–86 (2019)
53. Moreira, M.W., Rodrigues, J.J., Sangaiah, A.K., Al-Muhtadi, J., Korotaev, V.: Semantic Interoperability and Pattern Classification for a Service-oriented Architecture in Pregnancy Care. Future Generation Computer Systems 89, 137–147 (2018)
54. Novikov, S.P., Mikheenko, O.V., Kulagina, N.A., Kazakov, O.D.: Digital Registry of Professional Competences of the Population Drawing on Distributed Registries and Smart Contracts Technologies. Business Informatics (4 (46)) (2018)
55. Ometov, A., Bezzateev, S.V., Kannisto, J., Harju, J., Andreev, S., Koucheryavy, Y.: Facilitating the Delegation of Use for Private Devices in the Era of the Internet of Wearable Things. IEEE Internet of Things Journal 4(4), 843–854 (2016)
56. Pagliari, C., Sloan, D., Gregor, P., Sullivan, F., Detmer, D., Kahan, J.P., Oortwijn, W., MacGillivray, S.: What is eHealth (4): A Scoping Exercise to Map the Field. Journal of Medical Internet Research 7(1), e9 (2005)
57. Pearson, S., May, D., Leontidis, G., Swainson, M., Brewer, S., Bidaut, L., Frey, J.G., Parr, G., Maull, R., Zisman, A.: Are Distributed Ledger Technologies the panacea for food traceability? Global Food Security 20, 145–149 (2019)
58. Pirmagomedov, R., Ometov, A., Moltchanov, D., Lu, X., Kovalchukov, R., Olshannikova, E., Andreev, S., Koucheryavy, Y., Dohler, M.: Applying Blockchain Technology for User Incentivization in mmWave-Based Mesh Networks. IEEE Access 8, 50983–50994 (2020)
59. Poongodi, T., Krishnamurthi, R., Indrakumari, R., Suresh, P., Balusamy, B.: Wearable Devices and IoT. In: A Handbook of Internet of Things in Biomedical and Cyber Physical System, pp. 245–273. Springer (2020)
60. Radziwill, N.: Blockchain Revolution: How the Technology Nehind Bitcoin is Changing Money, Business, and the World. The Quality Management Journal 25(1), 64–65 (2018)
61. Rashid, M.A., Deo, K., Prasad, D., Singh, K., Chand, S., Assaf, M.: TEduChain: A Platform for Crowdsourcing Tertiary Education Fund using Blockchain Technology. arXiv preprint arXiv:1901.06327 (2019)
62. Rauchs, M., Glidden, A., Gordon, B., Pieters, G.C., Recanatini, M., Rostand, F., Vagneur, K., Zhang, B.Z.: Distributed Ledger Technology Systems: A Conceptual Framework. Available at SSRN 3230013 (2018)
63. Regulation, General Data Protection: Regulation EU 2016/679 of the European Parliament and of the Council of 27 April 2016. Official Journal of the European Union (2016)
64. Schoenhals, A., Hepp, T., Leible, S., Ehret, P., Gipp, B.: Overview of Licensing Platforms based on Distributed Ledger Technology. In: Proc. of 52nd Hawaii International Conference on System Sciences (2019)
65. Shala, B., Trick, U., Lehmann, A., Ghita, B., Shiaeles, S.: Distributed Ledger Technology for Trust Management Optimisation in M2M. In: Proc. of Mobile Communication-Technologies and Applications. pp. 1–6. VDE (2019)
66. Siano, P., De Marco, G., Rolán, A., Loia, V.: A Survey and Evaluation of the Potentials of Distributed Ledger Technology for Peer-to-Peer Transactive Energy Exchanges in Local Energy Markets. IEEE Systems Journal (2019)

67. Smetanin, S., Ometov, A., Kannengießer, N., Sturm, B., Komarov, M., Sunyaev, A.: Modeling of Distributed Ledgers: Challenges and Future Perspectives. In: Proc. of 22nd Conference on Business Informatics (CBI). vol. 1, pp. 162–171. IEEE (2020)
68. Thiebes, S., Kannengießer, N., Schmidt-Kraepelin, M., Sunyaev, A.: Beyond Data Markets: Opportunities and Challenges for Distributed Ledger Technology in Genomics. In: Proc. of Hawaii International Conference on System Sciences (2020)
69. Tran, A.B., Xu, X., Weber, I., Staples, M., Rimba, P.: Regerator: A Registry Generator for Blockchain. In: CAiSE-Forum-DC. pp. 81–88 (2017)
70. Turkanović, M., Hölbl, M., Košič, K., Heričko, M., Kamišalić, A.: EduCTX: A Blockchain-based Higher Education Credit Platform. IEEE access 6, 5112–5127 (2018)
71. Veljković, N., Milić, P., Stoimenov, L., Kuk, K.: Production of Linked Government Datasets Using Enhanced LIRE Architecture. Computer Science and Information Systems 17(2), 599–617 (2020)
72. Wohrer, M., Zdun, U.: Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity. In: Proc. of International Workshop on Blockchain Oriented Software Engineering (IWBOSE). pp. 2–8. IEEE (2018)
73. Xu, B., Agbele, T., Jiang, R.: Biometric Blockchain: A Better Solution for the Security and Trust of Food Logistics. In: Proc. of IOP Conference Series: Materials Science and Engineering. vol. 646, p. 012009. IOP Publishing (2019)
74. Yermack, D., Fingerhut, A.: Blockchain Technology's Potential in the Financial System. Tech. rep., NYU Stern School of Business, National Bureau of Economic Research, European Corporate Governance Institute (May 2019)
75. Yu, F.R., Liu, J., He, Y., Si, P., Zhang, Y.: Virtualization for Distributed Ledger Technology (vDLT). IEEE Access 6, 25019–25028 (2018)
76. Zhang, P., Schmidt, D.C., White, J., Lenz, G.: Blockchain Technology Use Cases in Healthcare. In: Advances in Computers, vol. 111, pp. 1–41. Elsevier (2018)
77. Zheng, X., Sun, S., Mukkamala, R.R., Vatrapu, R., Ordieres-Meré, J.: Accelerating Health Data Sharing: A Solution Based on the Internet of Things and Distributed Ledger Technologies. Journal of medical Internet research 21(6), e13583 (2019)
78. Zhidanov, K., Bezzateev, S., Afanasyeva, A., Sayfullin, M., Vanurin, S., Bardinova, Y., Ometov, A.: Blockchain Technology for Smartphones and Constrained IoT Devices: A Future Perspective and Implementation. In: Proc. of IEEE 21st Conference on Business Informatics (CBI). vol. 2, pp. 20–27. IEEE (2019)
79. Zhou, Q., Huang, H., Zheng, Z., Bian, J.: Solutions to Scalability of Blockchain: A Survey. IEEE Access 8, 16440–16455 (2020)
80. Zhu, Q., Loke, S.W., Trujillo-Rasua, R., Jiang, F., Xiang, Y.: Applications of Distributed Ledger Technologies to the Internet of Things: A Survey. ACM Computing Surveys (CSUR) 52(6), 1–34 (2019)

Maria Gorbunova is currently a Project Manager at Rostelecom, Moscow, Russia. She received her B.Sc. degree from Plekhanov Russian University of Economics in 2018 and M.Sc. in E-Business degree from National Research University Higher School of Economics in 2020. Her research interest are in Project Management, Information Technology, Blockchain, Internet of Things.

Pavel Masek received the M.Sc. and Ph.D. degrees in Electrical Engineering from the Faculty of Electrical Engineering and Communication at the Brno University of Technology (BUT), Czech Republic, in 2013 and 2017, respectively. He is currently a researcher at the Department of Telecommunications, BUT. Pavel is also co-supervising

the WISLAB research group, where his current research interests include various aspects in the area of heterogeneous wireless communication networks and systems, the Internet of Things, and Industry 4.0-driven research projects. Pavel (co-) authored more than 90 research works on a variety of networking-related topics in internationally recognized venues, including those published in the IEEE Communications Magazine, as well as several technology products.

Mikhail Komarov is a Professor at the Department of Business Informatics, Graduate School of Business, National Research University Higher School of Economics. He is a specialist in wireless data transmission and IT. He is IEEE Senior Member, Vice-chair of the Special Interest Group on IoT at the Internet Society.

Aleksandr Ometov is a Postdoctoral Research Fellow at Tampere University (TAU), Finland. He is currently working on A-WEAR and APROPOS MSCA projects. He received D.Sc. (Tech) in Telecommunications and M.Sc. in Information Technology from Tampere University of Technology (TUT), Finland, in 2018 and 2016, correspondingly. He also received his Specialist degree in Information Security from Saint-Petersburg State University of Aerospace Instrumentation (SUAI), Russia, in 2013. His research interests are wireless communications, information security, blockchain technology, and wearable applications.

Received: February 15, 2021; Accepted: July 1, 2021.

Computer Science and Information Systems 19(1):87–116 https://doi.org/10.2298/CSIS201229045I

Entropy-based Network Traffic Anomaly Classification Method Resilient to Deception

Juma Ibrahim and Slavko Gajin

University of Belgrade – School of Electrical Engineering,
Bul. kralja Aleksandra 73, 11000 Belgrade, Serbia
jumaibrahim04@yahoo.com
slavko.gajin@rcub.bg.ac.rs

Abstract. Entropy-based network traffic anomaly detection techniques are attractive due to their simplicity and applicability in a real-time network environment. Even though flow data provide only a basic set of information about network communications, they are suitable for efficient entropy-based anomaly detection techniques. However, a recent work reported a serious weakness of the general entropy-based anomaly detection related to its susceptibility to deception by adding spoofed data that camouflage the anomaly. Moreover, techniques for further classification of the anomalies mostly rely on machine learning, which involves additional complexity. We address these issues by providing two novel approaches. Firstly, we propose an efficient protection mechanism against entropy deception, which is based on the analysis of changes in different entropy types, namely Shannon, Rényi, and Tsallis entropies, and monitoring the number of distinct elements in a feature distribution as a new detection metric. The proposed approach makes the entropy techniques more reliable. Secondly, we have extended the existing entropy-based anomaly detection approach with the anomaly classification method. Based on a multivariate analysis of the entropy changes of multiple features as well as aggregation by complex feature combinations, entropy-based anomaly classification rules were proposed and successfully verified through experiments. Experimental results are provided to validate the feasibility of the proposed approach for practical implementation of efficient anomaly detection and classification method in the general real-life network environment.

Keywords: anomaly classification, anomaly detection, entropy, entropy deception, network behaviour analysis.

1. Introduction

The increasing complexity of modern networks is accompanied by constant changes in the security threat landscape. Signature-based intrusion detection methods are inefficient in detecting cryptographic traffic and zero-day attacks, while the intelligence put on the firewall does not protect from internal network usage. Therefore, network anomaly detection based on traffic pattern behaviour analysis is now recognized as a mandatory part of modern security analytics and protection solutions.

Several studies show that there is a significant interest in implementing entropy-based techniques for network behaviour analysis and anomaly detection 1. Their

efficiency is often demonstrated by using examples with heavily loaded anomalous traffic, such as intensive botnet or DDoS attacks. For attacks with less intensive traffic, such as SYN Flood, Port Scan or Dictionary attacks, the volumetric features do not provide sufficient information. Additional features must be used, such as the flow count and the degree of communication with other peers, the so-called behaviour features 2.

In contrast to widely presented entropy-based anomaly detection methods, significantly fewer efforts have been done on entropy-based anomaly classification. Most authors dealing with anomaly classification propose supervised machine learning techniques, even if detection is based on entropy 34. With such an approach, training with the labelled dataset is required, while the simplicity for practical implementation as one of the main benefits of entropy-based approaches, is significantly diminished.

One of the biggest weaknesses of entropy-based approaches is highlighted in 5, where the authors have shown the method to deceive flow-based detection systems by injecting additional spoofed network traffic during a DDoS attack. To the best of our knowledge, the proper solution to this problem has not been presented yet.

The motivation behind our research was to fill the above-mentioned gaps in this research problem, namely the classification of the detected anomalies which is resilient to entropy deception. The research method was based on conducting a detailed behaviour analysis of various types of anomalies caused by security attacks and investigating how they affect the entropy of the observed features, using various entropy types. The main research goal is to propose the anomaly classification method as an extension to the existing entropy detection systems, which is improved with the protection mechanism against entropy deception.

An important objective for the proposed solution is the feasibility for practical implementation in the general network environment. For this reason, only basic flow features have been chosen because they can be easily collected from network routers using NetFlow protocol 6 or similar industrial standards. The aggregation process is based on combinations of the basic flow attributes and additional so-called behaviour features which are calculated using the aggregation of the second degree. Accordingly, the presented research does not focus on specific attacks and particular use cases forcing the efficiency as high as possible by fine-tuning the parameters, but on providing a robust entropy-based method for both anomaly detection and classification that can be easily implemented in any type of the real-life network traffic.

The rest of the paper is organized as follows: the second section discusses the most relevant scientific publications related to this research. The third section outlines the proposed methodology, while section four presents and discusses the experimental results. Finally, the paper is concluded by summarizing the main contributions and results, and by defining directions for further research.

2. Related Work

Due to the relative simplicity and application in real networks, entropy-based anomaly detection still attracts great interest in the research community 789, along with more complex methods such as classification, clustering, deep learning or statistical-based approaches 10. It often relies on the flow feature distributions, based on data taken from

offline datasets for research purposes, or on data collected from real networks in practical implementation 1112.

A classical approach leverages the well-known Shannon entropy in the context of information theory 13. Feature selection and aggregation are used to generate distributions of all distinct elements and their aggregated metrics 14. A straightforward approach for DDoS attack detection is based on the volumetric feature, either using total byte and packet counts 15161718 1920 or using additionally derived features, such as average packets and bytes per flow 2122. However, volume-based metrics are insufficient for sophisticated attacks and less intensive anomalies.

Lakhina et al. in 23 used entropy measurements to analyse the real traffic aggregated inside the research networks Internet2 in the US and Geant in Europe. Using additionally injected synthetic flows, they found significant advantages of using entropy-based features over the traditional volume-based approach. The authors in 24 extended Lakhina's work using unidirectional flows and host-level granularity, modelling the behaviour for outgoing and incoming traffic.

The authors in 2 further contributed to better understand anomalous behaviour in a real network. They suggested the utilization of bidirectional data flows to avoid the biases arising from unidirectional flow analysis. Then, they analyzed the entropy of volumetric data, flow count, packet size distribution and host in/out-degree of communications with other hosts and reported a strong correlation of address and port features, emphasizing better detection abilities of behaviour features.

In 3, the authors proposed the utilization of parametrized Tsallis entropy 25 to capture separately the regions with high and low activity in the feature distribution. They modelled 20 anomaly types and injecting artificial flows into real background traffic they trained a support vector machine (SVM) to classify the anomalies.

In 26, entropy was used for profiling per-host behaviour in Internet traffic. Each of the source and destination IP addresses and ports was aggregated and the entropies of the three remaining features gave a three-dimensional entropy space with a total of 27 behaviour clusters. It was shown that different anomalies fit into particular clusters with high accuracy.

Bereziński et al. analysed realistic, synthetically generated botnet traffic injected into real flow data 4. They concluded that the parametrised Tsallis and Rényi entropy 27 provide better entropy change detection, depending on the applied parameter. They also confirmed the poor performance of volume-based approaches.

Giotis et al. in 28 presented an effective and scalable anomaly detection mechanism based on OpenFlow and sFlow data sources. The proposed architecture is modular and can accept any detection methods, such as statistical, data mining or machine learning anomaly detection. They validated the concept by adopting an entropy-based approach, using basic attributes from flow tuples, namely source and destination IP addresses and port numbers. Based on entropy changes of these four features, the proposed mechanism can identify the three most prominent network attacks, namely DDoS, port scan, and worm propagation. Using this classification method and taking advantages of SDN environments, they further contributed with an attack mitigation mechanism that identifies the attackers and protects the victim by blocking the attack.

The usability of entropy-based techniques was put under question when the authors in 5 demonstrated a method to deceive entropy-based detection by injecting additional traffic that camouflage the entropy change caused by the attack. The method exploits the simplicity of the entropy approach that transforms the whole data distribution into a

single metric. This work reveals the weakness of entropy-based techniques but has not been addressed well in the scientific literature so far.

Our previous work was focused on anomaly detection problem based on the entropy of flow features. In 29 we proposed architecture of network traffic anomaly detection system feasible for practical implementation, which includes data pre-processing, root-cause analysis, machine learning decision process, and control mechanisms for correcting, fine-tuning, and training the system. In 30 we investigated possibilities to improve the anomaly detection process by finding the outliers in time series data points using unsupervised machine learning techniques.

In this paper, we contribute to the above-mentioned research problems by providing a protection mechanism against deceiving the existing entropy-based anomaly detection techniques, further extended with a comprehensive network traffic anomaly classification method, which are the main novelties in our research.

Similar to most of the related previous work we also use bidirectional flows which are shown that provide more reliable information for the anomaly detection process, such as recognition of asymmetric traffic with no responses. However, our research is primarily based on the flow-count and behaviour features only, since the volumetric features have higher variation and generate more false positive alarms. We have analysed the characteristics of Shannon, Tsallis, and Rényi entropy types, pointing out their advantages and drawbacks. A developed method can accept any entropy types, but it is demonstrated and validated using Shannon entropy.

3. Proposed method

This section presents the problem analysis and the most relevant findings. The feature selection process is formalized and generalized defining the aggregation key features and calculated behaviour features and the feature annotation is proposed accordingly. All the entropy types are analysed in terms of changes in data distributions and their ability to detect anomalous behaviour. This analysis leads to our main contributions - the method for the protection against entropy deception and detection technique improved with the anomaly classification rules using a multivariate analysis of entropy results, which is based on the patterns in the way the features are affected by different anomalies in network communications. In contrast to similar works in this research area which are mostly based on supervised machine learning techniques, our approach is especially suitable for practical implementation in real-life network environments.

3.1. Flow feature selection

Original raw flow records, the so-called flows, are unidirectional, carrying the total packet and byte counts in the direction from the source to the destination. Combining two unidirectional flows from both directions into a single bidirectional flow offers more information about the communication pattern, and this is confirmed to be more useful in anomaly detection 24.

In the client-server communication model, which is considered in an ordinary network operation, the client initiates communication as a source in the bidirectional

flow, choosing a random source port to access the server on a fixed destination IP address and port number. The source and destination IP addresses and port numbers, as well as the protocol type, identify the flow, representing identification features, also known as a flow tuple. The packet and byte numbers in each direction are used as a metric for volume, representing volumetric features. In this paper, we use short labelling for the source and destination IP address with capital letters *S* and *D*, the source and destination ports with lowercase letters *s* and *d*, and the protocol with the letter *P*. The source and destination packet and byte counts are labelled as *sP*, *dP*, *sB* and *sB* respectively. More formally, we can introduce a set of identification features *I* and a set of volumetric features *V*:

$$I = \{S, D, P, s, d\} \tag{1}$$

$$V = \{sP, dP, sB, dB\} \tag{2}$$

Entropy calculation is based on data aggregation, which is the process of grouping flows based on the value of one or more flow features during a certain period, called epoch. For each distinct aggregated element, the so-called aggregation key, all related flows are counted into a flow number, labelled as *f*, while the volumetric features are summarized into total packets and bytes for both directions separately.

The flow identification features are the most meaningful to be used as the aggregation key. Having in mind that the protocol feature takes just a few distinct values, mostly TCP, UDP and ICMP, aggregation by this feature would not provide useful information. A set of aggregation features is therefore defined as follows:

$$\Phi = \{S, D, s, d\} \tag{3}$$

It should be noted that the aggregation can be done using more than one feature, altogether creating a complex aggregation key, or more formally, using features from any set of the power set of Φ, except an empty set. To annotate the complex aggregation key, we will use feature labels in the following order: *S*, *D*, *s*, and *d*, separated by the character '.'. Therefore, a total of 15 aggregation keys are available:

$$A = \{S, D, s, d, S.D, S.s, S.d, D.s, D.d, s.d, S.D.s, S.D.d, S.s.d, D.s.d, S.D.s.d\} \tag{4}$$

The straightforward aggregation will result in the distribution of flow count and the sum of source/destination packets/bytes of each distinct aggregated element. We will label these distributions using the feature label followed by the aggregation key in squared brackets. For instance, the distribution of the flow count feature (f), aggregated by all distinct pairs of source IP addresses (S) and destination ports (d) is labelled as f [S.d]. The flow count feature is a useful metric not only because the attacks generate a lot of malicious flows but it also influences normal traffic and increases its flow numbers due to exhaustion of the internal memory (flow cache) of the flow probes 31.

At this point, we will generalize the concept of the in-degree and out-degree features, used in 2632, which is defined by a total number of distinct source hosts per each destination host and a total number of distinct destination hosts per each source host, labelled as S[D] and D[S], respectively. Taking into consideration any other identifying features that are not used in the aggregation key, such as source and destination ports, we can additionally count the distinct occurrence of these features per aggregated element. Since they represent the communication behaviour of the main aggregated elements, we will call these additional features *behaviour* features. More formally, for

the set of aggregation features Φ and the set of aggregation keys K, a set of available behaviour features is:

$$B = \Phi \setminus \{K\} \tag{5}$$

Robust anomaly detection with a novel classification method proposed in this paper heavily utilizes behaviour features as the main source for network behaviour analysis along with the flow count feature. To briefly illustrate the usability of this approach, let us consider a DDoS amplification attack, where many source IP addresses send packets to a single destination host, all using the same source port, such as the UDP port number 53 used by DNS amplification attacks 33. This unusual network behaviour can be detected by counting the number of distinct source IP addresses (S) for each element aggregated by the destination IP address and the source port (D.s), labelled as S[D.s]. The same stands for a port scanning scenario, which can be captured by counting the occurrence of distinct destination port (d) in aggregation with the source and the destination IP address (S.D), i.e. d[S.D].

3.2. Entropy calculation

In anomaly detection techniques entropy is used to present the level of randomness in a data distribution. The changes in a data structure in a distribution obtained from the aggregation process will change the entropy value. If the entropy change is significant, it is considered as unusual behaviour in network communication or an anomaly, which often indicates security threats.

In many researchers the well-known Shannon entropy 13 is used, which is defined by the following equation:

$$H_S(X) = \sum_{i=1}^{N} p(x_i)\ \log_b \frac{1}{p(x_i)} \tag{6}$$

In the general case, N is a total number of elements in the distribution of feature values, while $p(x_i)$ is an empirical probability, calculated by the relative contribution of element x_i with value m_i in the total sum of all values, M:

$$p(x_i) = \frac{m_i}{M},\ M = \sum_{i=1}^{N} m_i \tag{7}$$

Rényi 27 and Tsallis 25 entropies involves an additional parameter α, where positive values put more weight on the highest values in the distribution (*peak*), while negative values favourite elements with low values in the distribution (*tail*):

$$H_R(X) = \frac{1}{1-\alpha}\log_b\left(\sum_{i=1}^{N} p(x_i)^{\alpha}\right) \tag{8}$$

$$H_T(X) = \frac{1}{1-\alpha}\left(\sum_{i=1}^{N} p(x_i)^{\alpha} - 1\right) \tag{9}$$

In this paper, we use a scaling factor to normalize the entropy to a value of 1 for fully randomized distribution. The scaling factor for Shannon and Rényi entropy is $1/\log_b N$ and for Tsallis entropy it is $(1-\alpha)/(N^{1-\alpha}-1)$. With such a scaling, the Shannon entropy always provides values between 0 and 1, as well as Rényi and Tsallis entropies

with positive parameter α, while the negative parameter α results in entropy values above 1.

3.3. Entropy changes detection

Over time, the aggregation and entropy calculation process generates many time series of entropy values for each feature. With normal network traffic, the entropy values are stable with minor deviations, while in the presence of an anomaly, some features are dramatically affected with significant entropy change (drop or increase). To detect these changes in the time series entropy values, a margin of accepted entropy deviation needs to be calculated first. A commonly used approach is based on the Exponential Moving Average (EMA) technique for short trend prediction 34 or taking maximum and minimum values from the sliding time window of some recent epochs. Both techniques can be used, but the rest of the presented research is based on the EMA prediction technique since it can be fined tuned to adapt more accurately and provides a baselining useful for data visualization and analysis.

With this approach, a predicted value in epoch *n*, denoted as $\hat{H}_n$, is calculated recursively, taking into account the previously predicted value $\hat{H}_{n-1}$ and the newly calculated entropy value H_{n-1} in epoch *n-1*:

$$\hat{H}_n = (1- \alpha_h)\, \hat{H}_{n-1} + \alpha_h H_{n-1} \tag{10}$$

The coefficient α_h represents the degree of weighting decrease, the so-called *smoothening factor*, which falls in the range between 0 and 1. A lower value for α_h gives a stronger influence of the previously predicted value $\hat{H}_{n-1}$, resulting in smoother baselining values, while at higher values for α_h the predicted values faster adopt and follow recent data H_{n-1} in the observed data sequence.

Some entropy time series can regularly vary their values more than the others. To identify significant entropy changes, we propose to analyse these relative variations in the context of the baselined standard deviation (*S*), using the same EMA approach as follows:

$$\hat{S}_n = (1- \alpha_s)\, \hat{S}_{n-1} + \alpha_s S_{n-1} \tag{11}$$

Finally, the range of acceptable entropy values considered as normal is defined by lower and upper thresholds, as measures of acceptable deviation from the baselined entropy value $\hat{H}_n$ as follows:

$$T_n = [\underline{T}_n , \overline{T}_n] \tag{12}$$

where

$$\underline{T}_n = \hat{H}_n - k_t \hat{S}_n \tag{13}$$

$$\overline{T}_n = \hat{H}_n + k_t \hat{S}_n \tag{14}$$

and k_t is the multiplication factor that makes the range wider, the so-called *threshold factor*. For any entropy value H_n that falls out of the threshold range T_n in epoch *n*, an alarm is triggered as an indication of an anomaly.

With proper tuning of parameters α_h, α_s and k_t, the above-mentioned technique efficiently detects significant changes in the observed time series values. We have empirically concluded that the optimal baselining trend is achieved by the following smoothing coefficient values of α_h=0.1 and α_s=0.05, while the threshold factor was set to k_t=4, which accurately captured the anomalies, while still eliminating most of the false positive alarms.

Some authors claim that parametrised Tsallis and Rényi entropy outperform the Shannon entropy in terms of the better detection of peaks or tails in the feature distributions 34. We believe that their conclusions are tightly related to the applied detection methods, data and features used in the experiments, and accordingly, this conclusion cannot be simply generalised. For that reason, in this paper, we analyse and compare the Shannon, Rényi, and Tsallis entropies from two main aspects: the ability to detect anomalies and sensitivity to deception. For Rényi and Tsallis entropies, we will use a fixed value of the parameter α (+2 and -2), which is shown to provide optimal performances 4.

To better understand the behaviour of each entropy type, we will consider a reciprocal distribution of 100 elements, given by the function 1/x, where the distribution starts with values 100, 50, 33, 25, and ends with *a long tail* of value 1. According to our experiments, this distribution roughly approximates a deviation of flow feature values in real network traffic, which is also reported in 3. Gradually increasing the peak of the distribution, from the value of 100 to 1000, the entropy is changed in the way presented in Fig. 1. The Shannon entropy, as well as parametrised entropies with the positive parameter α, results in decreased values, while the negative parameter α leads to an entropy increase. On the other hand, increasing the tail of the distribution up to 1000 new elements with value 1 involves more similarities in the data, and consequently, the entropies approach to value 1, which is shown in Fig. **2**. In all cases, the Rényi entropy with positive parameter gives the lowest entropy values, while Tsallis entropy gives much higher values (in a range from 1.7 to 106), which are not shown since they are out of the scale used in the chart. It is worth highlighting that the entropy with lower values leaves less space to detect a drop, especially when the standard deviation is higher. This is the case with the Rényi entropy with a positive parameter, which is more sensitive to the regular variation of data (highest slope in Fig. 1) and also provides the lowest values.

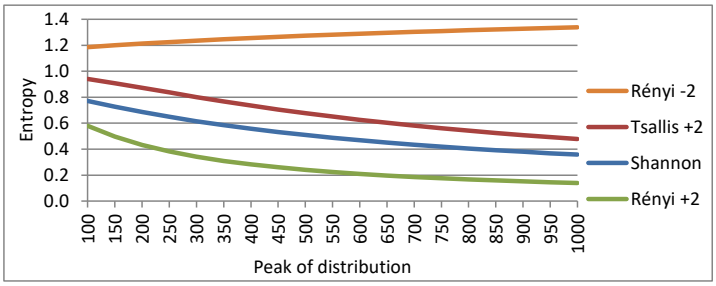


Fig. 1. The entropy change given by the increase of the distribution peak.

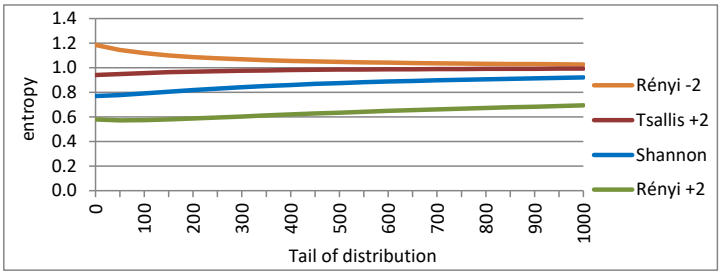


Fig. 2. The entropy change given by the increase of the distribution tail.

It should be highlighted that features with smaller standard deviation generally provide more distinguished changes, which gives better detection ability. Also, more randomized distribution and the entropy values near 1 generally leave more space for entropy drop and its detection. From the figure presented above, it should be concluded that the Rényi +2 entropy type gives the lowest values, and in case of higher standard deviation, there will be not enough space to detect changes.

3.4. Protection against entropy deception

In entropy-based approaches, anomalies are usually detected by features that generate a peak in the data distribution. This peak will make the entropy drop or increase with regards to entropy type and parameter α. Anyhow, the authors in 5 have shown that every entropy change caused by a peak in a distribution can be suppressed by adding more elements of the average value in the distribution to make data more even. The same effect can be also achieved using a value equal to 1 for each added element, but much more elements are needed in this case. With this method, attackers can camouflage the attack by generating spoofed traffic in parallel to the attack, and effectively deceive the entropy-based detection systems.

To provide a protection mechanism to this entropy deception, we analyzed the effect of entropy suppression on different entropy types, as well as on different features. The previously mentioned reciprocal distribution with peak values of 200, 500, and 1000, gives the average data values equal to 5, 9, and 13, respectively. The number of elements needed to suppress these peaks using these average values according to 5, as well as reference value equal to 1 for each entropy type, is given in Table 1. The Rényi entropy with positive parameter α (‘Rényi +2’) and Tsallis entropy with negative parameter α (‘Tsallis -2’) require the highest number of injected elements using the average value. However, this number is much higher for ‘Rényi +2’ entropy when adding elements at the end of distribution using a value equal to 1.

Table 1. The number of elements needed to deceive entropy.

Entropy type	Peak / average					
	200/5	200/1	500/9	500/1	1000/13	1000/1
Shannon	34	275	97	935	143	2135
Tsallis +2	53	280	130	1185	206	2750
Rényi +2	82	1135	207	5275	365	15200
Tsallis -2	38	45	265	166	694	348
Rényi -2	24	28	125	98	273	195

The results from Table 1 lead to the conclusion that the deception of one entropy type does not necessarily mean that the other entropy types are deceived too. This expectation is confirmed in Table 2 which shows the ratio of entropies before and after a deception in our base reciprocal distribution with a peak of 1000 and adding elements with average values 13. When nulling one entropy type (in rows), the other entropies (in columns) are below or above the initial values.

Table 2. Relative differences in deceiving different entropy types.

Entropy type	Peak=1000, average=13				
	Shannon	Tsallis +2	Rényi +2	Tsallis -2	Rényi -2
Shannon	0%	-4%	-27%	220%	3%
Tsallis +2	7%	0%	-17%	153%	2%
Rényi +2	16%	4%	0%	67%	-2%
Tsallis -2	22%	5%	18%	0%	-5%
Rényi -2	12%	2%	-8%	108%	0%

The entropy deception method proposed in 5 addresses only one feature distribution, while other features are not considered. Like the analysis of different entropy types, we can generally expect that different features are differently affected by spoofed traffic. This disbalance especially holds when injecting new elements in a behaviour feature distribution using average value, since the spoofed flows with aggregation attributes must be repeated using distinct values of behaviour feature. The easiest approach is to use full randomization of all attributes in the spoofed traffic, which would produce the elements with a value of 1 at the end of the feature distributions. However, this would significantly increase the number of distinct elements in a feature distribution, which is the case with the "Rényi +2" entropy in Table 1 with 15.200 new elements. It is also noteworthy that it is relatively easy to generate spoofed traffic to the targeted victim network, but this traffic will be highly asymmetric, mostly with no reply in opposite direction.

According to the previous analysis, we propose a protection method against entropy deception attempts, which relies on the detection of spoofed injected traffic that camouflage the attacks, based on the following principles:

- Prefer the entropy type which requires more injected elements to deceive the entropy (such as 'Rényi +2')
- Use the number of distinct elements in a feature distribution as a new detection metric, named as a *distribution length*. To the best of our knowledge, this metric has not been used in the scientific literature so far.

– Monitor the flow count of asymmetric traffic (traffic with no reply) as an indication of spoofed traffic.

The experimental results that validate the proposed protection method are presented in Section IV.

3.5. Multivariate analysis – a taxonomy of communication patterns

To identify the class of the anomaly as an indication of a particular type of security threats in addition to its detection, we propose a multivariate analysis of entropy values, which involves the observation and mutual analysis of many features. To better investigate the behaviour of different anomalies in terms of aggregation keys and the corresponding features, we have analysed the normal network behaviour and the communication characteristics of the most prominent network security attacks. Based on this analysis, we have defined flow-based taxonomy of communication patterns, which is further used for anomaly classification.

Security threats usually follow the client-server model, but the magnitude of some communication characteristics is much higher. DDoS amplification attacks utilize services such as DNS or NTP on servers that are not properly configured, the so-called *open servers* 33. The attacker sends a large number of small queries with a spoofed source IP address of the targeted host, and all servers reply to it, generating traffic of a much higher magnitude. In October and November 2016, two websites within the network of the University of Belgrade were attacked by NTP and DNS amplification attacks respectively. A single UDP source port number was used as a source of the attack (123 for NTP and 53 for DNS), but the destination port for the DNS attack was fixed to HTTP, while the NTP used a random destination port. In both cases, more than 1000 open servers generated up to 4Gbps traffic for 20 to 30 minutes, bringing down not only the attacked web servers but also disrupting other services due to the overload of the uplink of the entire national research and education network AMRES. The intensity of the attacks was easily detected and mitigated by the NetFlow Analyser tool using volumetric statistics only (bytes, packets, and flows) 35. However, to detect less intensive attacks that may remain under the radar, the communication pattern with other features must be analysed.

On the other hand, many security threats start much earlier, before real damage is caused. Network scan is looking for an open service on the network, generating flows from a single source IP address and usually an arbitrary source port toward a fixed destination port on many hosts over an enterprise network 36. Port scan is a method for determining which ports on the single host are open, producing many flows with a different destination port and a fixed destination IP address 36.

Once a host is located with the open TCP port requiring authentication, such as port 22 for SSH or 3389 for Microsoft Remote Desktop, the attacker can perform brute-force password-guessing activities, trying commonly used phrases by a dictionary attack 37. The footprint of this traffic structure is characterized by too many short flows with one or two packets transferred between two fixed IP addresses, using multiple source ports and a single destination port.

All the above-mentioned network behaviours have a very specific communication pattern marked by single or multiple sources and destination IP addresses and port numbers involved. These characteristics can be simply described using label ‘1’ for

single or ‘N’ for multiple occurrences of identification features in the order from the source IP address (S) and source port (s) to the destination IP address (D) and destination port (d), in form of ‘Ss-Dd’. In this way, previously analysed anomalies can be categorized as follows:

- DNS amplification DDoS N1-11
- NTP amplification DDoS N1-1N
- Port scan 1N-1N
- Network scan, worm propagation 1N-N1
- Dictionary attack 1N-11

This classification and labelling can be further generalized to cover all 16 permutations of labels of ‘1’ and ‘N’ for source and destination IP addresses and ports. With this generalized approach, network scan with a fixed source port is related to the communication pattern of class 11-N1, while a distributed SYN flood attack falls into the class NN-11 since the attack is performed from many source IP addresses and port numbers to a single destination IP address and TCP port number.

Regular network traffic can be also described with the introduced labelling of the communication patterns. A client can initiate many connections to a certain server, which falls into the 1N-11 class, while public servers, which are used by many clients, fall into the class NN-11. Additionally, DNS, SMTP, and HTTP proxy services follow the 1N-N1 pattern, acting as a client establishing communications with many other servers.

Along with the protection against entropy deception, another main goal in this research has been to extend the existing entropy-based anomaly detection approaches with a classification method using a multivariate analysis of different flow count and behaviour features, which is easy to implement in real-life networks. It is achieved by identifying a unique signature of the anomalous behaviour by analysing entropy changes of many observed features and developing rules for their classification. These classification rules are developed based on the analysis of the experimental results, and therefore they are analysed and presented in the next section, along with the validation of our findings.

3.6. System complexity

Calculating entropy values, with EMA prediction and standard deviations, is not a complex process once the distributions are generated by the aggregation process. However, many aggregation tasks need to process a great number of flow records, which are both CPU- and memory-intensive processes. Each flow record needs to be matched with all aggregation keys separately, counting or summarizing the corresponding values of the remaining features. Additionally, calculating behaviour features requires second-degree aggregation to count all distinct data-point occurrences.

The complexity of the algorithm highly depends on its implementation. The most efficient solution is achieved by using an unordered associative array (hash-map in Java), which in most cases has O(1) complexity in time, while the worst-case complexity is O(log n) when using balanced search trees, which are created only for a small number of entries sharing the same hash-map key. However, the complexity in the memory space is O(n) in all cases. With such implementation in Java programming

language, processing a dataset of one million flow records on a desktop computer and generating a total of 208 data distributions for all possible aggregation keys and feature combinations (including volumetric features), we have achieved a high processing speed of 30.000 flows per second consuming a total of 8 GB of RAM.

The root cause analysis requires keeping raw flow records data for at least one (previous) epoch, while a long history is always beneficial depending on the available storage space. An efficient method can be achieved by using the compressed text of binary files, while the more flexible solution for practical usage can be based on a NoSQL database, such as Elasticsearch.

Another concern relating to the real-time aggregation process and data storage is a high rate of incoming flows, such as tens of thousands per second. A solution to this is to use a flow sampling technique, processing only a statistical fraction of the flow data stream while rejecting the rest. Some information will be lost in that case, but a sufficient amount of data (up to the processing limit) is taken into account, resulting in a fairly good statistical approximation.

4. Evaluation

4.1. Datasets used

To validate the proposed approach we have chosen two labelled datasets, namely the CICIDS2017 dataset 38 and the CTU-13 dataset 39, each consisting of several flow data traces taken from real network communications and a controlled laboratory environment.

The CICIDS2017 dataset is one of the latest and most complete publicly available flow-based labelled datasets. It includes the most common attack scenarios, covering the profiles of Web-based, Brute force, DoS, DDoS, Infiltration, Heartbleed, Bot, and Scan attacks, each in a different file named according to the weekday when the dataset was created. A total of 80 flow-based features related to network communications were generated by processing real traffic with simulated attacks.

The CTU-13 dataset consists of 13 independent parts, each with internally controlled legitimate traffic, traffic generated by a real botnet network, and a large portion of the so-called background traffic, taken from the Czech Technical University network, in which minor ‘noise’ anomalies were intentionally retained. We have used the CTU-13 dataset and model communication patterns using synthetically generated anomalies to analyse their effect on each feature. Then the classification rules are validated by investigating minor ‘noise’ anomalies in the real background traffic from another trace in the same dataset.

Since our research is based on pure flow data that can be easily collected from the routers and used in real-life network environments, both datasets were slightly modified. At first, only the basic flow features were kept. Then, following the usual practice of flow configuration on network devices to avoid burst traffic load, long-lasting flows were proportionally fragmented into short equivalent flows, with the maximum duration of 60 seconds, which was set as the default epoch period.

4.2. Validation of the protection against entropy deception

Validation of the protection method against entropy deception is demonstrated on the CICIDS2017 dataset, using the trace named 'Friday afternoon'. It contains a PortScan attack when an attacker is trying to establish connections to many destination ports on a remote victim system to find vulnerabilities. In this case, a single source port is used during this process, which is described by the 11-1N communication pattern. Using the source and destination IP addresses as the aggregation key, the destination port behaviour feature d[S.D] generates a quite random distribution with the entropy value close to value 1 and a small standard deviation for all entropy types, shown in Fig. **3**. However, during the attack, in three series from epoch 112 till epoch 145, significant entropy drops are noticeable for all entropy types. The margin of acceptable deviation calculated by the EMA technique is too narrow and not shown.

We will demonstrate a deception mechanism on the second attack only, which occurs from epoch 129 till 131 with an average value in data distribution equal to 4, while the first and last attacks are left unchanged for comparison purpose. To deceive the Shannon entropy, an entropy value of 0.84 during the attack needs to be increased above the threshold value of 0.98, which requires a total of 3,000 new elements with an average value of 4. This is achieved by generating 3,000 series of 4 synthetic flows, where those 4 flows have unique source and destination IP addresses and distinct destination port number. For that reason, a total of 12,000 synthetic flows was generated and added to the dataset to deceive the Shannon entropy during this attack. Fig. 4 demonstrates that the Shannon, Rényi -2, and Tsallis +2 are successfully deceived, while Rényi +2 and Tsallis -2 are also affected, but still not sufficient to avoid detection. To camouflage the Rényi +2 entropy in this case, a total of 22,500 series of 4 synthetic flows need to be generated, which requires a total of 90,000 new spoofed flows during each of these epochs.

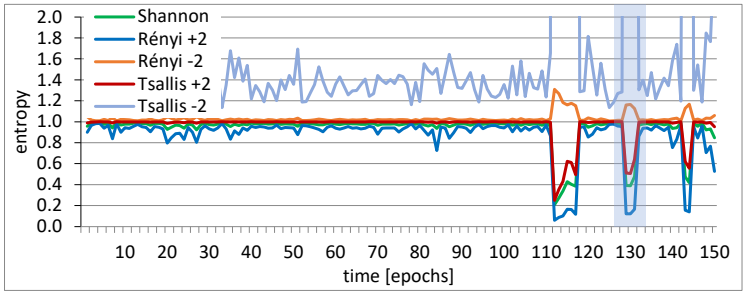


Fig. 3. The entropy of the d[S.D] feature – the original dataset.

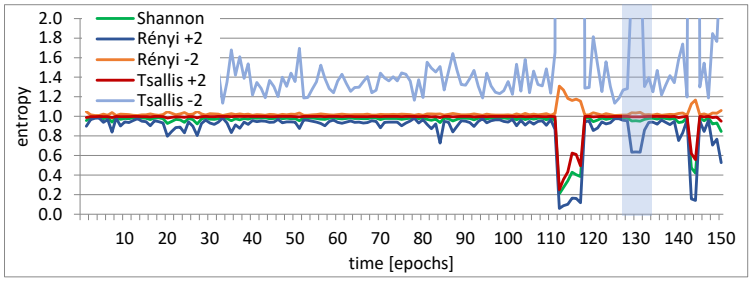


Fig. 4. The entropy of the d[S.D] feature – deceiving the Shannon entropy in epochs 129-131.

In parallel to entropy calculation, as a control mechanism to detect this deception attempt, we propose to monitor a total number of elements in feature distributions. Fig. 5 presents a total number of elements that need to be added to the d[S.D] feature distribution to deceive all entropy types. It is obvious that the injected traffic significantly exceeds the regular values and the, especially for Rényi +2 entropy type, which is far above the presented scale. The threshold can be based on a fixed value or dynamically applied using the EMA technique. It noteworthy that this metric is not affected by the other two anomalies which are not deceived.

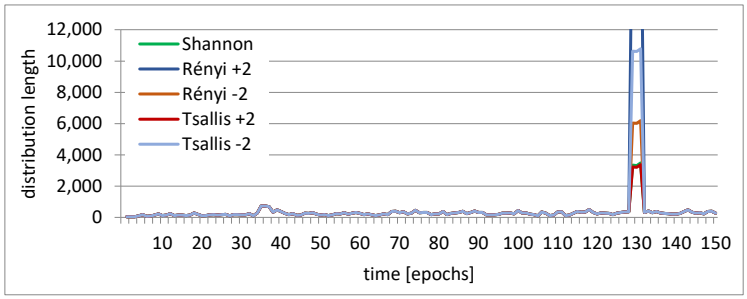


Fig. 5. The length of the d[S.D] feature distribution with spoofed traffic.

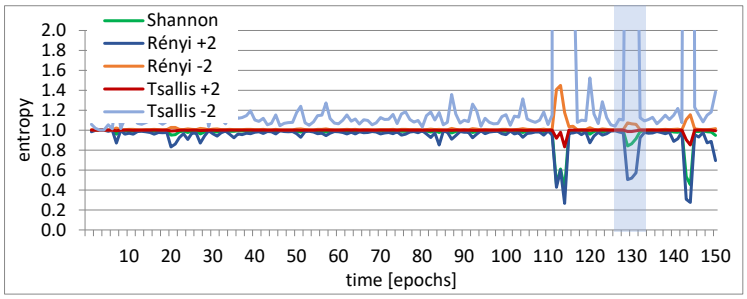


Fig. 6. The entropy of the d[S.s] feature, deceiving the feature d[S.D].

Even such a huge number of added elements for deceiving the d[S.D] feature is not enough to completely deceive the d[S.s] feature for all entropy types, which is shown in Fig. 6. To deceive the d[S.s] feature many more elements need to be injected, which is even easier to detect with the proposed metric.

4.3. Anomaly modelling

Entropy-based network traffic anomalies detection is efficient if the anomalous traffic is intensive enough to cause a significant spike in data distribution and change the entropy values for the observed feature. However, our attention is attracted by the fact that different types of anomalies leave different footprints in the entropy of the corresponding features. To better analyse this behaviour and provide a key instrument for the anomaly classification based on the proposed multivariate analysis, we have modelled characteristic anomalies for each of the 16 communication pattern classes (from 11-11 to NN-NN).

For each anomaly model, we have generated a modified dataset, combining flows of normal network traffic with the synthetically generated flows representing modelled anomalous behaviour. Normal traffic was extracted from the CTU-13 dataset, the trace named '51', with around one million flow records collected during four hours. We additionally removed smaller 'noise' anomalies and obtained a stable traffic structure with no significant deviations over time. This traffic is not used to test anomaly detection accuracy but rather as ground truth for the analysis of which features are affected by different anomalies, even those of small intensity.

Anomalies have been modelled by synthetic traffic using a flow generator software, developed by Bereziński 4 and slightly modified following our dataset format. Starting very modestly with only 25 anomalous flows per epoch, the intensity gradually increased producing a total of 50, 100, 200, 500 and 5000 flows per epoch. Small random variations were involved to present a stochastic traffic nature more realistically. It should be mentioned that the last anomaly burst was extremely huge to check whether the entropy of some features was completely immune to the anomaly. Moreover, this burst was repeated twice. The first traffic burst had 5000 purely random and mostly unique values of the aggregation feature (labelled in the model with 'N'). The second burst had the same amount of flows, but containing 10 times fewer distinct elements, each of them repeated 10 times on average. With this method, having a DDOS attack as an example described by the N1-1N model, the source port and destination IP address were fixed in the corresponding synthetic flows, while the source IP address and destination port numbers were randomized. The generated synthetic flows for each modelled anomaly class were injected separately into the dataset with the normal traffic, starting from epoch 80 in short series of three epochs, increasing the intensity every 20 epoch.

4.4. The entropy of anomaly models

The experiments were conducted for each of the 16 anomaly models separately, starting from 11-11 up to NN-NN, aggregating by all aggregation keys defined by Equation 4.

Calculating the total flow count and all behaviour features, a total of 103 feature distributions were generated for each model. As the result, a total of 1648 series of each entropy type were calculated. Volumetric features, such as the source and destination byte and packet counts, have not been used since they are efficient only for DDoS and similar volume-intensive attacks, but useless for many other attacks, such as Port Scan, Network Scan or Dictionary attack.

It is already highlighted that the entropy is changed due to a spike or *a long tail* in feature distribution. Having as an example the N1-1N model, which relates to DDoS NTP amplification attacks 35, both the destination IP address and the source port number are unique during the attack and they are good candidates for the aggregation key to capturing a spike in distribution. On the other hand, the source IP address and the destination port, which relates to the label 'N' in the N1-1N model, can be used in the aggregation key to detecting *a long tail* of the distribution. Using the Shannon entropy these two characteristic cases are demonstrated in Fig. 7 and Fig. 8 for the feature f[s] and S[d] respectively.

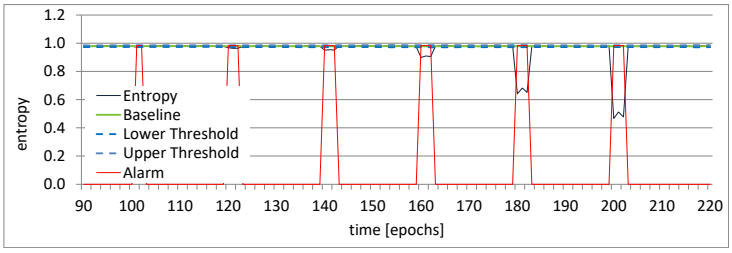


Fig. 7. The Shannon entropy and the N1-1N model - the flow count feature aggregated by the source port (f[s]).

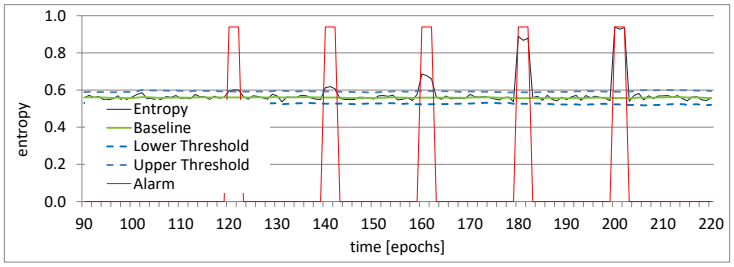


Fig. 8. The Shannon entropy and the N1-1N model - the source IP address feature aggregated by the destination port (S[d]).

Due to the natural randomness of the source port number in network communications, which is used in the aggregation key in the feature f[s], the entropy of regular traffic reaches its maximum value of 1, with a small standard deviation. With DDoS NTP amplification traffic, distinct pair of the victim IP address and the source port number used in many flows of the attack (UDP port number 123) makes a significant spike in the flow count distribution, resulting, in turn, in significantly decreased entropy. The opposite stands for the feature S[d] - the entropy values of

regular traffic are lower (around 0.55) and using a highly randomized destination port number as an aggregation key produces many elements with only one occurrence in the feature distribution.

The high sensitivity of the feature f[s] on the observed anomaly, demonstrated in Fig. 7, is used to further validate the feasibility of the protection method against entropy deception on low-rate attacks. The increase of the distribution length with the spoofed traffic to deceive all entropy types of the f[s] feature, presented in Fig. 9, can be easily detected. Only a deception of the smallest and barely noticeable anomaly around epoch 100 (with 25 synthetic flows only) can not be detected due to a high number of data elements in the distribution of the feature f[s] since the source port is highly randomized in regular network communications.

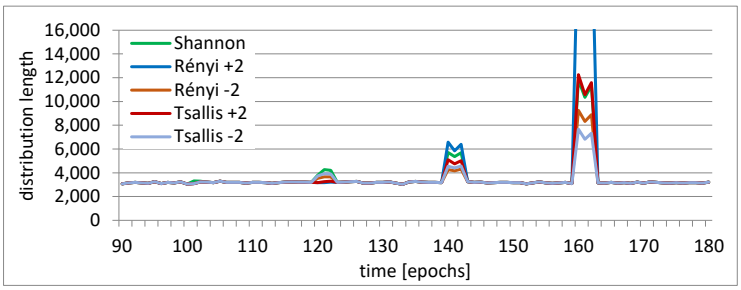


Fig. 9. Protection against entropy deception - the length of the f[s] feature distribution with spoofed traffic applied on the N1-1N model.

An extensive analysis of all entropy types has been done for all anomaly models and complete behaviour features set. Even though that some authors reported better detection ability of the Rényi and Tsallis entropy over Shannon entropy, our experiments confirmed that this is not the general rule. For the previously introduced base ground dataset with synthetically generated 7 anomaly series and the N1-1N anomaly model as an example, Table 3 presents the number of anomalies detected by different entropy types for the most characteristic features. The Shannon entropy outperforms some other entropy types for the feature S[s], there are other cases and features when other entropy types perform better.

In all cases, the difference in detection ability is related only to the smallest anomalies, while all entropy types successfully detected all anomalies of modest and especially high intensity. For that reason, we believe that the right selection of the entropy type is not a straightforward task and should consider many aspects, such as specific network traffic and its variety and deviations, a technique used for entropy change detection, as well as previously analysed resilient to deception.

Table 3. Number of detected anomalies in N11N model by different entropy types

N11N	Shannon	Renyi +2	Renyi -2	Tsallis +2	Tsallis -2
f[S]	3	2	4	3	3
S[D]	3	2	4	2	4
d[D]	6	6	6	6	6
S[s]	6	6	5	5	6
d[s]	5	4	5	4	6
S[d]	5	3	4	3	4
f[S.d]	3	2	3	3	3
S[D.s]	6	6	5	5	6

The thorough analysis of all experimental results is summarized in Table 4, which describes entropy changes for flow count and behaviour features using each aggregation key (shown in rows) and each anomaly model (shown in columns). The label 'X' in the table denotes the entropy change caused by a peak in the feature distribution, while the label 'o' denotes the entropy changes due to a tail of the distribution. The results are equal for all entropy types which confirms already demonstrated similar detection ability of all entropy types. Therefore, in the rest of the paper, we will consider the results of the Shannon only, where the labels 'X' and 'o' related to entropy drop and increase respectively. The length of the labels expresses the detection efficiency of the observed features, where more characters in the label reflect a higher efficiency, while only one character indicates a low detection ability useful only in case of extremely intensive anomalies. More precisely, one character is used when the feature is affected only by two of the most intensive synthetic anomalies in our reference dataset (the last two anomalies in Fig. 7), two characters for anomalies of moderate intensity, while the label with three characters is used for the most sensitive features able to detect even the low-rate anomalies (two left most anomalies in Fig. 7). For example, the previously mentioned features f[s] and d[S] for the N1-1N model can detect most of the generated anomalies and, therefore, they can be considered as very sensitive and are labelled with 'XXX' and 'ooo' respectively.

Even a brief look at the table reveals that the entropies of different features behave differently for different anomaly models, while some of them are not affected by a particular anomaly at all (the empty cells in the table). More importantly, how the entropies are affected by the modelled anomalies follows a very specific periodic pattern. It can be observed that the entropy drop (marked with 'X') occurs only when all identification features in the aggregation key have a single occurrence in the anomaly model (marked with '1' in the anomaly model label). In this case, entropy is always affected for the flow count feature (such as f[S.s] in the first four columns), while the behaviour features are affected only when it corresponds to mark 'N' in the anomaly model label (such as D[S.s] in 11-N1 and 11-NN models, and d[S.s] in 11-1N and 11-NN models). An increase in entropy values (marked with 'o') occurs when at least one identification feature in the aggregation key has many occurrences in the anomaly model (marked with 'N' in the anomaly model label) since this element will produce many new elements in the distribution tail. It should be noted that when the source port feature is used in the aggregation key, it can result only in an entropy drop, and not in an increase. The reason for this lies in the behaviour of the regular network, where a source host as a client initiates connections using a random source port number so that the corresponding distribution is already randomized.

Table 4. Entropy changes of the flow count and behaviour features affected by the anomaly models.

Feature		Anomaly model															
Behaviour	Flow count	11-11	11-1N	11-N1	11-NN	1N-11	1N-1N	1N-N1	1N-NN	N1-11	N1-1N	N1-N1	N1-NN	NN-11	NN-1N	NN-N1	NN-NN
D[S]				XX	XX			XX	XX	oo	oo	oo	oo	oo	oo	oo	oo
s[S]						X	X	X	X	oo	oo	oo	oo	oo	oo	oo	oo
d[S]			XXX		XXX		XXX		XXX	o	o	o	o	o	o	o	o
	f[S]	XX	XX	XX	XX	XX	XX	XX	XX	oo	oo	oo	oo	oo	oo	oo	oo
S[D]				oo	oo			oo	oo	XX	XX	oo	oo	XX	XX	oo	oo
s[D]				oo	oo	X	X	oo	oo			oo	oo	X	X	oo	oo
d[D]			XXX				XXX				XXX				XXX		
	f[D]	X	X	oo	oo	X	X	oo	oo	X	X	oo	oo	X	X	oo	oo
S[s]										XXX	XXX	XXX	XXX				
D[s]				XXX	XXX							XXX	XXX				
d[s]			XXX		XXX						XXX		XXX				
	f[s]	XXX	XXX	XXX	XXX					XXX	XXX	XXX	XXX				
S[d]			oo		oo		ooo		oo	X	ooo	X	oo	X	oo	X	oo
D[d]			ooo	XX	oo		ooo	XX	oo		ooo	XX	oo		ooo	XX	oo
s[d]			ooo		oo	X	oo	X	oo		ooo		oo	X	ooo	X	oo
	f[d]	X	ooo	X	oo	X	oo	X	oo	X	ooo	X	oo	X	oo	X	oo
s[S.D]				oo	oo	X	X	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo
d[S.D]			XXX				XXX										
	f[S.D]	XX	XX	oo	oo	XX	XX	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo
D[S.s]				XXX	XXX												
d[S.s]			XXX		XXX												
	f[S.s]	XXX	XXX	XXX	XXX												
D[S.d]			o	XX	o		o	XX	o	o	o	o	o	o	o	o	o
s[S.d]			oo		oo	X	oo	X	oo	oo	oo	oo	oo	oo	oo	oo	oo
	f[S.d]	XX	oo	XX	oo	XX	oo	XX	oo	oo	oo	oo	oo	oo	oo	oo	oo
S[D.s]										XXX	XXX						
d[D.s]			XXX								XXX						
	f[D.s]	XXX	XXX							XXX	XXX						
S[D.d]			oo	oo	oo		oo	oo	oo	X	oo	oo	oo	X	oo	oo	oo
s[D.d]			oo	oo	oo	X	oo	oo	oo		oo	oo	oo	X	oo	oo	oo
	f[D.d]	X	oo	oo	oo	X	oo	oo	oo	X	oo	oo	oo	X	oo	oo	oo
S[s.d]										XXX		XXX					
D[s.d]				XXX								XXX					
	f[s.d]	XXX		XXX						XXX		XXX					
d[S.D.s]			XXX														
	f[S.D.s]	XXX	XXX														
s[S.D.d]			oo	oo	o	X	o	o	oo	oo	o	oo	oo	o	oo	oo	oo
	f[S.D.d]	X	oo	oo	oo	X	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo
D[S.s.d]				XXX													
	f[S.s.d]	XXX		XXX													
S[D.s.d]										XXX							
	f[D.s.d]	XXX								XXX							
	f[S.D.d.s]	XXX															

The described behaviour, i.e. how the entropy of different features is affected by different anomaly models, is the key reason for the clear periodic pattern noticeable from the table. This is also the explanation of why some features are very effective in detecting some anomalies while being completely useless for others. This finding is consistent with the previous research 32628 but covers the full feature set and all the anomaly models. Moreover, it reveals that behaviour features or complex aggregation key for some anomaly models outperform commonly used flow count feature of basic

flow attributes. For instance, the authors in 28 classify a DDoS attack by detecting entropy decrease in the flow count of destination IP address and port number. This corresponds to our NN-11 anomaly model and features f[D] and f[d], which are sensitive only to the most intensive anomalies, while behaviour feature S[D] can detect less intensive anomalies (up to 10 times in our experiment). The difference in the metric performances is more obvious in the case of port scan attack, defined by the 11-1N model (using fixed source port) and the 1N-1N model (using random source port). In both cases the best performance is achieved using behaviour features d[S], d[D] and d[S.D], marked with 'XXX' label in Table 4.

This periodic pattern in the feature sensitivity to different anomalies leads to an important conclusion that each anomaly model has a unique footprint of triggered entropies, which is used in the development of the classification rules, as explained further in the text.

4.5. Classification rules

The aim of the proposed multivariate analysis is, firstly, to select the right features to ensure the most efficient detection of anomalies, and secondly, to accurately classify a detected anomaly to identify more precisely a potential security threat. Several methods for feature selection, including feature correlation, are proposed in the literature 232. In our approach, Table 4 reveals which feature is the most appropriate for which anomaly type. More importantly, a unique pattern of how the features are triggered by different anomalies can be recognised and used for defining the rules that classify an anomaly into an appropriate model.

Due to feature correlation, it is possible to minimize the set of features while keeping the ability in anomaly detection and classification. This could be done in several different ways so that the following principles are used to select the optimal rules for anomaly recognition and classification, based on the results from Table 4:

- Prefer the most efficient features ('XXX' or 'ooo').
- Prefer features affected by the minimal number of models.
- Prefer features with a simpler aggregation key.
- Use the 'Not affected' rule to differentiate feature behaviour from another model (empty cells in the table).
- Use the 'Not decrease/increase' rule to differentiate feature behaviour from another model (make a difference between 'X' and 'o' cells in the table)
- Select model identified by the smallest number of unique features first, then proceed with others.

By applying these principles to the results from Table 4, the following classification rules are jointly defined in the columns of Table 5, where all conditions must be satisfied to match the anomaly model given by the rows.

Table 5. Anomaly models identification and classification rules.

	Anomaly model	Affected (decrease)	Affected (increase)	Not affected	Not decreased
1	**11-11**	f[D.s]		S[D.s], d[D.s]	
2	**11-1N**	f[s], d[S.D]			
3	**11-N1**	D[s.d]		S[s.d]	
4	**11-NN**	d[s], D[S.s]			
5	**1N-11**	f[S.D]		d[S.D]	
6	**1N-1N**	d[S.D]		f[s]	
7	**1N-N1**	D[S.d]		D[s.d]	
8	**1N-NN**	D[S], d[S]		d[s]	
9	**N1-11**	S[D.s]		d[D.s]	
10	**N1-1N**	S[D.s], d[D.s]			
11	**N1-N1**	S[s.d], D[s.d]			
12	**N1-NN**	S[s], D[s], d[s]			
13	**NN-11**	s[D]		d[D]	d[S]
14	**NN-1N**	d[D]		S[s]	d[S]
15	**NN-N1**	S[d], D[d]		S[s]	
16	**NN-NN**		d[S], S[d]	S[s]	

For example, the 1N-N1 model, related to the horizontal port scan attack using many source port numbers, is affected by the feature D[S.d] but not by the feature D[s.d], which primarily identifies the similar 11-N1 model.

Defined classification rules include a minimal set of features, which is important for performance optimization since aggregation is a CPU and memory consuming process. However, in an anomaly detection process, it is useful to keep more features, even if they are correlated and redundant, to minimize false alarms.

4.6. Classification rules validation

The classification ability and the usefulness of the methodology presented in this paper are demonstrated on real network data taken from the dataset CTU-13. More precisely, data trace named ‘43’ was used, where intensive botnet traffic was excluded from the dataset, keeping a large portion of real-life background traffic with several anomalies of smaller intensity. Using only the flow count and behaviour features, the most characteristic results are presented below.

The entropy of the flow count feature aggregated by the source IP addresses (f[S]), shown in Fig. 10, reveals several smaller anomalies, including some minor deviations which generate false positive alarms.

The entropy of other features, such as destination port behaviour feature with the same aggregation key, namely d[S], illustrated in Fig. 11, shows that a part of the anomalies has disappeared, indicating the presence of different anomaly types in the traffic over time.

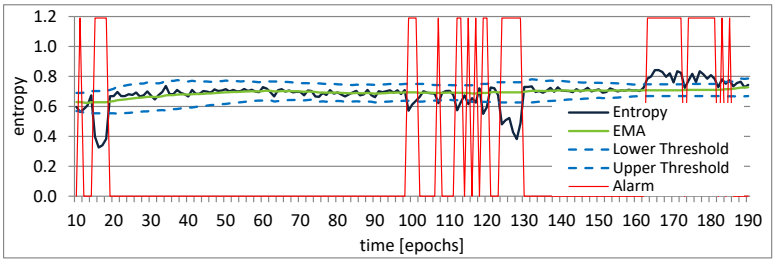


Fig. 10. The CTU-13 dataset, trace 43, regular traffic, feature f[S].

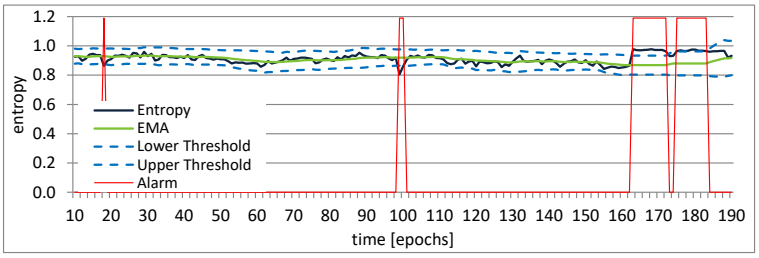


Fig. 11. The CTU-13 dataset, trace 43, regular traffic, feature d[S].

On the other hand, the entropy applied to the partition of the traffic, filtered by the protocol field, reveals new anomalies in different epochs. For TCP traffic only, the entropy of the destination port behaviour feature aggregated by the source IP addresses is shown in Fig. 12, while the entropy for the ICMP traffic using only the flow count feature aggregated by the source and the destination IP addresses are shown in Fig. 13. These less intensive anomalies were masked by the total traffic, but taking only a smaller portion of the traffic into account, the entropy changes become obvious and relevant. A small entropy deviation around epochs 100 in the total traffic had been barely noticeable in Fig. 11 and subject to suspicion as a false positive alarm until it was analysed for TCP traffic only (Fig. 12). These cases clearly demonstrate that data filtering into smaller parts is a simple method to achieve better detection sensitivity and efficiency.

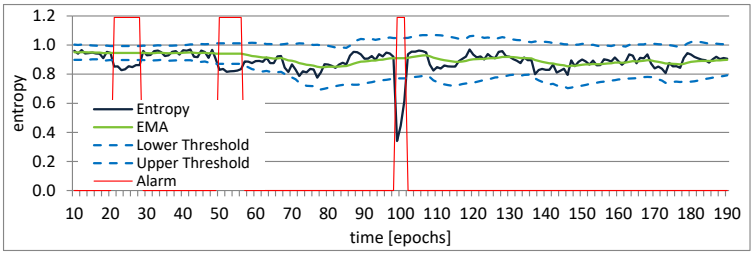


Fig. 12. The CTU-13 dataset, trace 43, TCP traffic only, feature d[S].

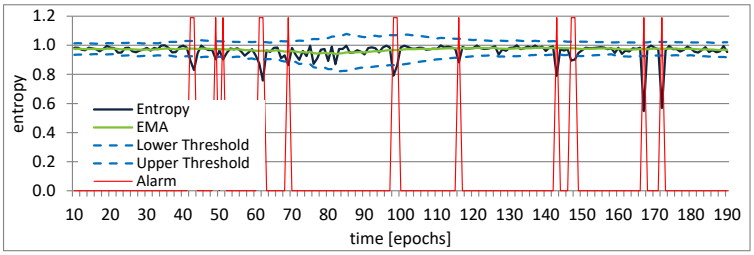


Fig. 13. The CTU-13 dataset, trace 43, ICMP traffic only, feature f[S.D].

The results of entropy analysis of the CTU-13 data trace named '43' using the proposed classification rules in different epochs for the most severe anomalies are presented in Table 6.

Table 6. Verification of anomaly classification rules using real network traffic.

Traffic	All	All	All	TCP	TCP	TCP	TCP	ICMP	ICMP
Epohs	15-19	124-130	163-180	21-29	51-57	99-102	170-171	167-168	172-173
Anomaly model	1N-11	1N-11	NN-11	1N-1N	1N-11 (1N-1N)	1N-1N	11N1	1N-11	1N-1N
D[S]							XXX		
s[S]	XXX	XX	o		XXX	X			
d[S]			o	XX	XX	XXX			XXX
f[S]	XXX	XXX	o		XXX	XX			
S[D]			XX						
s[D]	X	X	X	X	XXX	XXX			
d[D]				XXX	XXX	XXX			XXX
f[D]	XXX	X	X	X	XXX	XXX		XX	XXX
f[s]							XXX		
S[d]			XX			oo			
D[d]						oo	XX		
s[S.D]	XXX	XX			XXX	XXX			
d[S.D]				XXX	XXX	XXX			XXX
f[S.D]	XXX	XXX		X	XXX	XXX		XXX	XXX
D[S.s]							XXX		
D[S.d]							XXX		
s[S.d]	XXX	XXX			XXX				
f[S.d]	XXX	XXX			XXX		X	XXX	
S[D.d]			XX						
S[s.d]									
D[s.d]							XXX		

All detected anomalies follow the unique signature presented in Table 4 and can be properly classified by the developed rules. Only the TCP anomaly in epochs 51–57 presents a combination of two similar anomaly models: 1N-1N and 1N-11. A drill-down analysis of raw data has confirmed that the anomaly consists of flows with a larger

number of distinct destination ports according to the 1N-1N model, while one of them occurs more frequently, following the 1N-11 model. This is the case that demonstrates the feasibility of the proposed method to identify multi-vector attacks, combining two or more anomaly models. Raw data forensic is still needed for root cause analysis to mitigate the attack and proactively protect the victim.

4.7. Comparison with machine learning approaches

Entropy-based network traffic anomaly detection in many aspects completely differs from the machine learning methods, which makes it difficult and even impossible to directly compare their performances. For that reason, we rather discuss their general characteristics and leave a decision on which one is better for specific use-cases.

The main difference lays in the fact that entropy-based detection operates on the time interval level, detecting anomalies in epochs, while machine learning detection methods provide detection granularity on the data level, classifying each data point as normal or anomalous. This fundamental difference implies the following consequences:

- Anomalies detected using the entropy-based approach require further root-cause analysis to extract the information about the attackers, victims and services used.
- The entropy-based approach does not require training with a labelled dataset, as opposed to supervised machine learning, which makes it attractive for general purpose application in real-life networks with any kind of traffic unknown in advance.
- The entropy-based approach requires less processing power than most of the other techniques, which makes it attractive for real-time application.
- Performance metrics used in machine learning (Accuracy, Precision, Recall, ROC curve etc.) take into account individual labelled data and, therefore, they are inconvenient for application in the entropy-based approach.

As previously stated, the motivation behind our research has been to extend the anomaly detection technique with the classification method, for practical use in a general network environment. The entropy-based approach was chosen having in mind the above-mentioned characteristics. Anomaly detection is based on the data obtained by NetFlow or similar protocols, which are industry standards and the most convenient way to collect information about the network traffic structure. Flow data collected from network routers provide only basic information about communication peers (IP addresses, protocol, and port numbers), duration and total bytes and packets transferred. Enriched with flow count and behaviour features obtained in the aggregation process, this basic information appears to be sufficient for entropy calculation. Our experimental results confirm that this approach is efficient when the traffic structure is significantly changed during the attack, while it is useless for other attacks whose communication characteristics cannot be distinguished from regular traffic. In this work, we have solved the entropy deception problem, as a main weakness of the existing entropy detection methods.

On the other hand, machine learning approaches to network behaviour analysis rely on other communication details, such as TCP flags and window size, packets length, packet inter arrival time, jitters and their statistical parameters (average, min, max, standard deviation). Obtaining these data is based on processing raw traffic on the

packet level, which requires direct access to network traffic and demanding data processing, especially for real-time application.

The CICIDS2017 dataset was generated in this way and the authors originally used it for anomaly detection based on supervised machine learning 38. For each simulated attack, they achieved very high detection performances using various features and the following metrics: Precision (Pr - the ratio of the correctly detected attacks to all triggered alarms), Recall (Rc - the ratio of the correctly detected attacks to all attacks), and F-Measure (F1 - the harmonic mean of the Precision and Recall).

We have reproduced their experiment with the same dataset named "Thursday morning", which consists of Brute Force, Cross Site Scripting (XSS) and SQL Injection web attacks. We have also used Random Forest (RF), Multilayer Perceptron (MLP), and Naive-Bayes (NB) machine learning algorithms, and the same features used by the authors in 38, namely the initial TCP window size in both directions and the total bytes transferred from the source to destination.

In our reproduced experiments in the Weka software, using 70% of training and 30% of testing data randomly chosen from the dataset, we have generally confirmed their results, especially in terms of the Recall performance metrics.

Furthermore, we have performed a deeper investigation of raw data, which has revealed that most attack flows used the initial TCP window size of 29,200 and 28,960 bytes from the source and destination directions, respectively. Since the TCP window can take an arbitrary value even in attack communications, we wanted to check the detection capability of the machine learning algorithms when these values were changed. For this reason, we manually increased the initial TCP windows of attack flows in the testing dataset by 3%, 10% and 30% and repeated the experiments. From Table 7, which summarises the results, it is obvious that the Random Forest algorithm dramatically lost the detection capability even with small changes of 3%, while the Multilayer Perceptron algorithm was not able to detect any attack at all. Only the Naive-Bayes algorithm was more resilient to the initial TCP window value changes, but its performance was the lowest.

Table 7. Supervised machine learning performance evaluation

Alg.	Dataset	Precision	Recall	F1
RF	**Original**	**0.850**	**0.981**	**0.911**
	Modified, 3%	0.176	0.037	0.061
	Modified, 10%	0.176	0.037	0.061
	Modified, 30%	0.176	0.037	0.061
MLP	**Original**	**0.771**	**0.840**	**0.804**
	Modified, 3%	0.000	0.000	N/A
	Modified, 10%	0.000	0.000	N/A
	Modified, 30%	0.000	0.000	N/A
NB	**Original**	**0.132**	**0.909**	**0.230**
	Modified, 3%	0.132	0.908	0.230
	Modified, 10%	0.123	0.842	0.215
	Modified, 30%	0.123	0.842	0.215

The above example demonstrates that some machine learning algorithms, which are based on such specific feature values, can be easily deceived with just a small variation in the attack scenario. Rather than just presenting a pure performance measurement,

which can be misleading, we suggest further analysis of raw data and the meaning of the features in the context of the applied machine learning algorithms.

5. Conclusions

In this paper, we have presented a comprehensive method for entropy-based network traffic anomaly classification empowered by a novel protection mechanism against the deception of entropy detection capabilities. We contribute to the research topics in several directions.

Firstly, we have compared the ways how the Shannon, Tsallis and Rényi entropies respond to changes in feature distribution caused by spoofed traffic injected to deceive the entropy detection systems. We have found that a total number of elements in distribution, so-called a distribution length, is an efficient metric to detect entropy deception attempts. If deception is applied, a distribution length is much longer, exceeding the threshold, either fixed or dynamically calculated. We have shown that the Rényi entropy with positive parameter α is the most resilient to deception since it requires the largest amount of spoofed traffic. However, this entropy type provides the lowest entropy values, and for some features with higher data variation, it is not suitable to detect entropy drops.

Secondly, we have formalized and generalized the concept of aggregation and behaviour features, which better represents the network traffic structure using only basic flow attributes. Based on these features, we have modelled 16 anomaly models, associate with a wide range of security attacks. Extensive experiments were conducted for all anomaly models using full features set, calculating the Shannon, Tsallis and Rényi entropies, with both positive and negative parameter. Contrary to the widely accepted belief that the parameterized Tsallis and Rényi entropies outperform the Shannon entropy, we have shown that there is no significant difference in anomaly detection capability between these entropy types. The right choice of entropy type rather depends on the specific network traffic, its variety and deviations, used features and other parameters and characteristics, including the resilience to deception.

Thirdly, the conducted experiments confirmed that each anomaly model leaves a unique signature in the behaviour, indicating how entropies of different features are affected. Based on the multivariate analysis of different features, the original anomaly classification rules have been developed, which is another novel contribution presented in this paper. The efficiency of the anomaly classification method is validated through the presented experimental results.

Finally, but not less important, we believe that our work contributes in many respects to a better understanding of the entropy-based network behaviour analysis and anomaly detection, despite many papers in this research field. Based on the comprehensive experimental results and the conducted analysis, we have also concluded that supervised machine learning methods used for network behaviour analysis involve significant limitations for efficient practical use in real-time. Consequently, the proposed method based on the entropy of the basic flow data seems to be more feasible for practical implementation and general use. In this context, unsupervised machine learning, with no training required, could be a promising alternative solution.

Therefore, our further work will be oriented towards unsupervised machine learning, along with testing the concept and performances in various real-time network environments. This includes a classical approach using external data collection and processing system, as well as data plane programmability techniques on modern software defined networking architecture.

Acknowledgement. This work was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia under the EUREKA project "Network Traffic Anomaly Detection system based on NetFlow data analysis – TRADE" (grant number E! 13304).

References

1. J. Mazel, R. Fontugne, K. Fukuda, A taxonomy of anomalies in backbone network traffic, in: Proceedings of the 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, Cyprus, 4-8 Aug 2014: 30-36. IEEE. doi: 10.1109/IWCMC.2014.6906328.
2. G. Nychis, V. Sekar, D.G. Andersen, H. Kim, H. Zhang, An Empirical Evaluation of Entropy-based Traffic Anomaly Detection, in: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC "08), Vouliagmeni, Greece, 20–22 October 2008: 151–156. ACM New York, NY, USA. doi: 10.1145/1452520.1452539.
3. B. Tellenbach, M. Burkhart, D. Schatzmann, D. Gugelmanna, D. Sornette, Accurate network anomaly classification with generalized entropy metrics, Computer Networks 55 (11), (2011) 3485-3502, doi: 10.1016/j.comnet.2011.07.008.
4. P. Berezinski, B. Jasiul, M. Szpyrka, An entropy-based network anomaly detection method, Entropy 17 (4): 2367–2408, (2015) doi: doi.org/10.3390/e17042367.
5. I. Özçelik, R. R. Brooks, Deceiving entropy based DoS detection, Computers & Security 48, (2015) 234-245, doi: 10.1016/j.cose.2014.10.013.
6. B. Claise, Cisco Systems NetFlow Services Export Version 9, RFC 3954.
7. B. Li, J. Springer, G. Bebis, M.H. Gunes, A survey of network flow applications, Journal of Network and Computer Applications 36 (2), (2013) 567–581. doi: 10.1016/j.jnca.2012.12.020.
8. M. Ahmed, A.N. Mahmood, J. Hu, A survey of network anomaly detection techniques, Journal of Network and Computer Applications 60, (2016) 19–31. doi: 10.1016/j.jnca.2015.11.016.
9. V. Chandola, A. Banerjee, V. Kumar, Anomaly Detection: A Survey, ACM Computing Surveys 41 (3), (2009), doi: 10.1145/1541880.1541882.
10. N. Moustafa, J. Hu, J. Slay, A holistic review of Network Anomaly Detection Systems: A comprehensive survey, Journal of Network and Computer Applications 128, (2019) 33-55. doi: 10.1016/j.jnca.2018.12.006.
11. M.F. Umer, M. Fahad, M. Sher, Y. Bi, Flow-based intrusion detection: Techniques and challenges, Computers & Security 70, (2017) 238-254. doi: 10.1016/j.cose.2017.05.009.
12. A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, B. Stiller, An Overview of IP Flow-Based Intrusion Detection, IEEE Communications Surveys & Tutorials 12 (3), (2010) 343 – 356. doi: 10.1109/SURV.2010.032210.00054.
13. C.E. Shannon, A mathematical theory of communication, Bell system technical journal, 27(3), 1948, 379-423, doi: 10.1002/j.1538-7305.1948.tb01338.x.
14. N. Moustafa, G. Creech, J. Slay, Flow Aggregator Module for Analysing Network Traffic, Progress in Computing, Analytics and Networking, Advances in Intelligent Systems and Computing, vol. 710 (2018) 19-29. Springer, Singapore. doi: 10.1007/978-981-10-7871-2_3.

15. A. Lakhina, M. Crovella, C. Diot, Diagnosing Network-Wide Traffic Anomalies, ACM SIGCOMM Computer Communication Review 34 (4), (2004) 219-230. doi: 10.1145/1030194.1015492.
16. P.D. Bojovic, I. Basicevic, S. Ocovaj, M. Popovic, A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method, Computers and Electrical Engineering 73, (2018) 84-96. doi: 10.1016/j.compeleceng.2018.11.004.
17. O. Joldzic, Z. Djuric, P. Vuletic, A transparent and scalable anomaly-based DoS detection method, Computer Networks 104, (2016) 27-42. doi: 10.1016/j.comnet.2016.05.004.
18. D. Roosi, S. Valenti, Fine-grained traffic classification with netflow data, in: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, Caen, France, June 28 - July 02 2010: 479-483. ACM New York, NY, USA. doi: 10.1145/1815396.1815507.
19. P. Barford, J. Kline, D. Plonka, A. Ron, A signal analysis of network traffic anomalies, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, Marseille, France, November 06 – 08 2002: 71-82. doi: 10.1145/637201.637210.
20. H.A. Nguyen, T. Van Nguyen, D.I. Kim, D. Choi, Network Traffic Anomalies Detection and Identification with Flow Monitorin, 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08), Surabaya, Indonesia, 5-7 May 2008. IEEE. doi: 10.1109/WOCN.2008.4542524.
21. R. Braga, E. Mota, A. Passito, Lightweight DDoS flooding attack detection using NOX/OpenFlow, IEEE 35th Conference on Local Computer Networks, Denver, USA, 10-14 Oct. 2010. IEEE: 408–415. DOI: 10.1109/LCN.2010.5735752.
22. Y. Feng, R. Guo, D. Wang, B. Zhang, Research on the Active DDoS Filtering Algorithm Based on IP Flow, 2009 Fifth International Conference on Natural Computation. Tianjin, China, 14-16 Aug. 2009: 628–632. IEEE.
23. A. Lakhina, M. Crovella, C. Diot, Mining Anomalies Using Traffic Feature Distributions, Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications 35 (4), (2005) 217-228. doi: 10.1145/1080091.1080118.
24. T. Pevný, M. Rehák, M. Grill, Identifying suspicious users in corporate networks, Proceedings of workshop on information forensics and security: 1–6, (2012).
25. C. Tsallis, Possible generalization of Boltzmann–Gibbs statistics, Journal of Statistical Physics 52 (1-2), (1988) 479–487, doi: 10.1007/BF01016429.
26. K. Xu, Z.L. Zhang, S. Bhattacharyya, Internet traffic behaviour profiling for network security monitoring, IEEE/ACM Transactions on Networking 16 (6), (2008) 1241-1252, doi: 10.1109/TNET.2007.911438.
27. A. Rényi, On measures of entropy and information, in: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability 1, (1961) 547–561.
28. K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V.Maglaris, Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments, Computer Networks, Vol 62, (2014), 122-136, doi: 10.1016/j.bjp.2013.10.014
29. J. Ibrahim, V. Timčenko, S. Gajin, A comprehensive flow-based anomaly detection architecture using entropy calculation and machine learning classification, Proceedings of the 9th International Conference on Information Society and Technology, ISBN 978-86-85525-24-7, (2019) 138-143
30. V. Timčenko, S. Gajin, Time-series entropy data clustering for effective anomaly detection, Proceedings of the 10th International Conference on Information Society and Technology, Information Society of Serbia, ISBN 978-86-85525-24-7, (2020) 170-175.
31. R. Sadre, A. Sperotto, A. Pras, The effects of DDoS attacks on flow monitoring applications, 2012 IEEE Network Operations and Management Symposium, (2012) 269-277, doi:10.1109/NOMS.2012.6211908.

32. L. Ertöz, E. Eilertson, A. Lazarevic, P.N. Tan, V. Kumar, J. Srivastava, P. Dokas, Chapter 3: The MINDS - Minnesota Intrusion Detection System, Next Generation Data Mining, MIT Press, Boston.
33. C. Fachkha, E. Bou-Harb, M. Debbabi, Fingerprinting Internet DNS Amplification DDoS activities, in: 6th International Conference on New Technologies, Mobility and Security (NTMS), Dubai, United Arab Emirates, 30 March-2 April 2014, 1–5, doi: 10.1109/NTMS.2014.6814019.
34. A. J. Lawrance, P.A.W. Lewis, An exponential moving-average sequence and point process (EMA1), Journal of Applied Probability 14 (1), (1977) 98-113, doi: 10.2307/3213263.
35. NetVizura, "NetVizura Netflow Analyzer, Case study – DDoS Attack by NTP Amplification.", Accessed 22 July 2020. https://www.netvizura.com/files/products/netflow/resources/doc/DDoS-Attack-by-NTP-Amplification-NetVizura.pdf.
36. M. Allman, V. Paxson, J. Terrell, A brief history of scanning, in: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, San Diego, California, USA — October 24 - 26, (2007) 77-82. doi: 10.1145/1298306.1298316.
37. R. Hofstede, L. Hendriks, A. Sperotto, A. Pras, SSH compromise detection using NetFlow/IPFIX, ACM SIGCOMM Computer Communication Review 44 (5): 20–26. ACM New York, NY, USA, (2014) doi: 10.1145/2677046.2677050.
38. I. Sharafaldin, A.H. Lashkari, A. Ghorbani, Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterizatio, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP, ISBN 978-989-758-282-0, (2018) pages 108-116. doi: 10.5220/0006639801080116.
39. S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Computers and Security Journal 45, (2014) 100-123. doi: 10.1016/j.cose.2014.05.011.

Juma A. Ibrahim received his BSc from the University of Tripoli, Faculty of science, Computer science department, and MSc in Computer Engineering from the University of Belgrade, School of Electrical Engineering. He worked as a professor at the College of Computer Technology, Tripoli, and CCNA instructor at the Cisco Networking Academy, Tripoli and Injella, Libya. He is currently a PhD student at the University Of Belgrade, School Of Electrical Engineering, at the Department of Computer Science and Information Technology. His research interest includes computer networks and security, with the special attention to network intrusion detection and prevention systems.

Slavko Gajin received his BSc, MSc and PhD in Computer Engineering from the University of Belgrade, School of Electrical Engineering. He is currently working as a director of Computer Centre of the University of Belgrade and an associate professor at the University of Belgrade, School of Electrical Engineering at the Department of Computer Engineering and Information Theory, teaching topics in the field of computer networks. His research interests span from network performance monitoring and management to network security, network behaviour analysis and machine learning. He worked on the development of the Serbian Research and Education Network (AMRES).

Received: December 29, 2020; Accepted: July 12, 2021.

Computer Science and Information Systems 19(1):117–139 https://doi.org/10.2298/CSIS200531039S

Scaling industrial applications for the Big Data era

Davor Šutić and Ervin Varga

Faculty of Technical Sciences, Trg D. Obradovića 6,
21000 Novi Sad, Serbia
{sutic, evarga}@uns.ac.rs

Abstract. Industrial applications tend to rely increasingly on large datasets for regular operations. In order to facilitate that need, we unite the increasingly available hardware resources with fundamental problems found in classical algorithms. We show solutions to the following problems: power flow and island detection in power networks, and the more general graph sparsification. At their core lie respectively algorithms for solving systems of linear equations, graph connectivity and matrix multiplication, and spectral sparsification of graphs, which are applicable on their own to a far greater spectrum of problems. The novelty of our approach lies in developing the first open source and distributed solutions, capable of handling large datasets. Such solutions constitute a toolkit, which, aside from the initial purpose, can be used for the development of unrelated applications and for educational purposes in the study of distributed algorithms.

Keywords: distributed computing, big data, smart grid.

1. Introduction

Large industrial complexes, e.g. utilities or factories, rely on timely and accurate telemetry data as well as layers of redundancies that keep the production going even in the case of a failure. Up until the recent past, applications behind their operation were focused on a relatively small and static amount of data. Once set up, the infrastructure needed periodic maintenance, but had little demand for a change of scale.

With the advent of Smart Grid infrastructures this concept gradually changed. It became common to change the scale of the operation by adding more customers or introducing smart devices into the ecostructure. The increasing amount of required data demands puts also an additional strain on the available computational power. The algorithms used for analyzing aspects of the operation are usually non-trivial and thus the large amounts of data challenge their applicability. However, with the changing scope of the applications, the preferred infrastructure shifted to larger distributed systems, whether in the cloud or not.

In this paper, we present three projects that illustrate how hard industrial computational problems are solved on large datasets. The unifying factors of all three solutions are a common framework and distributed environment. They rely on Apache Spark to provide a common infrastructure in order to share a communication foundation and facilitate comparison between them. They also primarily target large datasets, i.e.

scales that would be hard for non-distributed applications to compute in a reasonable time.

The first two projects can be put under an umbrella of smart grid power analysis. They introduce support for the power flow and contingency analysis functions. The power flow analysis is performed using the Newton-Raphson method, while the contingency analysis is performed in two distinct approaches, the network connectivity state is assessed through the analysis of the graph constructed from the connected components of the network and through the binary multiplication of Boolean matrices.

The third addresses a missing utility in graph processing algorithms. The complexity of processing a graph quickly increases with its size, so it would be beneficial to decrease the complexity of the graph while maintaining its mathematical properties. Here, we provide a reusable distributed spectral graph sparsification solution. Reducing the number of vertices is usually not desired due to their semantic importance. Luckily, real-world graphs tend to have more edges than vertices, so reducing the number of edges both reduces the size of the problem and doesn't affect the semantic of the dataset.

This paper shows the applicability of the Apache Spark framework to industrial applications. The open source [15][16] solutions herein are the first of their kind both in handling large datasets in a distributed manner and in the map reduce paradigm, which also motivated the choice of the problems.

The paper is outlined as follows: The next section addresses related work. Section 3 presents the power flow problem and outlines our solution. Section 4 presents the island detection problem and outlines our solution. Section 5 presents the spectral graph sparsification problem and outlines our solution. The following section details the experimental setup and its results. Finally, we conclude the paper, by presenting a short overview of the contributions and an outlook for future research.

2. Related work

Being a distributed data processing engine, Apache Spark [1] has since its introduction found a wide range of users. Applications include various disciplines where the problem can be reduced to analyzing large amounts of data, like genomic analysis [2] and specialized mathematical methods for matrix computation [3].

The power flow problem was stated decades ago [4], however, once the Newton-Raphson method was introduced [5], only one other solution method was developed [6]. Improvements have mainly been directed towards the benefit of mathematical apparatus, pre-dominantly focused on matrix algebra, used by the solutions methods.

The island detection problem, that is an integral part of contingency analysis, was prominently approached in [7], by using a network connectivity matrix in conjunction with Boolean algebra. Yet, in the paper discussion J. L. Marinho et al. challenge the solution by calling it "unnecessarily complex" when compared to graph analysis approach. The authors' rebuttal accentuates the advantages of their solution and state that in their experience the proposed graph-search algorithms were not faster. This exchange is important as it sets the main directions of island detection research early on, towards matrix analysis improvement [10], [8], or towards more advanced topology analysis [9]. Finally, it constitutes the main incentive for us to compare both approaches.

Currently, the most prominent open source power analysis tools are based on MATLAB [11], [12]. From the Java based tools, it is worth to mention DCOPFJ [13] and InterPSS [14]. Yet, what all these tools lack is a distributed solving mechanism.

The notion of spectral graph sparsification, that is relevant to this paper, was introduced by Spielman and Teng [17] and focuses on the spectral similarity between a graph and its sparsifiers. Their main result is the proof that every graph has a near-linear sized spectral sparsifier that can be computed in near-linear time. However, as they state, the powers of logarithms and the constants in their achieved upper bound are too large to be of practical importance, but their goal was in any case to prove that such sparsifiers exists and not the optimization of the process. In this paper we focus on exactly that practical aspect and show a solution that can successfully sparsify large graphs with relatively modest resources in a practically acceptable amount of time. Further, [17] is the second in a series of three papers [18][19] with the ultimate goal to develop efficient methods for solving linear systems in symmetric, weakly diagonally dominant matrices. In terms of spectral graph theory, that is important for finding the eigenvalues of Laplacian matrices. Our paper uses the graph partitioning algorithms presented in [18] to support the sparsification effort. The quest for the eigenvalues of a graph's Laplacian [19] is beyond the scope of this paper, however, it constitutes a logical continuation of the herein described approach. Another implementation was done by Perraudin et al. [35] in order to extend an existing open-source graph signal processing toolbox. Being written in MATLAB, the solution is inherently non-distributed and highly specialized, which limits its efficacy when managing large datasets. Thus, the approach fundamentally differs to our solution.

Spielman and Teng [17], and Spielman and Srivastava [20] inspired further research in finding a distributed approach. Koutis and Xu [21] introduce a theoretical algorithm for spectral sparsification. Their work focuses on the use of weighted spanners. The computation of which is, however, complex and expensive in terms of resources. That is why we choose to build upon the original algorithm [17], which carries an arguably more intuitive set theory mindset. Unfortunately, Koutis and Xu [21] didn't provide any benchmark, so it is hard to compare our solution to their approach. Similarly, Sun and Zanetti [22] approach the sparsification problem from a clustering perspective avoiding spectral methods. Their experiments focus on the functionality of their algorithm. The datasets they use are magnitudes smaller than ours, so a direct comparison is hard to make. However, they argue that spectral sparsification methods are complex and thus unsuitable for the distributed setting, which we disprove in this paper.

Šutić and Varga [23] expanded the Apache Spark GraphX library [1][25] with the notion of distributed spectral graphs and basic spectral analysis operations. Some of them, e.g. the Laplacian matrix calculation, are used in this paper, but otherwise are the contributions of this work a logical extension of the existing framework.

Spectral graph sparsification has important applications. By reducing the number of edges in graphs, while preserving the properties, methods that were previously too expensive to use, become tractable and applicable. A particularly illustrative example is the problem setting of the work by Zhao et al. [26]. Namely, they propose a new method based on spectral graph sparsification for the modelling and simulation of large power delivery networks. They look back at various methods for achieving that and conclude that none can rise to the challenges of the complexity of contemporary power grids while simultaneously keeping the required accuracy. The proposed solution itself uses

spectral sparsification to reduce the grid graphs, however, it is deemed too computationally expensive for large graphs, so the authors resort to grid partitioning to keep the resulting graphs manageable for sparsification. Our goal is to provide feasible and scalable sparsification of large graphs.

3. Power flow solution

3.1. Problem statement

The subject of the power flow problem is balancing the required load and associated losses with the production capacities of generators in a power grid. The knowledge of one complex characteristic in all nodes of the system, in this case voltage, can be used to reconstruct the complete regime of the system. In other words, the voltage magnitudes and phase angles of each bus constitute the stationary state of the system.

The state of a system of N nodes is defined by N complex equations. The Newton-Raphson method is a well-known approach for solving systems of non-linear equations. Its modus operandi is to approximate a non-linear problem, like a system of N complex power-flow equations, into a linear matrix equation and solve it iteratively.

For massive grids, the corresponding system of linear equations is large. Solving such system is non-trivial, as just computing the inverse of the matrix is challenging, and it has to be done iteratively, which further increases the challenge.

3.2. Solution Approach

The algorithm implemented in our solution is presented in **Fig. 1**. The general idea is to perform fast localized tasks (e.g. initializing the input parameters of the matrix equation and checking the solution) on the driver machine, while distributing compute intensive operations to the worker nodes (e.g. solving the matrix equation).

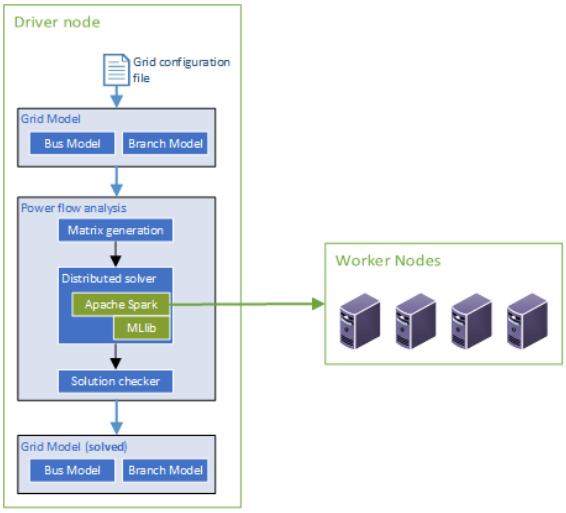


Fig. 1 An overview of the architecture. It shows what parts of the power flow analysis are performed on the driver node and which are distributed across a collection of worker nodes

The input parameter is a GridModel [33], loaded from structured text files that define all known values in a grid. First, it needs to be initialized, i.e. the starting assumptions set and the admittance matrix generated.

Attempts to arrive at a satisfying solution are iteratively made until either the solution is accepted or the maximum number of iterations is reached.

Using Apache Spark, we devised an algorithm that solves the general type of matrix equations ($\mathbf{b} = \mathbf{Ax}$) in a distributed manner. This addresses the hard problem in the power flow calculation [34]: efficiently and repeatedly solving a matrix equation. As a high-level overview of this algorithm, the input parameters are prepared on the driver node and then distributed as Resilient Distributed Datasets (RDDs) [3] across the allocated worker nodes for solving. The result of the operation is passed back to the driver for checking and the setup of the next iteration.

The initial step is parallelizing the input matrix into an RDD. RDDs are special abstractions of collections of objects that represent the basic operating object of all Apache Spark jobs. They are partitioned and distributed across available worker nodes and are fault-tolerant in terms of failure of a job or worker machine. Every RDD has exactly one underlying type which is defined by the collection used for RDD creation.

Spark's MLLib library extends the RDD paradigm by introducing distributed abstractions atop of it. Particularly interesting here are distributed matrix types. For instance, the RowMatrix represents a row-oriented matrix with no meaningful indices, while the IndexedRowMatrix introduces indexed rows. The BlockMatrix views a matrix as a distributed collection of smaller submatrices and a CoordinateMatrix is particularly suitable for sparse matrices, as it is organized as a distributed collection of tuples that define the value of the entry and its coordinates, i.e. the row and column, in the matrix. All of them provide different operations, depending on their logical organization, but converting from one to another must be done carefully, since that may result in

reshuffling of data, arguably the most expensive operation in the Spark environment from a performance perspective.

```
Algorithm 1: Distributed solving of matrix equation b = A·x
1.    procedure Solve (Matrix A, Vector x, Vector b)
2.       rowsRDD ← convert A into RDD of IndexedRows
3.       indexedRowMatrix ← convert rowsRDD into IndexedRowMatrix
4.
5.       sdd ← compute the Singular Value Decomposition of
      indexedRowMatrix
6.       U ← get the U value from sdd, as IndexedRowMatrix
7.       S ← get the S value from sdd, as Vector
8.       V ← get the V value from sdd, as Matrix
9.
10.      Utrans ←  transpose the value U, as IndexedRowMatrix
11.      UtransB ← Utrans multiplied with Vector b, as
      IndexedRowMatrix
12.      UtransBSinv ← UtransB multiplied with inversed Vector S, as
      RDD of Tuples
13.
14.      UtransBSinvVector ← UtransBSinv collected to local Vector
15.
16.      x ← UtransBSinvVector multiplied with V
```

Fig. 2. Algorithm for distributed solving of matrix equations

The IndexedRowMatrix is particularly interesting for the problem at hand, because it offers two methods of matrix factorization: the QR decomposition and the Singular Value decomposition (SVD). And it offers the additional benefit of indexed rows over the RowMatrix. The only drawback is that, per Spark source code, many operations perform a to-RowMatrix cast first and then issue the operation on the RowMatrix type with optional re-indexing afterwards, which carries a certain performance penalty. The Spark documentation suggests that casting from one distributed type to another may be expensive, however our experience shows that in some scenarios, a cast outperforms an alternative, more complex implementation. This is typically the case when a data shuffle is inevitable, be it performed by a cast or required by a custom implementation.

The choice of a matrix factorization is important for finding the inverse of $\mathbf{A}$ to solve the matrix equation

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \ . \qquad (1)$$

Our choice is in favor of SVD [28]. It produces orthogonal matrices, so their conjugate transpose is at the same time the inverse. Therefore, by performing a SVD, obtaining the inverse of the matrix becomes easier.

Once the matrix equation is solved (**Fig. 2**), a convergence check is performed.

In the case that the current iteration converged, the input GridModel contains the recent changes in terms of bus parameters, i.e. the solved state of the system. The GridModel instance can be used for analysis of the system state or for further simulations.

Finally, in the case when the Newton-Raphson method fails to arrive at an acceptable solution after a given number of iterations, the calculation fails. The predefined maximum number of iteration is also empirically determined. For flat start calculations,

we found that the maximum number of iterations needed is 6 (see experimental results), so 10 iterations present a reasonable maximum iteration threshold.

4. Island detection

4.1. Problem statement

The purpose of contingency analysis is assessing the impact of potential outages on the smart grid. This makes it an important simulation tool.

A contingency analysis is performed in two steps, as shown in **Fig. 3**. First, the connectedness of the network is determined. Second, if any islands were found, their state is assessed by running power flow analyses on each island and checking for irregularities.

The connectedness analysis determines whether all nodes in a network are connected to every other node, at least indirectly. This problem can be reduced to the connected components problem, which is a common challenge in graph theory.

Another method is the full matrix analysis [29], which may be considered the "classic approach". The idea is to generate a Boolean connectedness matrix whose element with the indices *i* and *j* is set to one, if nodes *i* and *j* are connected, and zero otherwise. The matrix is symmetrical. The elements on the main diagonal are always set to one.

Algorithm 2: Contingency analysis

```
1.    procedure PerformContingencyAnalysis (GridModel gridModel,
      Branch from, Branch to)
2.        baseGridModel ← PowerFlowCalculation(gridModel)
3.        gridModelWithOutage ← CreateOutage(gridModel, from, to)
4.
5.        islandGridModels ←
          PerformConnectednessAnalysis(gridModelWithOutage)
6.
7.        foreach island in islandGridModels do
8.            PowerFlowCalculation(island)
9.            Identify irregularities in grid parameters
```

Fig. 3. Contingency analysis algorithm

Binary multiplying the connectedness matrix with itself yields a connectedness matrix of the second level. If an element with the indices *i* and *j* was zero in the first level matrix, but changed to one in the second level matrix, that means that the nodes *i* and *j* are indirectly connected with one other node in-between. Further binary multiplication with the resulting matrix yields connectedness matrices of higher levels, each identifying deeper connections between disconnected nodes. Consequently, in a system of N nodes, the algorithm stops after (N-2) iterations. If the initial matrix, raised to the (N-2) power through binary multiplication, still contains elements equal to zero, these nodes have a distance larger than N elements between each other. Therefore, that set of zero-valued elements indicates components that are not connected to the main body of the network.

A fully connected network would result in a matrix where all the elements are equal to one.

One stopping criterium is checking whether the (N-2)-th power of the initial connectedness matrix contains any zero-valued elements. However, it can be relaxed. The network is fully connected, if any binary multiplication arrives at an all-ones matrix. The algorithm can then be halted, since further multiplications will not change the outcome. The more general halting condition is therefore that two subsequent multiplications did not change the resulting matrix. That means there can be no more connections, since no new links between any nodes were discovered.

In the final matrix, islands are identified by analyzing its rows or columns for linear dependence. Each linearly independent row or column represents an island. The nodes constituting it are defined by the indices of all the elements equal to one within that row or column.

4.2. Solution Approach

It is our goal to distribute the compute intensive work as much as possible. For that purpose, we implemented two approaches. The first uses the Apache Spark GraphX [25] library to analyze the network with graph analysis and the second uses binary matrix multiplication to isolate the potential islands. A high-level overview of the solution is shown in **Fig. 4**.

As above, the input is the GridModel.

We need to determine a base state of the system before simulating any outages, by performing a power flow calculation with the unchanged GridModel. The base state serves to assess the deviations introduced by outages once the procedure completes.

Performing a connectedness analysis, whose output is a list of potential islands, is where the two mentioned implementations differ. Both approaches share the same interface.

We shall first discuss the graph approach, as it is more straightforward and then move on to the matrix binary multiplication algorithm.

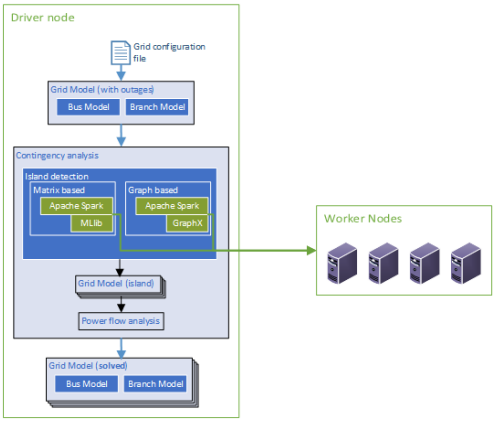


Fig. 4. An overview of the architecture. It shows what parts of the contingency analysis are performed on the driver node and which are distributed across a collection of worker nodes

Representing a smart grid network with a graph comes naturally. A vertex collection is created from all buses in the bus model. Edges are similarly created from the branch model. The GraphX graph object is created using those two RDD collections. The graph object supports a connectedComponents operation, which returns another graph object. Practically, transforming the vertex RDD into a list of islands and returning this list to the driver, solves the problem.

The matrix binary multiplication approach is still the de facto standard method, albeit with significant improvements developed over the years, for many industrial analysis systems. Further, distributed matrix operations, including multiplication, are the domain of Spark's MLlib [30]. So, the reasons of legacy consideration and challenge to implement a distributed binary matrix multiplication in Spark prompted us to develop the second island detection approach.

Our distributed binary matrix multiplication operates in three phases:

1. Generate the initial connectivity matrix based on the state of the system
2. Find the stable island matrix
3. Identify the islands from the island matrix

The connectivity matrix is generated from the branch model. For its generation it is convenient to use a CoordinateMatrix type, because the underlying MatrixEntry, which consists of the two coordinates and the value, best fulfills the need to set a number of elements individually. Further, as the connectivity matrix represents the physical connections between the nodes in the network, it is very sparse, especially for large systems, so relatively few MatrixEntries are required. Yet, because the BlockMatrix is the only matrix type that supports multiplication with another distributed matrix, which is the focus of this algorithm, the generated CoordinateMatrix is cast to a BlockMatrix at the end of this phase. This cast does not induce any significant performance penalty, as opposed to directly creating a BlockMatrix, because of the sparsity of the matrix and the overhead the block-based approach would require during the initialization.

The next phase determines the island matrix. Taking the BlockMatrix from the previous phase, it is repeatedly multiplied with itself. The maximum number of multiplications depends on the number of buses in the system and is equal to

$\lceil \log_2(n_{buses}-2) \rceil$. The goal is to raise the connectivity matrix to the $(n_{buses}-2)$ power and this is a much more efficient way. After each multiplication, all values greater than zero are set to one. The procedure halts when the maximum number of iterations is reached or when the matrix is completely filled with ones. A real-world, connected network usually yields the all-one matrix after few iterations.

The final phase evaluates the island matrix for occurring islands. To keep the result consistent with the graph approach, a connected network results in a single list of all nodes, with the consideration that that network contains the designated number of outages.

Regardless of chosen approach, the obtained list of islands is evaluated. From each list a new GridModel is created that matches the subnetwork of the island.

5. Spectral graph sparsification

5.1. Problem statement

The most interesting fact about spectral sparsification is that it is possible to sparsify every graph in this way. In a narrower sense, spectral sparsification implies that a graph $G = (V, E, w)$ can be approximated by a sparse subgraph that retains the same Laplacian quadratic form as the original graph [31]. The Laplacian quadratic form is given by

$$x^T L_G x = \sum_{(u,v)\in E} w_{(u,v)} \big(x(u) - x(v)\big)^2, \quad (14)$$

where x is a real vector of V elements, and L_G is the Laplacian matrix of G [12][17]. Strictly speaking, Spielman and Teng [17] consider $\tilde{G}$ to be a σ-sparsification of G if the following relation holds for all x:

$$\frac{1}{\sigma} x^T L_{\tilde{G}} x \le x^T L_G x \le \sigma x^T L_{\tilde{G}} x. \quad (15)$$

Basically, finding $\tilde{G}$ for a given graph G is the aim of our solution and its core method *sparsify*.

5.2. Solution Approach

Overview

Here we are extending the existing open source spectral graph library [23][16], that builds upon the theoretical algorithms and examples of Spielman et al [17]-[20][31]. It already has some basic methods, some of which are used for the subsequent implementation.

We extend the public API with the following methods: *volume*, *conductance*, *sum*, and *sparsify*. The starting point for the considerations is always graph G, the GraphX graph object the class is initialized with and which serves as the primary input. When we discuss edges, vertices, subgraphs, etc., it is done with regard to G unless otherwise noted. Further, the cornerstone of the approach is viewing the problem from the set perspective. By focusing on that aspect, the set of vertices, edges, weights, i.e. the usual ways to define a graph, become natural subjects of GraphX and the underlying Spark RDD [1] paradigm. Thus, it is easy to scale a large graph to a distributed environment and making computationally expensive operations feasible on a large dataset.

Volume

Volume calculates the volume of a subset of vertices. It is a sum of the degrees of each vertex in the set. The degree of a vertex is equal to the number of its incident edges. Here, we distinguish two cases. The volume is constant, if the subset is in fact the whole set of vertices of G, and is equal to twice the number of edges in G. In the case of a true subset, we first calculate the degrees of all vertices and create a set of tuples that matches each vertex' unique ID with its degree. The resulting set is joined with the subset of vertices by vertex ID, leaving only the vertices that were part of the subset mapped together with their respective degrees. This set is reduced to a sum of the degrees within producing the required volume value.

Conductance

The conductance of a graph, also called Cheeger's constant, is formally related to the convergence of a random walk on the graph to a uniform distribution. The name derives from the similarity to the significance of random walks in electrical networks. Here, conductance is used primarily for graph partitioning in the sense of evaluating the quality of a local cluster. A cluster is considered of high quality, if it is extensively interconnected within itself, but rather sparsely with the rest of the graph. In other words, conductance is the ratio of the number of edges connecting the cluster with the rest of the graph and the number of edges within [18].

In this solution, the conductance is calculated with regard to a given vertex set, i.e. a cluster, that is part of a given subgraph, which in the general case can also be the whole graph G. First, we calculate the number of edges crossing out of the set, i.e. we identify the edges whose one vertex is inside the set, while the other reaches outside. Next, the volumes of both the given set of vertices and the volume of the remaining set of vertices in the whole graph G are calculated. At this point it is important to emphasize, that when measuring and volumes of vertices in the vertex-induced subgraphs, we will continue to measure the volume according to the degrees of vertices in the original graph G. As mentioned above, the resulting conductance is the ratio of the obtained inter-cluster edge count and the smaller of the two volumes.

Sum

The concept of adding two or more graphs together might seem counterintuitive. However, as we will see later on, an important step in graph sparsification is the so-called partitioning into certain subgraphs, whose sum results in a single sparsified graph.

In the spectral sense, a sum of two graphs produces a graph whose Laplacian matrix is equal to the sum of their Laplacian matrices. In practical terms, this means that every edge in the resulting graph is equal to the sum of the corresponding edges in the two constituent graphs. In the case that the vertex sets of the corresponding graphs are disjoint, the sum is a simple union of graphs.

The resulting graph is defined by its vertex and edge sets. Obtaining the vertex set is trivial, it is only a simple union of the vertex sets of the two addend graphs. In the case that an edge is part of both addends, the corresponding edge in the resulting graph will have a weight equal to the sum of their weights. If an edge is only present in one of the addends, the same principle applies, while the non-existing edge will be treated to have weight zero. One problem that arises is the uniform designation of edges. In the general case, an edge is defined by its source vertex, destination vertex, and weight. Given that we consider here undirected graphs, there is no distinction between edges that have their (same) vertices swapped, i.e. for summation purposes, such edges are considered the same and should be added accordingly. However, there is no guarantee nor binding rule that the edges in the addends are not swapped. To alleviate this case, we manipulate both edge sets by assuring an ascending order of vertex IDs in the tuple that defines the edge, i.e. $(a, b) => (b, a), if\ a > b$. This provides a kind of unique key so that the corresponding edges, that generally can have different weights, can be joined together by the vertices they connect, which results in set that uniquely maps the incident vertices to the weights that the corresponding edge has in both addend edge sets. A full outer join guarantees that even when an edge is not present in the other graph, it will still be present in the joined set with a special designation (concretely, the type None) depicting the “missing” weight. The resulting edge set is obtained by conditionally summing up the weights and keeping the vertex IDs. Thus completing the other requirement for the sum of two graphs.

Sparsify

Sparsify is a complex method at the core of the sparsification process. Broadly speaking, it consists of two major steps, that are distinctive, yet conjoined through common weight adjustments. First, we partition the graph and sample the resulting subgraphs’ edge sets. Then, the resulting graphs are contracted together into a single sparsified graph.

In order to support graphs with arbitrary weights, sparsify is limited to graphs that have fractal weights that are greater than zero and at most equal to one. This restriction can be easily overcome by simply scaling all weights down before sparsifying and scaling them up afterwards. This is another task that is rather trivially fulfilled using Spark RDD operations, even for very large graphs. However, the problem here is not the possibility and cost of scaling, but the fact that the weights can be truly arbitrary, i.e. a large number of digits after the decimal point, and increase the complexity of the calculation and can lead to the inability to construct a sparsified graph. Therefore, the

weights are scaled down to a certain number of bits after the decimal point before proceeding. Depending on this number of bits, the same number of subgraphs is created based on the binary representation of the individual adjusted edge weights. Each of those subgraphs is then subjected to the following operations.

Before proceeding to the partitioning and sampling of the graph, there is an issue worth mentioning. There is a reported, and as of yet not completely resolved, issue [32] that under certain circumstances the execution of *connectedComponents* gets stuck in an endless loop. We did occasionally observe such behavior when working with massive graphs. Given that the issue seems also related to the workload of the environment and other factors, the simplest workaround is to restart the calculation.

When partitioning, the goal is to isolate a portion of the graph with a specified target conductance [18]. If the obtained partition is large, i.e. has a large conductance, both the partition and its remaining complement are recursively cut further, until the target conductance is reached. If a partition fulfills the target, further cutting is applied to its complement, until it too reaches the target. The result of the partitioning is a collection of subgraphs, that are induced by the obtained sufficiently small cuts. Each of the subgraphs is sampled, which is a random procedure, where the subgraph's vertex set remains unchanged while the edges are scaled and reduced based on a probability distribution. Formally [18], this step creates a $(1+\epsilon)$-approximation of the subgraph, where the ϵ is a rational parameter. This subgraph collection is, once the processing is completed, summed up back into a single graph. Such resulting graphs can already be considered sparsified to a degree.

At this moment, the initial graph has been decomposed into edge induced subgraphs, following the realignment of the edges' weights. Each of these subgraphs is individually partitioned and sampled into a sparsified version of itself, as we saw above. We further sparsify the current subgraph's edges by identifying those that contribute the least to the overall conductance of the subgraph. All that remains is to sum those modified subgraphs into a single sparsified graph that is returned as the result of the operation.

At a high level, the algorithm repeatedly breaks the graph into ever smaller, yet mostly overlapping, subgraphs and attempts to reduce the number of edges at each step. We observe another similar pattern at each of the sparsification steps. Each of the currently relevant subgraphs is broken up into a collection of subgraphs, their edges processed in some way, and reduced to a single graph by adding them up together (the operation Sum from above). Nothing is lost due to those repeated breakdowns, even as the subgraphs are usually overlapping, because the addition of graphs preserves the vertices and only manipulates the weights of edges that exist in any of the subgraphs. The abovementioned operations Conductance and Volume are used as limits while partitioning graphs.

6. Experimental results

6.1. Environment

For the evaluation computing infrastructure, we've chosen Amazon cloud computing platform, Amazon Web Services (AWS). Testing was conducted on configurations of Amazon Elastic Compute Cloud (EC2) instances. The performance results are limited to computing optimized (c types) and general-purpose GPU compute instances (p types), where applicable (see **Table 1**). Both are suitable for computation centered parallel tasks, which conceptually fits the needs of this paper. However, although significantly more powerful, the GPU-based instances proved to offer little performance gain, as seen below. The reason is that at the moment of testing, Spark didn't support the utilization of CUDA cores. The execution practically ignored the available GPU cores and focused solely on the CPU cores.

The experiments are executed using the Amazon Elastic MapReduce (EMR) framework. It offers an abstraction over the "vanilla" EC2 instances, which allows Spark and any related services (e.g. Ganglia, S3 support, etc.) to be automatically deployed and accessed when starting up a cluster.

Our experiments typically include two configurations for each instance type, in order to gauge the scaling-out of the solution. One that has five workers and one of ten, while both have one driver machine.

Table 1. EC2 instances used in the experiments and their technical specification

Instance type	CPU cores	GPU cores	GPU RAM	RAM
c3.4xlarge	16	n/a	n/a	30 Gb
c3.8xlarge	32	n/a	n/a	60 Gb
p2.8xlarge	32	8	96 Gb	488 Gb
p2.16xlarge	64	16	192 Gb	732 Gb

6.2. Test cases

For evaluating the smart grid analysis part of the solution, we used a collection of test cases which encompass experimental and special case networks, as well as real-world installations. A useful source of preselected grid examples is also the case repository that comes with MatPower [11]. We have conducted experiments on an array of 21 cases, with bus size ranging from 4 to 9241.

These networks proved unsuitable for proper graph sparsification demonstration, because they are not large enough. We chose two real world graph datasets for experimental evaluation. The data is publicly available [24] and represents weighted graphs. The first dataset is called "bio-mouse-gene" and represents a mouse gene regulatory network derived from analyzing gene expression profiles. It consists of 45101 vertices and 14506196 edges. The other dataset is called "bio-human-gene2" and similarly represents a gene regulatory network, but this time for humans. It consists of

14340 vertices and 9041364 edges. These graphs were chosen, because they have both semantic significance, i.e. are not artificially generated, and similarity, i.e. they represent genetic networks. Further, these are graphs that have a relatively high edge count, different densities, weight distributions, and topographical layouts.

It is important to point out that gene networks hold no special significance for the solution's approach or performance. For example, power networks and smart grids are also excellent inputs, especially due to the importance of weights (which are usually line admittances) in such graphs. However, such networks that are publicly available, tend to be rather small for demonstration purposes. We want to emphasize here the ability of our solution to operate in line with performance expectations of Spark, which excels in large datasets.

6.3. Results and discussions

Here, we show the experimental results achieved in various environments and parameters. The graphical representations follow a pattern in order to facilitate understanding: The vertical axis shows the subject being measured (e.g. execution time or reduced edge count), the horizontal axis shows the parameters used in each experiment, while the table that follows shows the exact values as opposed to graph lines. The first column there indicates the configuration on which the value was obtained (e.g. the number of worker nodes and the EC2 instance types).

Power flow solution

The execution results of a series of experiments under which the power flow calculation was tested, is shown in **Fig. 5**. The running times are scaled to the duration of one iteration of the Newton-Raphson method, as it takes a variable number of iterations to complete different cases (**Table 2**). We found that on average less than 2% of the total duration of a calculation is spent on the driver node preparing the current iteration (e.g. generating the Jacobian and $\Delta \mathbf{S}$ matrices) and evaluating the obtained results (checking the validity of the solution).

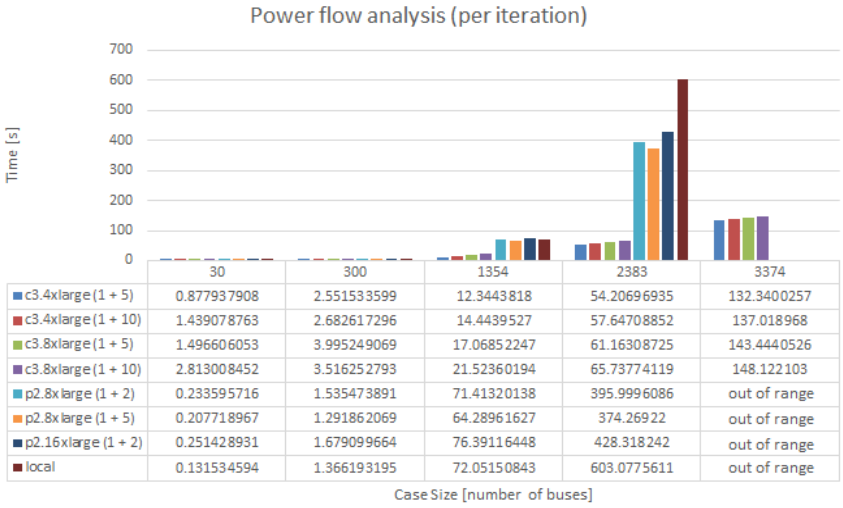


Fig. 5. The performance results of the power flow analysis, scaled to per-iteration values

Table 2. Number of iterations per power flow calculation

Buses	30	300	1354	2383	3374
Iterations	3	5	5	6	6

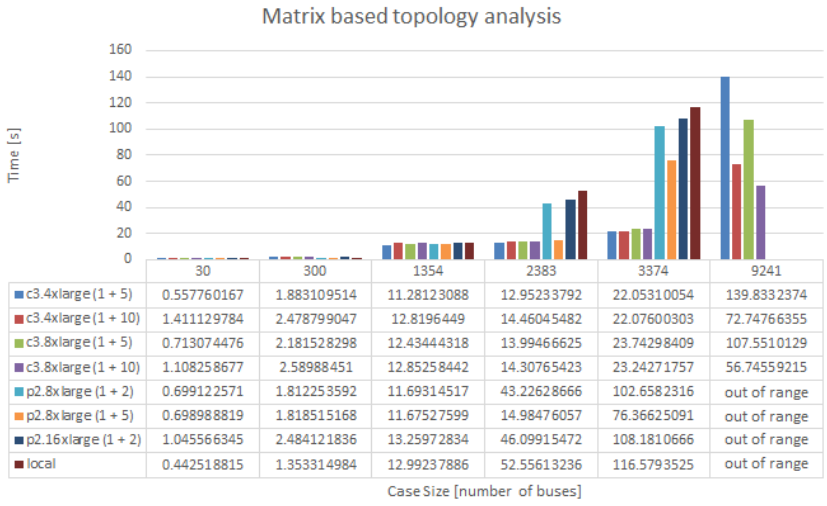
Measured times indicate a relative equality between distributed and local executions for lower bus-counts. We even observe that the local performance is better than the distributed equivalent. This is expected behavior, since the datasets are small enough that the overhead of partitioning them into many more partitions, sending them to the workers, and collecting them back, exceeds the computing effort itself. Further, it also shows that reasonably small cases are practically computable on single machines, which is why commercial power analytics software systems tend to break the network into smaller chunks before performing a power-flow calculation. However, the differences rapidly escalate with the growth of the network. For the grid of 3374 buses it became unfeasible to chart the result of the local execution, as it was in the domain of several hours. This is where the advantages of a distributed implementation truly outperform the local equivalent. Unfortunately, Spark's ability to perform a SVD is limited to matrices of at most 17515 columns. In the source code of the RowMatrix class, this is explained with the rationale that the matrix dimension "exceeds the breeze svd capability", so it is the issue of the Breeze [27] package, a numerical processing library for Scala that Spark uses. Thus, we were unable to perform power-flow analyses for the 9241-bus and larger grids. This also explains the slight performance degradation when scaling out and upgrading the c3 instances. Namely, the idea is to fully utilize the available computation power as much as possible. The 3374-bus test case had an average CPU utilization of 80% percent, when performed on a cluster of five workers, while the same case used no more than 30% percent on a cluster of ten instances. Once the SVD size limitation is

alleviated, the full performance impact can be gauged with larger networks, which would make full use of the available cluster resources.

Experiments on GPU instances were conducted using Databricks' Spark GPU. Due to the inhibition of the Spark functionality, they've shown similar performance as the local machine. The minor performance benefit is due to more advanced CPU and memory configuration.

Island detection solution

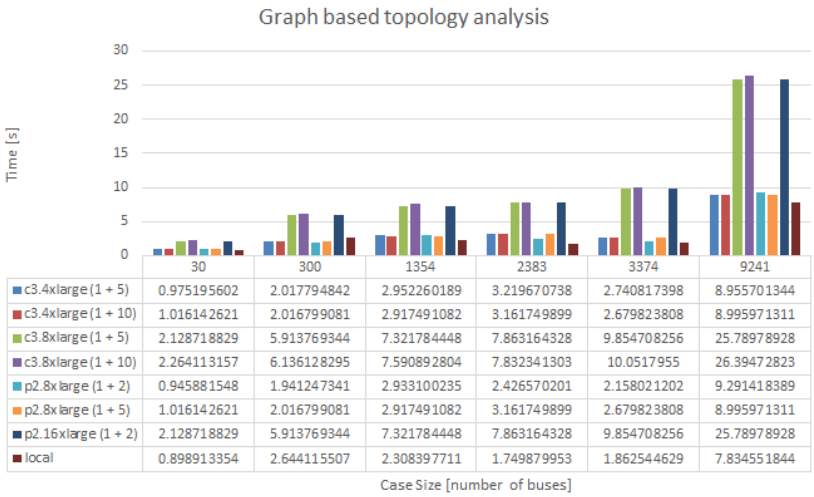
Measured times indicate a relative equality between distributed and local executions for lower bus-counts. We even observe that the local performance is better than the distributed equivalent. This is expected behavior, since the datasets are small enough that the overhead of partitioning them into many more partitions, sending them to the workers, and collecting them back, exceeds the computing effort itself. Further, it also shows that reasonably small cases are practically computable on single machines. However, the differences rapidly escalate with the growth of the network and the advantages of a distributed implementation truly outperform the local equivalent.



	30	300	1354	2383	3374	9241
c3.4xlarge (1 + 5)	0.557760167	1.883109514	11.28123088	12.95233792	22.05310054	139.8332374
c3.4xlarge (1 + 10)	1.411129784	2.478799047	12.8196449	14.46045482	22.07600303	72.74766355
c3.8xlarge (1 + 5)	0.713074476	2.181528298	12.43444318	13.99466625	23.74298409	107.5510129
c3.8xlarge (1 + 10)	1.108258677	2.58988451	12.85258442	14.30765423	23.24271757	56.74559215
p2.8xlarge (1 + 2)	0.699122571	1.812253592	11.69314517	43.22628666	102.6582316	out of range
p2.8xlarge (1 + 5)	0.698988819	1.818515168	11.67527599	14.98476057	76.36625091	out of range
p2.16xlarge (1 + 2)	1.045566345	2.484121836	13.25972834	46.09915472	108.1810666	out of range
local	0.442518815	1.353314984	12.99237886	52.55613236	116.5793525	out of range

Fig. 6. The performance results of the matrix based topology analysis

The performance chart of matrix based topology analysis is shown in **Fig. 6**. We observe performance improvement when increasing the number of workers and using better EC2 instances. The importance of scale-out is also evident, ten c3.4xlarge instances outperform five c3.8xlarge workers by about 30%. While in the extreme case, ten c3.8xlarge instances are close to three times faster than five c3.4xlarge workers. If we additionally consider that smaller networks show similar performance differences between c3 configurations, the results of the matrix topology analysis support the claim, that larger networks would make full use of the available cluster resources.



	30	300	1354	2383	3374	9241
c3.4xlarge (1 + 5)	0.975195602	2.017794842	2.952260189	3.219670738	2.740817398	8.955701344
c3.4xlarge (1 + 10)	1.016142621	2.016799081	2.917491082	3.161749899	2.679823808	8.995971311
c3.8xlarge (1 + 5)	2.128718829	5.913769344	7.321784448	7.863164328	9.854708256	25.78978928
c3.8xlarge (1 + 10)	2.264113157	6.136128295	7.590892804	7.832341303	10.0517955	26.39472823
p2.8xlarge (1 + 2)	0.945881548	1.941247341	2.933100235	2.426570201	2.158021202	9.291418389
p2.8xlarge (1 + 5)	1.016142621	2.016799081	2.917491082	3.161749899	2.679823808	8.995971311
p2.16xlarge (1 + 2)	2.128718829	5.913769344	7.321784448	7.863164328	9.854708256	25.78978928
local	0.898913354	2.644115507	2.308397711	1.749879953	1.862544629	7.834551844

Fig. 7. The performance results of the graph based topology analysis

Experimental results of the graph based topology analysis, **Fig. 7**, show that it clearly outperforms its matrix equivalent. A significant conclusion is that for such small graphs, there is no real need to distribute the work. The increased computing power of a cluster is largely unused and overshadowed by the overhead costs.

Spectral graph sparsification

The *sparsify* method accepts two parameters, ϵ and p. According to the sparsification theorem by Batson et al. [31], the parameters are bounded as follows: $\epsilon \in (1/n, 1/3)$ and $p \in (0, 1/2)$, where n is the number of vertices. As we had no general guideline how to choose values in order to produce optimal results, we selected uniformly a few values, shown in **Table 3**, to show how the algorithm behaves at various parts of the bounded spectrum.

Table 3 Parameters used as sparsify input

ϵ	0.1	0.15	0.2	0.25
p	0.1	0.2	0.3	0.4

The experiment procedure included parsing the graph data into a GraphX graph object and sparsifying it. We analyzed two outputs: the time it took to finish the sparsification (excluding the preparation time) and the degree of the sparsification, i.e. how many edges were left afterwards. As said above, we had two setups, with five and ten worker nodes. On each, all listed parameter pairs were used and iterated a number of times in order to obtain mean values for both outputs.

	e=0.1 p=0.1	e=0.15 p=0.2	e=0.2 p=0.3	e=0.25 p=0.4
bio-human-gene2 (5 workers)	5914.869081	5380.573035	6097.81851	6209.277437
bio-human-gene2 (10 workers)	4153.301844	3732.046778	4227.041625	4313.345095
bio-mouse-gene (5 workers)	9990.18734	9575.352944	10244.72488	10628.05124
bio-mouse-gene (10 workers)	7111.963286	6458.472137	7384.569416	7515.986213

Fig. 8. The times it took to sparsify both datasets in both environments consisting of one master node and five and ten worker nodes

The execution time results are shown in **Fig. 8**. We observe a performance gain of approximately 30% when scaling up to ten workers. Given that we kept the cluster and Spark configuration as uniform as possible during the experiments, there is still room for fine tuning, which could yield better results. The observed deviation is explained by two dominant factors. First, the random nature of the sampling procedure, and, second, the distributed cloud environment, which cannot guarantee the same conditions for every execution. It is interesting to note that the performance gained through parameter modification is rather consistent across the iterations, which indicates that the choice of parameter values provides another tuning opportunity, independent from the random process.

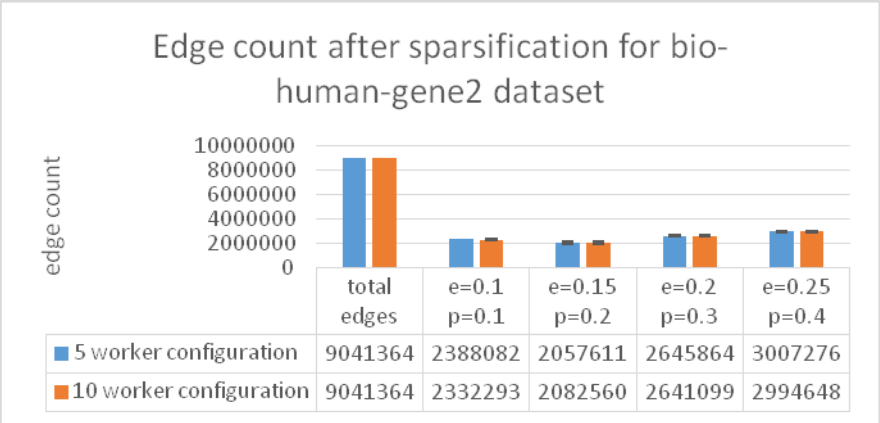


Fig. 9. The degree of sparsification, i.e. the remaining edge counts after the calculation, for the bio-human-gene2 dataset. The results are shown for some parameter combinations and compared to the edge count of the dataset before sparsification

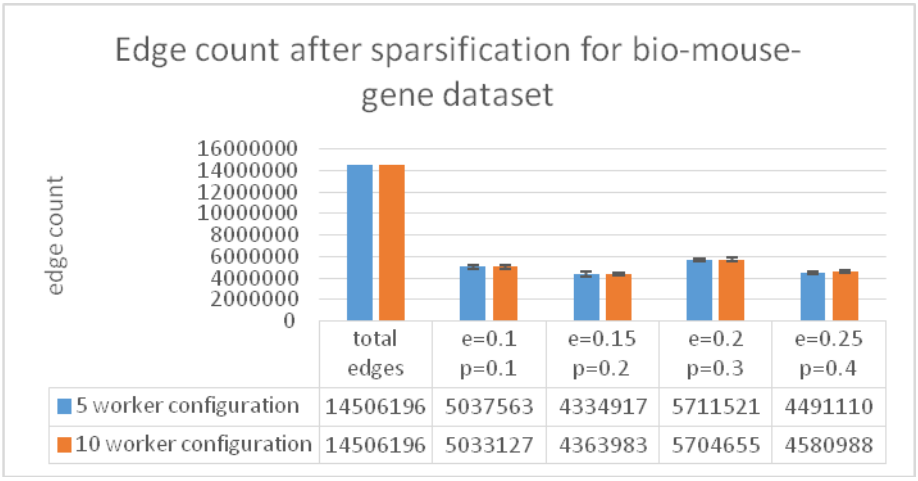


Fig. 10. The degree of sparsification, i.e. the remaining edge counts after the calculation, for the bio-mouse-gene dataset. The results are shown for some parameter combinations and compared to the edge count of the dataset before sparsification

Fig. 9 and **Fig. 10** show the achieved sparsification for each dataset. The graphs compare the edge count of the initial dataset with the edges obtained after the sparsify procedure with the same parameter variations and environments as before. First of all, note the magnitude of sparsification. In some instances, the resulting edge set is less than 30% of the original one. That means that we can get a graph, spectrally similar to the original one, with just a third of the starting edges. Again we observe a slight deviation in the resulting edge counts. This means that, all the things being the same, we can expect to get a different number of edges across multiple runs. Although that may seem

surprising, it is a consequence of the random process. Also, one should keep in mind, that the variance is negligible compared to the edges count and that the aim of the sparsification is to get an approximation of the initial graph. The choice of parameter values has a more profound impact here than on the previous time analysis. We can observe somewhat of a trend, different for both datasets, however, it indicates the existence of a minimum configuration, which would result in the most sparsification.

7. Conclusion

Our primary goal in this paper was to reflect on complex algorithms and adapt them to the requirements of the Big Data era. At the core lie mathematical problems that are generic enough to be applicable to broader spectrum of applications. We achieved this by contributing open source solutions for a few chosen problems and showing their performance under load.

Our three projects demonstrate how complex and computationally demanding solutions are applied to large datasets in a distributed and scalable environment. The operations themselves are illustrative to the broad spectrum of algorithmic approaches that can be optimized in this manner.

Underlying the power flow problem is the Newton-Raphson iterative method. The system size directly affects the number of required equations and thus the size of the problem.

The island detection problem compares two approaches. Iterative binary matrix multiplication addresses another common complex algebraic operation. The connectedness analysis is also an important tool in graph manipulations.

Spectral graph analysis is a relatively specific concept that demonstrated useful applications. It is still a complex and demanding procedure, however, we showed it could be effectively parallelized and thus made applicable to even the most complex of graphs.

The experimental results indicate that the solutions perform well on large datasets and that they easily scale.

There is room for performance improvement as well as expanding the existing toolkit with more solutions. Further research can also be directed towards combining the presented solutions to other derivative applications.

Reference

1. Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.
2. Li, Xueqi, et al. "Accelerating large-scale genomic analysis with Spark." Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on. IEEE, 2016
3. Ji, Hao, et al. "An Apache Spark implementation of block power method for computing dominant eigenvalues and eigenvectors of large-scale matrices." Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable

Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on. IEEE, 2016.
4. Van Ness, James E. "Iteration methods for digital load flow studies." Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems 78.3 (1959): 583-586.
5. Tinney, William F., and Clifford E. Hart. "Power flow solution by Newton's method." IEEE Transactions on Power Apparatus and systems 11 (1967): 1449-1460.
6. Trias, Antonio. "The holomorphic embedding load flow method." Power and Energy Society General Meeting, 2012 IEEE. IEEE, 2012.
7. Goderya, F., A. A. Metwally, and O. Mansour. "Fast detection and identification of islands in power networks." IEEE transactions on power apparatus and systems 1 (1980): 217-221.
8. Montagna, M., and G. P. Granelli. "Detection of Jacobian singularity and network islanding in power flow computations." IEE Proceedings-Generation, Transmission and Distribution 142.6 (1995): 589-594.
9. Guler, Teoman, and George Gross. "Detection of island formation and identification of causal factors under multiple line outages." IEEE Transactions on Power Systems 22.2 (2007): 505-513.
10. Stott, Brian, Ongun Alsac, and Alcir J. Monticelli. "Security analysis and optimization." Proceedings of the IEEE 75.12 (1987): 1623-1644.
11. Zimmerman, Ray Daniel, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education." IEEE Transactions on power systems 26.1 (2011): 12-19
12. Beerten, Jef, and Ronnie Belmans. "Development of an open source power flow software for high voltage direct current grids and hybrid AC/DC systems: MATACDC." IET Generation, Transmission & Distribution 9.10 (2015): 966-974.
13. Li, Hongyan, Junjie Sun, and Leigh Tesfatsion. "Dynamic LMP response under alternative price-cap and price-sensitive demand scenarios." Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE. IEEE, 2008.
14. Zhou, Michael, and Shizhao Zhou. "Internet, open-source and power system simulation." Power Engineering Society General Meeting, 2007. IEEE. IEEE, 2007.
15. https://bitbucket.org/suticd/sparkpowertools/src/master/
16. https://bitbucket.org/suticd/spectralgraphanalysistool/src/master/
17. Spielman, Daniel A., and Shang-Hua Teng. "Spectral sparsification of graphs." SIAM Journal on Computing 40.4 (2011): 981-1025.
18. Spielman, Daniel A., and Shang-Hua Teng. "A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning." SIAM Journal on Computing42.1 (2013): 1-26.
19. Spielman, Daniel A., and Shang-Hua Teng. "Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems." SIAM Journal on Matrix Analysis and Applications 35.3 (2014): 835-885.
20. Spielman, Daniel A., and Nikhil Srivastava. "Graph sparsification by effective resistances." SIAM Journal on Computing 40.6 (2011): 1913-1926.
21. Koutis, Ioannis, and Shen Chen Xu. "Simple parallel and distributed algorithms for spectral graph sparsification." ACM Transactions on Parallel Computing (TOPC) 3.2 (2016): 14.
22. Sun, He, and Luca Zanetti. "Distributed graph clustering and sparsification." ACM Transactions on Parallel Computing (TOPC) 6.3 (2019): 17.
23. Šutić, Davor, and Ervin Varga. "Spectral Graph Analysis with Apache Spark." Proceedings of the 2018 International Conference on Mathematics and Statistics. ACM, 2018.
24. Rossi, Ryan, and Nesreen Ahmed. "The network data repository with interactive graph analytics and visualization." Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.

25. Xin, Reynold S., et al. "Graphx: A resilient distributed graph system on spark." First International Workshop on Graph Data Management Experiences and Systems. ACM, 2013.
26. Zhao, Xueqian, Zhuo Feng, and Cheng Zhuo. "An efficient spectral graph sparsification approach to scalable reduction of large flip-chip power grids." Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design. IEEE Press, 2014.
27. Jancauskas, Vytautas. “Scientific Computing with Scala.” Packt Publishing Ltd, 2016
28. Hong, Yoo Pyo, and C-T. Pan. "Rank-revealing QR factorizations and the singular value decomposition." Mathematics of Computation 58.197 (1992): 213-232.
29. Goderya, F., A. A. Metwally, and O. Mansour. "Fast detection and identification of islands in power networks." IEEE transactions on power apparatus and systems 1 (1980): 217-221.
30. Bosagh Zadeh, Reza, et al. "Matrix computations and optimization in apache spark." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.
31. Batson, Joshua, et al. "Spectral sparsification of graphs: theory and algorithms." Communications of the ACM 56.8 (2013): 87-94.
32. https://issues.apache.org/jira/browse/SPARK-10335
33. Šutić, Davor, and Ervin Varga. " Appendix - Grid model", https://bitbucket.org/suticd/sparkpowercalculations/src/master/Documentation/Appendix%20-%20Grid%20Model.pdf
34. Šutić, Davor, and Ervin Varga. " Appendix - Power flow problem formulation", https://bitbucket.org/suticd/sparkpowercalculations/src/master/Documentation/Appendix%20-%20Power%20flow%20problem%20formulation.pdf
35. Perraudin, Nathanaël, Johan Paratte, David Shuman, Lionel Martin, Vassilis Kalofolias, Pierre Vandergheynst, and David K. Hammond. "GSPBOX: A toolbox for signal processing on graphs." arXiv preprint arXiv:1408.5781 (2014).

Davor Šutić was born in 1987. He holds the BSc degree in Computer Science from the School of Electrical Engineering, Belgrade, Serbia and the MSc degree from the Faculty of Technical Sciences, Novi Sad, Serbia. Currently, he is a PhD candidate at the same school.

Ervin Varga was born in Kikinda, Serbia on 19.11.1970. He graduated in 1994 and earned his BSc title in electrical engineering at the University of Novi Sad, Faculty of Technical Sciences Novi Sad, Serbia. In 1999 he finalized his master studies and earned the MSc title in computer science at the same university. Ervin defended his PhD thesis in 2007 and earned the PhD title in electrical engineering (his thesis was an application of software engineering and computer science in the domain of electrical power systems) at the same university. Ervin is a Senior Member of the IEEE.

Received: May 31, 2020; Accepted: July 15, 2021.

Computer Science and Information Systems 19(1):141–164 https://doi.org/10.2298/CSIS201220042Y

A Graph-based Feature Selection Method for Learning to Rank Using Spectral Clustering for Redundancy Minimization and Biased PageRank for Relevance Analysis*

Jen-Yuan Yeh[1,†] and Cheng-Jung Tsai[2]

[1] Dept. of Operation, Visitor Service, Collection and Information Management,
National Museum of Natural Science,
No. 1, Guanqian Rd., North Dist.,
Taichung City 404, Taiwan (R.O.C.)
jenyuan@nmns.edu.tw
[2] Graduate Institute of Statistics and Information Science,
National Changhua University of Education,
No. 1, Jinde Rd., Changhua City,
Changhua County 500, Taiwan (R.O.C.)
cjtsai@cc.ncue.edu.tw

Abstract. This paper addresses the feature selection problem in learning to rank (LTR). We propose a graph-based feature selection method, named FS-SCPR, which comprises four steps: (i) use ranking information to assess the similarity between features and construct an undirected feature similarity graph; (ii) apply spectral clustering to cluster features using eigenvectors of matrices extracted from the graph; (iii) utilize biased PageRank to assign a relevance score with respect to the ranking problem to each feature by incorporating each feature's ranking performance as preference to bias the PageRank computation; and (iv) apply optimization to select the feature from each cluster with both the highest relevance score and most information of the features in the cluster. We also develop a new LTR for information retrieval (IR) approach that first exploits FS-SCPR as a preprocessor to determine discriminative and useful features and then employs Ranking SVM to derive a ranking model with the selected features. An evaluation, conducted using the LETOR benchmark datasets, demonstrated the competitive performance of our approach compared to representative feature selection methods and state-of-the-art LTR methods.

Keywords: Feature selection, Feature similarity graph, Spectral clustering, Biased PageRank, Learning to rank, Information retrieval.

* This paper is an extended version of the ICCIP 2020 paper "Graph-based Feature Selection Method for Learning to Rank" [70].

† Corresponding author

1. Introduction

Ranking, a crucial task in information retrieval (IR), involves creating an ordered list of documents in which the relative order of documents represents their degree of relevance to the given query or their importance. In the last decade, learning to rank (LTR), which leverages machine learning to build effective ranking models, has received much attention. LTR automatically learns from the training data for tuning model parameters or by combining some features (or ranking models in context) into one more effective model [40]. Existing literature has proposed a variety of approaches, such as McRank [38], PRank [14], Ranking SVM [27][30], RankBoost [21], RankNet [8], FRank [62], AdaRank [67], SVM-MAP [72], and ListNet [9] (see Section 2.1).

As LTR algorithms incorporate more and more features, feature selection for ranking is needed because high-dimensional features tend to include irrelevant and redundant features, which can deteriorate the models' performance and make the models difficult to understand. High-dimensional features also lead to high computational costs in training and prediction. However, feature selection, which constructs and selects useful subsets of features for building a good predictor [25], reduces data dimensionality and eliminates redundant and irrelevant features. Thus, much work has been done in recent years to develop feature selection methods dedicated to LTR since the pioneering work of [22]. See Section 2.2 for an overview of feature selection methods for LTR.

We propose a graph-based feature selection method for LTR, referred to as FS-SCPR (Feature Selection Using Spectral Clustering and Biased PageRank) (see Fig. 3). We then develop a new LTR for IR approach that exploits FS-SCPR as a preprocessor to determine discriminative and useful features. This approach employs Ranking SVM [27][30] to derive a ranking model with the selected features (see Fig. 2). FS-SCPR selects a subset of features that have minimum redundancy with each other and have maximum relevance to the ranking problem. To minimize redundancy, FS-SCPR drops redundant features that are grouped in the same cluster. To maximize relevance, FS-SCPR greedily collects a representative feature with high relevance to the ranking problem from each cluster.

FS-SCPR comprises four steps. First, it uses ranking information to assess the similarity between two features and construct an undirected feature similarity graph. Second, it applies spectral clustering [44] to cluster features based on eigenvectors of matrices derived from the feature similarity graph. Then, it utilizes biased PageRank [26] to create a relevance score with respect to the ranking problem for each feature by analyzing the link structure of the feature similarity graph while incorporating each feature's ranking performance as preference to bias the PageRank computation. Finally, it applies optimization to select the feature from each cluster with both the highest relevance score and most information of the features in the cluster.

The main contributions of this paper are twofold:

1. We propose FS-SCPR, a graph-based feature selection method for LTR, to model feature relationships as a graph and leverage the graph model to select features using spectral clustering for redundancy minimization and biased PageRank for relevance analysis. In addition, we develop a new LTR for IR approach that integrates FS-SCPR and Ranking SVM.
2. We perform extensive experiments to evaluate the performance and effectiveness of the proposed approach using the LETOR benchmark datasets. The experimental results suggest that FS-SCPR helps improve the ranking performance. We

show the performance gains of the proposed approach compared to other feature selection methods and state-of-the-art LTR methods.

The remainder of this paper is structured as follows. Section 2 briefly reviews the related work. Section 3 elaborates the technical details of our LTR for IR approach, which incorporates FS-SCPR. Section 4 presents and discusses the experimental results. Finally, Section 5 concludes and points out possible directions for further work.

2. Related Work

2.1. LTR Methods

An LTR task consists of training and testing processes (see Fig. 1). Suppose that $F = \{f_1, ..., f_{|F|}\}$ is the feature set, $Q = \{q_1, ..., q_{|Q|}\}$ is the query set, and $D = \{d_1, ..., d_{|D|}\}$ is the document set. In the training process, the learning algorithm takes training data as inputs. In IR, the training data $\{(q_i, d_j), y_{i,j}\}$ comprise query-document pairs, each pair $(q_i, d_j) \in Q \times D$ is associated with a relevance label $y_{i,j}$ that indicates the relationship between q_i and d_j. Each query-document pair is modeled by a vector in an $|F|$-dimensional feature space, and each component of the vector denotes the degree of relevance of document d_j to query q_i respecting feature f_k. The training process aims to learn a ranking model (or function) f from the training data and $f(q_i, d_j)$ is assumed to assign the "true" relevance judgment for q_i and d_j. In the testing process, the model f is utilized to decide the relevance between a new query q and each document d_i in D. Then, sorting documents based on the relevance judgments constructs the document ranking list for query q.

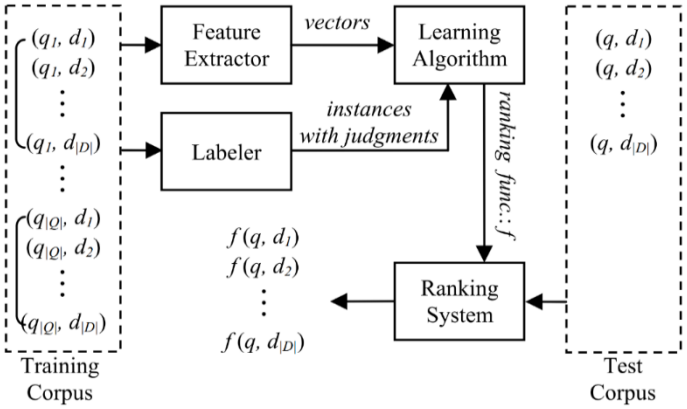


Fig. 1. Framework of LTR for IR [69]

Existing literature has explored three categories of LTR methods [40]: pointwise approaches, pairwise approaches, and listwise approaches.‡ In *pointwise* approaches, the relevance label associated with each instance (q_i, d_j) is either a class of relevance or a

‡ See [40] which provides a comprehensive survey of the literature.

relevance score (ordinal or numerical). The goal is to find a model that assigns each instance a class or a relevance score as close as possible to the instance's true class or relevance score. There are three main streams: classification-based methods (e.g., [43] and McRank [38]) and ordinal regression-based methods (e.g., [55] and PRank [14]) for dealing with classes of relevance; and regression-based methods (e.g., 13) for tackling a relevance score. The *pairwise* approach is based on learning pairwise preferences. This approach views a pair of instances, (q_i, d_j) and (q_i, d_k), as a new single instance and learns a binary classifier that can predict the preference between d_j and d_k for q_i. Example algorithms include Ranking SVM (or RankSVM for short) [27][30], RankBoost [21], RankNet [8], LambdaRank [7], and FRank [62]. The *listwise* approach takes the document ranking lists as instances and builds a model that can directly produce the ordered list (or permutation) of the documents according to a score assigned to every document. Most methods of this type focus on the direct optimization of ranking performance (e.g., AdaRank [67], SVM-MAP [72], SoftRank [61], and PermuRank [68]) or on permutations count (e.g., ListNet [9], ListMLE [66], RankCosine [52], and BoltzRank [63]).

2.2. Feature Selection Methods for LTR

Three general categories of feature selection methods for LTR are filter, wrapper, and embedded approaches. A *filter* approach performs feature ranking based on a relevance criterion. As a preprocessing step, it selects subsets of features independently of the chosen LTR algorithm. Feature selection for ranking was pioneered in [22], which addressed a multi-objective optimization problem in greedily finding a feature subset with minimum total similarity scores and maximum total importance scores. Two method variants, GAS-E and GAS-L, were proposed that utilize performance measures and loss functions in ranking, respectively, to assess feature importance. A hierarchical feature selection strategy was developed in [28] by which clusters of features are constructed and the best performing feature is selected from each cluster. RankFilter [71] extends Relief [32] to compute feature weights from multi-level relevance judgments. In [24], the authors selected the subset of features according to their expected divergence over relevance classes and their importance derived from evaluation scores. The work in [42] exploited greedy result diversification techniques, including maximal marginal relevance (MMR), max-sum dispersion (MSD), and modern portfolio theory (MPT). In [56], the subset of features was selected via minimum redundancy maximum relevance (mRMR) based on their importance and similarity. [23] devised several algorithms, including NGAS that greedily selects the subset of features by minimizing similarity and maximizing relevance, XGAS (an extension of NGAS) that considers more features at each selection iteration, and HCAS that selects the feature with the largest relevance score from each feature cluster, built through hierarchical clustering. In [49], an architecture-agnostic neural feature selection approach was proposed based on a neural LTR model. The approach consists of neural model training, feature group mining based on saliency map, and feature selection based on hierarchical clustering.

With an LTR algorithm as a "black box," a *wrapper* approach scores subsets of features according to their ranking performance. In [28], the authors proposed a

hierarchical feature selection strategy that builds feature clusters with a linear ranking model trained per cluster to select the feature of the highest model weight. Methods using boosted regression trees were explored in [47], including two greedy approaches (selecting the features with the highest relative importance as computed by boosted trees and discounting importance by feature similarity) and a randomized approach with feature-importance-based backward elimination. RankWrapper [71] extends Relief [32] to compute the feature weights from relative orderings. The best first search was used in [15] to greedily partition features into subsets and coordinate ascent was then used to combine features in each subset into one single feature. Greedy RankRLS [45] selects the feature subset of the maximal ranking performance for RankRLS [46] based on greedy forward selection and leave-query-out cross-validation. In [39], language modeling smoothing approaches with different parameters were proposed for selecting the ranking features. [16] considered a multi-objective Pareto-efficient method that optimizes both risk-sensitive evaluation and ranking performance. MOFSRank [11] is a multi-objective evolutionary algorithm consisting of an instance selection strategy, a multi-objective feature selection algorithm, and an ensemble strategy. [17] adopted forward stepwise selection and chose Akaike's information criterion [1] to decide which feature to be added to the selected subset. In [4], a subset of features was viewed as a state in the search space, and simulated annealing was utilized to find the best subset of features.

In an *embedded* approach, the feature selection procedure is integrated into the LTR algorithm. SuperSelRank [33] is a general framework for sparse LTR based on a hierarchical Bayesian model. RSRank [59] performs ℓ_1 regularization using truncated gradient descent to achieve sparsity in ranking models. FenchelRank [34], a primal-dual algorithm for sparse LTR, minimizes the ℓ_1 regularized pairwise ranking loss while simultaneously conducting model selection. SparseRank [35] is a gradient descent algorithm for minimizing the ranking errors with the ℓ_1 regularization. FSMRank [36] is a one-stage method for solving a joint convex optimization problem in which the ranking errors are minimized and meanwhile feature selection is conducted. A general framework using SVM (support vector machines) with sparse regularizations to handle nonconvex penalties was presented in [37]. EGRank [18] uses exponentiated gradient updates to solve a convex optimization problem on a sparsity-promoting ℓ_1 constraint and a pairwise ranking loss. In [53], a deep neural LTR model was provided. The authors used group ℓ_1 regularization to optimize the weights of a neural network, select the relevant features with active neurons at the input layer, and remove inactive neurons from hidden layers. The work in [19] incorporated the ℓ_1 regularized sparse term into the cost-sensitive ListMLE model proposed in [41], and an efficient proximal gradient descent learning method with adaptive Lipschitz constant was applied to obtain the global optimal parameters of the model.

Feature extraction is another technique for dimension reduction. In contrast to feature selection which selects a subset of the original features, feature extraction creates a small set of new features to represent the input data by merging or transforming the original features. LifeRank [48], for instance, views the input dataset as a matrix and constructs a new low-rank dataset with the projection of a transformation matrix that is optimized for the original dataset by minimizing the pairwise ranking loss. More examples of feature extraction methods for LTR can refer to [2] and [20].

3. Proposed Method

FS-SCPR identifies a subset of features that can accurately represent the data, reduce the complexity of the feature space, and enhance performance in ranking problems. This study develops a new LTR for IR approach by extending the framework of LTR for IR in Fig. 1 with the proposed feature selection method, FS-SCPR. See Fig. 2.

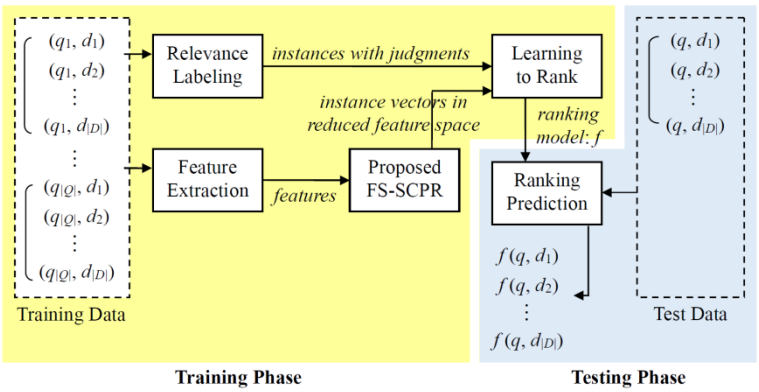


Fig. 2. The proposed LTR for IR approach that incorporates FS-SCPR

3.1. Relevance Labeling

Relevance labeling, which is often done by human annotators, assigns each instance a proper relevance judgment, which plays the role of answers (or observations) that guide the learning algorithm to learn an effective ranking model. Possible relevance judgments include (1) a class; (2) an ordinal rating; (3) a ranking order; and (4) a relevance score [69]. For the labeling scheme, this study adopts an n-star rating. To be specific, each relevance label $y_{i,j} \in \{0, 1, \ldots, n-1\}$, 0 indicates not relevant, $n-1$ means definitely relevant, and higher $y_{i,j}$ indicates higher relevance.

3.2. Feature Extraction

Feature extraction transforms the data into numerical values of ad hoc features. Let fv_k be the feature extraction function for feature f_k, and $w_{i,j,k} = \text{fv}_k(q_i, d_j)$ denote the degree of relevance of document d_j to query q_i respecting feature f_k. The value of $w_{i,j,k}$ is normalized via query-level min-max normalization as follows:

$$w_{i,j,k} = \frac{\text{fv}_k(q_i, d_j) - \min\{\text{fv}_k(q_i, d_l)\}}{\max\{\text{fv}_k(q_i, d_l)\} - \min\{\text{fv}_k(q_i, d_l)\}}, \tag{1}$$

where all $d_l \in D$, $\min\{\cdot\}$ and $\max\{\cdot\}$ respectively stand for the minimum and maximum values of $\text{fv}_k(q_i, d_l)$.

The extracted features in this study cover low-level content features (e.g., the occurrences of a query term in a document and the document length), high-level content features (e.g., BM25 [54] and LMIR [73]), and other features (e.g., the number of out- or in-links of a webpage and the PageRank centrality [6] of a webpage). See Section 4.1.

3.3. Proposed Feature Selection Method, FS-SCPR

The proposed feature selection method, FS-SCPR, is a filter approach. It targets at selecting a subset of features that have minimum redundancy with each other and have maximum relevance to the ranking problem. To minimize redundancy, FS-SCPR drops redundant features, which are grouped into the same cluster. To maximize relevance, FS-SCPR greedily collects a representative feature with high relevance to the ranking problem from each cluster. To produce a feature subset F^* ($F^* \subseteq F$), the process flow of FS-SCPR (see Fig. 3) involves the steps below.

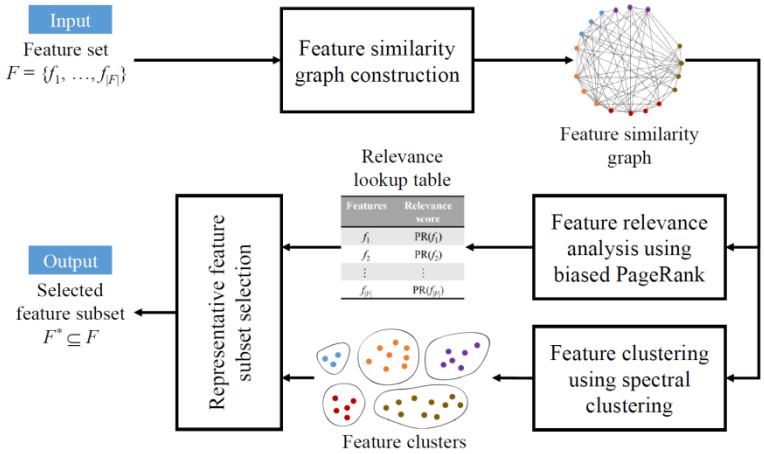


Fig. 3. Process flow of FS-SCPR

1. *Feature similarity graph construction*. Features are modeled as an undirected feature similarity graph. A vertex refers to a feature and an edge indicates that the corresponding features relate to each other. The pairwise feature similarity is measured relying on the correlation of two features' ranking results.
2. *Feature clustering using spectral clustering*. To find redundant features, similar features are grouped into clusters. This study applies spectral clustering [44] that groups data based on eigenvectors of matrices extracted from the feature similarity graph.
3. *Feature relevance analysis using biased PageRank*. The PageRank [6] centrality is utilized to capture the relative "importance" of features by analyzing the link structure of the feature similarity graph. This study further incorporates the ranking performance of each feature as preference to bias the

PageRank computation, giving each feature a more accurate relevance score with respect to the ranking problem.

4. *Representative feature subset selection*. Representative features are selected from each cluster to form the feature subset F^*. For each cluster, this study selects the feature that not only has the highest relevance score but also contains most information of the features in the cluster.

With the feature subset F^* comprising an $|F^*|$-dimensional space, every query-document pair (q_i, d_j) is depicted as a vector in the *reduced* feature space. Every component of the vector is obtained using Eq. (1).

Feature Similarity Graph Construction

Given a query q, there are $|F|$ document ranking lists $\{R_{q,1}, \ldots, R_{q,|F|}\}$. Here, $R_{q,i}$ is established by sorting (in descending order) the retrieved documents D_q according to their feature values regarding feature f_i. Widely-used non-parametric measures of ordinal association, e.g., Spearman's *rho* (ρ) [57] and Kendall's *tau* (τ) [31], can assess the degree of correlation (or similarity) between two ranking lists. This study refers to the correlation between two document ranking lists as the similarity between two features with respect to the given query.

This study chooses Kendall's τ. For two document ranking lists $R_{q,i}$ and $R_{q,j}$, the Kendall's τ value is computed as

$$\tau(R_{q,i}, R_{q,j}) = \frac{|\{(d_s, d_t) \mid d_s \prec_{R_{q,i}} d_t \text{ and } d_s \prec_{R_{q,j}} d_t\}|}{|\{(d_s, d_t)\}|}, \quad \textbf{(2)}$$

where d_s, $d_t \in D_q$, (d_s, d_t) represents a document pair, $d_s \prec_{R_{q,i}} d_t$ denotes that d_t is ranked ahead of d_s in $R_{q,i}$. For a set of queries, the overall similarity between features f_i and f_j is defined in Eq. (3) as the average of their Kendall's τ values for all the queries:

$$sim(f_i, f_j) = \frac{1}{|Q|} \sum_{q \in Q} \tau(R_{q,i}, R_{q,j}). \quad \textbf{(3)}$$

Given the pairwise similarities between features, this study thus represents features as an undirected similarity graph $G = (V, E)$. A vertex denotes a feature, i.e., $V = \{f_1, ..., f_{|F|}\}$, and $E \subseteq V \times V$. Two vertices f_i and f_j are connected if $sim(f_i, f_j) \geq \sigma$[§], and the edge weight is given by $sim(f_i, f_j)$. The graph G can be represented by an adjacency matrix $W = [w_{i,j}]_{i,j=1, \ldots, |F|}$, and each element $w_{i,j}$ is denoted by

$$w_{i,j} = \begin{cases} sim(f_i, f_j) & \text{if } i \neq j \text{ and } sim(f_i, f_j) \geq \sigma \\ 0 & \text{otherwise} \end{cases}. \quad \textbf{(4)}$$

Note that the matrix W is symmetric since $w_{i,j} = w_{j,i}$ holds.

[§] This study empirically sets σ to 0.1.

Feature Clustering Using Spectral Clustering

To cluster features into k subsets, this study applies the normalized spectral clustering algorithm in [44]. The algorithm uses k eigenvectors of a normalized graph Laplacian simultaneously for spectral graph partitioning, as the eigenvectors carry clustering information. By the spectral graph theory [12], the normalized graph Laplacian matrix L is formulated as

$$L = A^{-1/2}(A-W)A^{-1/2} = I - A^{-1/2}WA^{-1/2}, \quad (5)$$

where W is the aforementioned adjacency matrix, I is the unit matrix, and A = diag(a_1, …, $a_{|F|}$) is the diagonal matrix whose every diagonal element $a_i = \sum_j^{|F|} w_{i,j}$. Since both matrices W and A are symmetric real matrices, L is also symmetric and real. Additionally, L has $|F|$ eigenvalues, λ_1, …, $\lambda_{|F|}$, and $0 = \lambda_1 \leq ... \leq \lambda_{|F|}$.

In our problem, the inputs contain the feature set F accompanying the matrix W and the number k of clusters to build. The output is the set of clusters of features {C_1, …, C_k}. Algorithm 1 states the steps of the normalized spectral clustering algorithm.

Algorithm 1 Normalized Spectral Clustering [44]

Input: The feature set F = {f_1, …, $f_{|F|}$} accompanying the matrix W and the number k of clusters to build.

Output: The feature clusters {C_1, …, C_k}. $\forall f_i, \exists j$ s.t. $f_i \in C_j$.

Procedure:

1. Compute the matrix $L = I - A^{-1/2}WA^{-1/2}$.
2. Build the matrix $X = [x_1\ x_2\ \ldots\ x_k] \in \mathbb{R}^{|F|\times k}$ whose columns x_1, …, x_k are the k smallest eigenvectors of L.
3. Build from X the matrix $Y \in \mathbb{R}^{|F|\times k}$ whose every element $y_{i,j} = \frac{x_{i,j}}{\sqrt{\sum_j x_{i,j}^2}}$.
4. Let Y's every row be a data point in $\mathbb{R}^k$, and build k clusters via bisecting K-means.
5. Assign feature f_i to cluster C_j if Y's row i is in cluster C_j.

There are two points to note. First, [44] constructs the matrix $I - L$ in Step 1, which only changes the eigenvalues (from λ_i to $1 - \lambda_i$) and not the eigenvectors. Thus, in Step 2, [44] finds the k largest eigenvectors (referring to the k largest eigenvalues), we instead consider the k smallest eigenvectors (referring to the k smallest eigenvalues). Second, [44] uses K-means in Step 4. This study utilizes bisecting K-means [58] because it in practice produces better-quality clustering results (see [60]).

The trick of spectral clustering is to embed the data in a low-dimensional space wherein the data's cluster properties become prominent. The method's success is mainly owing to that no assumptions are made on the form of the clusters and their statistics [64] (as opposed to, for example, K-means, where the clusters are convex sets). Thus, spectral clustering very often outperforms conventional clustering algorithms. Additionally, spectral clustering is simple to implement, can be solved efficiently by standard linear algebra software, is efficient to obtain near-optimal partitions, and is reasonably fast for large sparse data sets [64]. Furthermore, spectral clustering does not necessarily need the data in the embedded form (i.e., featured objects) [65]. The data can be represented as relationships between objects, as in this work features are

modeled by a feature similarity graph. For these reasons, we choose spectral clustering to obtain feature clusters instead of conventional clustering methods.

Feature Relevance Analysis Using Biased PageRank

This study assesses feature relevance to the ranking problem via biased PageRank [26]. Given a feature similarity graph $G = (V, E)$, each vertex (i.e., feature in this context) is scored by applying biased PageRank on the graph. The score s for a vertex f_i is assigned by the recursive equation

$$s(f_i) = (1-\alpha) \times p(f_i) + \alpha \times \sum_{f_j \in M(f_i)} \frac{w_{i,j}}{\sum_{f_k \in M(f_j)} w_{j,k}} \times s(f_j), \quad (\mathbf{6})$$

where α is a damping factor ($0 \le \alpha < 1$)[**], $p(f_i)$ is the preference weight[††] assigned to vertex f_i, and $M(f_i)$ is the set of those vertices that have links to the vertex f_i.

The feature relevance analysis approach iterates until convergence is achieved. When iterations stop, a score is associated with every vertex as its feature relevance. In each iteration, function $p(\cdot)$ introduces additional preferences to the appropriate vertices. By selecting the appropriate preference weights, the PageRank computation can be made to prefer certain vertices. This study assigns larger preference weights to those features that have better ranking performance. The idea is to incorporate ranking performance into the biased PageRank to better capture feature relevance. This study uses MAP (see Section 4.2 for its definition) for $p(\cdot)$. Note that the preference weights of features are normalized by $\frac{p(f_i)}{\sum_j p(f_j)}$ before they are used in Eq. (6).

Representative Feature Subset Selection

After feature clustering, redundant features are grouped into the same cluster. Additionally, each feature is scored for its relevance to the ranking problem after feature relevance analysis. This study selects one representative feature from each cluster, according to which feature that not only has the highest relevance score but also contains most information of the features in the cluster. All the other features in the cluster are discarded. Thus, the resulting feature subset contains features that have minimum redundancy with each other and have maximum relevance to the ranking problem.

Algorithm 2 depicts the steps of our feature subset selection approach. In Step 2.1, this study measures the similarity between features using the matrix Y in Algorithm 1. To determine which feature in a cluster has the highest relevance score and contains most information of the other features, we deal with the multi-objective problem using a linear combination of a feature's relevance score and its sum of pairwise similarities to the other features, as shown in Step 2.2.

[**] This study sets α to 0.85 according to [6].

[††] In the original PageRank [6], each vertex is weighted with an equal preference of $1/|V|$.

Algorithm 2 Representative Feature Subset Selection

Input: The feature clusters $\{C_1, \ldots, C_k\}$.
Output: The selected feature subset F^*.
Procedure:

1. Set F^* to an empty set, $F^* = \varnothing$.
2. For each cluster C_i, do:
   2.1. For each feature f in C_i, compute *SSim*(f), i.e., the sum of its pairwise similarities to the other features in C_i.
   2.2. From C_i, find feature f that has not only the largest *SSim*(f) but also the highest feature relevance, as scored by Eq. (6). That is, $f = \arg\max\left[0.5\times s(f) + 0.5\times \frac{SSim(f)}{|C_i|-1}\right]$. Note that *SSim*($f$) is normalized by $|C_i| - 1$.
   2.3. Assign feature f to F^*, i.e., $F^* = F^* \bigcup \{f\}$.

3.4. Ranking Model Learning and Prediction

This study employs Ranking SVM [27][30] to derive a ranking model since previous studies have demonstrated its feasibility and effectiveness. Ranking SVM views the LTR problem as binary classification on pairs of documents and applies SVM (support vector machines) to solve the classification problem. In other words, Ranking SVM targets binary ordering relations between documents with respect to queries and learns, based on parts of the observations of the target (or optimal) ranking lists, a model that minimizes the count of discordant pairs. Considering the class of *linear* ranking functions, the following optimization problem is solved in Ranking SVM [30]:

$$\begin{aligned} &minimize: \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum \xi_{i,j,q} \\ &subject\ to: \\ &\quad \forall q, \forall (d_i, d_j) \in r_q^* : \vec{w}\Phi(q, d_i) \geq \vec{w}\Phi(q, d_j) + 1 - \xi_{i,j,q} \\ &\quad \forall i \forall j \forall q : \xi_{i,j,q} \geq 0 \end{aligned} \quad (7)$$

Here, the weight vector $\vec{w}$ is arranged in learning; C trades-off between margin and training error; $\xi_{i,j,q}$ is a non-negative slack variable; r_q^* is the target ranking list, given query q; and $\Phi(q, d_i)$ is a feature vector that depicts the relevance of document d_i to query q in terms of features.

For all the queries, pairs of instances and their relative preferences are inputted into Ranking SVM for training. Note that each instance is modeled as a vector in the reduced feature space (see Section 3.3). Regarding ranking prediction, the learned ranking model decides for a new query whether pairs of documents are in concordant order. The final document ranking list can thus be established according to the outputted binary ordering relations between documents.

4. Evaluation

4.1. Datasets

To evaluate the performance and effectiveness of the proposed LTR for IR approach, we conducted experiments on the publicly available LETOR[‡‡] benchmark collections. We selected the following four datasets: HP2004, NP2004, OHSUMED, and MQ2008. The first three datasets are from LETOR 3.0 and the last one is from LETOR 4.0. The datasets come as query-document pairs. A pair contains a feature vector and its relevance judgment. For cross-validation, each dataset is split into five subsets. In each fold, three subsets are used for learning, one subset for validation, and the other one for testing. See [51] and [50] for details on the selection of document corpora, the sampling of documents, the extraction of features and meta-information, and the finalization of datasets. Table 1 depicts the statistics of the datasets. Table 2 illustrates some sample data; each row stands for a query-document pair.

Table 1. Statistics of the datasets. For HP2004 and NP2004, the relevance judgments are on two levels (relevant and not relevant); for OHSUMED and MQ2008, the relevance judgments are on three levels (definitely relevant, possibly relevant, and not relevant)

	HP2004	NP2004	OHSUMED	MQ2008
No. of queries	75	75	106	784
No. of query-document pairs (i.e., instances)	74,409	73,834	16,140	15,211
No. of features	64	64	45	46
Relevance levels	2	2	3	3

Table 2. Sample data excerpted from MQ2008

Label	Query	f_1	…	f_{46}	Note
2	qid:10032	1:0.056537	…	46:0.076923	#doc: GX029-35-5894638
0	qid:10032	1:0.279152	…	46:1.000000	#doc: GX030-77-6315042
0	qid:10032	1:0.130742	…	46:1.000000	#doc: GX140-98-13566007
1	qid:10032	1:0.593640	…	46:0.000000	#doc: GX256-43-0740276
⋮	⋮	⋮	⋮	⋮	⋮

4.2. Evaluation Measures

We use two common measures, namely, MAP (mean average precision) [5] and NDCG (normalized discounted cumulative gain) [29], as the evaluation measures.

[‡‡] https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/.

Eq. (8) denotes the average precision (AvgP) for a query, and for all the queries the mean of their average precisions is the MAP.

$$\text{AvgP} = \frac{\sum_{n=1}^{N} \text{P@}n \times rel(n)}{\text{\# of relevant documents for the query}}, \tag{8}$$

In the equation, N is the number of retrieved documents, P@n (namely, precision at position n) is the fraction of relevant documents among the top n results, and $rel(n) \in \{0, 1\}$ implies that the document at position n is relevant or not.

For a query's ranking list, the NDCG at position n is calculated by

$$\text{NDCG@}n = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{\log_2(1+j)}, \tag{9}$$

in which Z_n is a normalization parameter that allows producing an NDCG of 1.0 for the perfect list, and $r(j)$ means the rating of the document at position j. The NDCG@n values for all queries are averaged and reported.

We present the results of NDCG@1, NDCG@3, NDCG@5, NDCG@10, and MAP for comparisons.

4.3. Experimental Setup

We conducted experiments to verify whether FS-SCPR helps improve the ranking performance and to understand whether FS-SCPR outperforms other baseline feature selection methods and state-of-the-art LTR approaches. Five-fold cross-validation is conducted, and all the presented results are the average performance on the testing set. In each fold, we use the training set to select features, and train a ranking model from the training set with the selected features. The validation set is utilized for parameter tuning and model selection. The above two steps are repeated to identify the best ranking model. Then, the obtained ranking model is evaluated on the testing set.

For simplicity, we denote the proposed approach as FS-SCPR and use "feature selection" and "feature selection for LTR" interchangeably for the remainder of this paper. Additionally, for efficient learning, we use RankSVM-Primal [10] (an efficient version of Ranking SVM) instead of Ranking SVM.

4.4. Baseline Algorithms

We tested two groups of baseline algorithms.[§§] The first group tested LTR methods without using feature selection. This study selects AdaRank-MAP (a listwise method) [67], RankSVM-Primal (a pairwise method) [10], ListNet (a listwise method) [9], and RankBoost (a pairwise method) [21]. AdaRank-MAP, RankSVM-Primal, and ListNet learn linear ranking models, while RankBoost learns a non-linear ranking model.

The second group tested feature selection methods, including GAS-E (a filter method)

[§§] The presented results of the baselines are cited from the LETOR datasets and the original papers.

[22], FSMSVM (a wrapper method) [36], and FSMRank (an embedded method) [36]. See Section 2.2 for a brief description of these methods. As the proposed approach adopts RankSVM-Primal as the learning algorithm, the implementation of GAS-E in this work also chooses RankSVM-Primal (instead of Ranking SVM or RankNet used in [22]). As a simple feature selection method, FSMSVM selects top features with large weights according to their weights in a pre-trained model (which in [36] is learned by FSMRank) and uses the selected features to learn a ranking model by RankSVM-Primal.

4.5. Results

Comparison with RankSVM-Primal

This experiment compares the ranking performance of FS-SCPR with RankSVM-Primal. FS-SCPR considers only the selected features, while RankSVM-Primal uses all the features. The objective of this experiment is to empirically justify whether the proposed feature selection method helps enhance the performance of ranking predictions. Tables 3–6 present the results on four datasets. The row named "Imp." in each table denotes the relative improvement[***] of FS-SCPR versus RankSVM-Primal.

Table 3. Ranking performance of FS-SCPR and RankSVM-Primal on HP2004 (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.6337**	**0.7590**	**0.7893**	**0.8179**	**0.7216**
RankSVM-Primal	0.5733	0.7129	0.7528	0.7720	0.6712
Imp.	+10.54%	+6.47%	+4.85%	+5.95%	+7.51%

Table 4. Ranking performance of FS-SCPR and RankSVM-Primal on NP2004 (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.5527	**0.7603**	**0.7842**	**0.8159**	**0.6809**
RankSVM-Primal	**0.5600**	0.7236	0.7719	0.7950	0.6755
Imp.	–1.3%	+5.07%	+1.59%	+2.63%	+0.8%

Table 5. Ranking performance of FS-SCPR and RankSVM-Primal on OHSUMED (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.5459	**0.4959**	**0.4785**	**0.4584**	**0.4491**
RankSVM-Primal	**0.5460**	0.4855	0.4689	0.4504	0.4446
Imp.	–0.02%	+2.14%	+2.05%	+1.78%	+1.01%

[***] When b is compared to a, the relative improvement is calculated as $(b - a) / a \times 100\%$.

Table 6. Ranking performance of FS-SCPR and RankSVM-Primal on MQ2008 (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.3692	**0.4375**	**0.4770**	**0.2318**	**0.4776**
RankSVM-Primal	**0.3725**	0.4333	0.4765	0.2309	0.4744
Imp.	–0.89%	+0.97%	+0.1%	0.39%	+0.67%

The results of FS-SCPR are significantly boosted by the proposed feature selection method. Looking at NDCG@10, the performance of FS-SCPR is enhanced by 5.95% on HP2004, by 2.63% on NP2004, by 1.78% on OHSUMED, and by 0.39% on MQ2008. In terms of MAP, FS-SCPR has relative increases of 7.51% on HP2004, 0.8% on NP2004, 1.01% on OHSUMED, and 0.67% on MQ2008. Similar enhancements can be seen in other measures. For each dataset, the maximum enhancements over distinct measures are increases of 10.54% in NDCG@1 on HP2004, 5.07% in NDCG@3 on NP2004, 2.14% in NDCG@3 on OHSUMED, and 0.97% in NDCG@3 on MQ2008. However, there are few exceptions, including the NDCG@1 scores on NP2004, OHSUMED, and MQ2008. In these cases, the performance of FS-SCPR deteriorated by 1.3%, 0.02%, and 0.89%, respectively, compared to the performance of RankSVM-Primal.

Comparison with Feature Selection Methods

This experiment focuses on understanding how effectively FS-SCPR performs compared to other feature selection methods. Tables 7–10 present the comparison results. In each column, the methods are ranked by their scores, and the rankings are shown in parentheses.

First, FS-SCPR is observed in most cases to have superior performance to GAS-E (a filter method) and FSMSVM (a wrapper method). The few exceptions are the cases of NDCG@1 on NP2004 and OHSUMED and the case of NDCG@5 on MQ2008. Taking NDCG@10 as an example, FS-SCPR outperforms GAS-E by 4.2%, 2.53%, 1.82%, and 1.22% on HP2004, NP2004, OHSUMED, and MQ2008, respectively. Compared to FSMSVM, FS-SCPR has performance gains of 3.81%, 1.23%, 3.08%, and 2.98% in NDCG@10 on HP2004, NP2004, OHSUMED, and MQ2008, respectively. Regarding MAP, FS-SCPR outperforms GAS-E by 4.2%, 1.08%, 0.36%, and 0.19% on HP2004, NP2004, OHSUMED, and MQ2008, respectively. Compared to FSMSVM, FS-SCPR has performance gains of 2.88%, 0.9%, 1.13%, and 0.67% in MAP on HP2004, NP2004, OHSUMED, and MQ2008, respectively.

Second, compared to FSMRank (an embedded method), FS-SCPR performs competitively only in a few cases. For instance, it outperforms FSMRank in MAP on HP2004 and MQ2008 with increases of 0.15% and 0.1%, respectively. As another example, its NDCG@1 scores on the four datasets are superior to those of FSMRank with increases of 3.33% on HP2004, 1.1% on NP2004, 1.21% on OHSUMED, and 0.16% on MQ2008. The comparison results in most cases demonstrate that FS-SCPR does not perform better than FSMRank, especially in NDCG@3, NDCG@5, and NDCG@10. These observations are not unexpected since an embedded method that

conducts feature selection inside the LTR algorithm generally tends to have superior performance to a filter method that selects features independently of the LTR algorithm.

Finally, from an overall perspective, the experimental results show that FS-SCPR practically performs well. In terms of MAP, FS-SCPR ranks first on HP2004 and MQ2008 and ranks second on NP2004 and OHSUMED. Considering NDCG@10, FS-SCPR is the second best performer on the four datasets. To further identify which method demonstrates the best results in various measures on different datasets, Table 11 presents a unified ranking of the methods. According to [3], the unified rank of a method is defined by

$$Rank = \sum_{r=1}^{M} \frac{(M - r + 1) \times R_r}{M}, \tag{10}$$

in which M is the number of compared methods and R_r is the count the method appears in the r-th rank. From Table 11, a unified ranking of the methods is obtained: FSMRank $\succ$ FS-SCPR $\succ$ GAS-E $\succ$ FSMSVM, in which the proposed FS-SCPR ranks second.

Table 7. Ranking performance of FS-SCPR and other feature selection for LTR methods on HP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.6337** (1)	*0.7590* (2)	*0.7893* (2)	*0.8179* (2)	**0.7216** (1)
GAS-E	0.6133 (3)	0.7280 (3)	0.7679 (3)	0.7849 (4)	0.6925 (4)
FSMSVM	*0.6267* (2)	0.7136 (4)	0.7635 (4)	0.7879 (3)	0.7014 (3)
FSMRank	0.6133 (3)	**0.8070** (1)	**0.8187** (1)	**0.8383** (1)	*0.7205* (2)

Table 8. Ranking performance of FS-SCPR and other feature selection for LTR methods on NP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	*0.5527* (2)	*0.7603* (2)	*0.7842* (2)	*0.8159* (2)	*0.6809* (2)
GAS-E	**0.5600** (1)	0.7236 (4)	0.7617 (4)	0.7958 (4)	0.6736 (4)
FSMSVM	0.5467 (3)	0.7538 (3)	0.7830 (3)	0.8060 (3)	0.6748 (3)
FSMRank	0.5467 (3)	**0.7784** (1)	**0.8000** (1)	**0.8279** (1)	**0.6837** (1)

Table 9. Ranking performance of FS-SCPR and other feature selection for LTR methods on OHSUMED (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.5459 (3)	*0.4959* (2)	*0.4785* (2)	*0.4584* (2)	*0.4491* (2)
GAS-E	**0.5547** (1)	0.4794 (3)	0.4720 (3)	0.4502 (3)	0.4475 (3)
FSMSVM	*0.5492* (2)	0.4690 (4)	0.4640 (4)	0.4447 (4)	0.4441 (4)
FSMRank	0.5394 (4)	**0.5013** (1)	**0.4824** (1)	**0.4613** (1)	**0.4498** (1)

Table 10. Ranking performance of FS-SCPR and other feature selection for LTR methods on MQ2008 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.3692** (1)	*0.4375* (2)	0.4770 (3)	*0.2318* (2)	**0.4776** (1)
GAS-E	0.3601 (4)	0.4345 (3)	*0.4772* (2)	0.2290 (3)	0.4767 (3)
FSMSVM	0.3652 (3)	0.4278 (4)	0.4701 (4)	0.2251 (4)	0.4744 (4)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FSMRank	*0.3686* (2)	**0.4399** (1)	**0.4791** (1)	**0.2327** (1)	*0.4771* (2)

Table 11. Unified ranking of methods (the higher Rank value, the better)

	$R_r = 1$	$R_r = 2$	$R_r = 3$	$R_r = 4$	Rank
FS-SCPR	4	14	2	0	15.5
GAS-E	2	1	10	7	9.5
FSMSVM	0	2	8	10	8
FSMRank	14	3	2	1	17.5

Comparison with State-of-the-Art LTR Methods

This experiment compares the ranking performance of FS-SCPR with other state-of-the-art LTR methods (all without using feature selection). Tables 12–15 present the results. In each column, the methods are ranked by their scores, and the rankings are shown in parentheses.

Table 12. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on HP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.6337** (1)	*0.7590* (2)	*0.7893* (2)	*0.8179* (2)	*0.7216* (2)
AdaRank-MAP	*0.6133* (2)	**0.8164** (1)	**0.8277** (1)	**0.8328** (1)	**0.7219** (1)
RankSVM-Primal	0.5733 (4)	0.7129 (4)	0.7528 (4)	0.7720 (4)	0.6712 (4)
ListNet	0.6000 (3)	0.7213 (3)	0.7694 (3)	0.7845 (3)	0.6899 (3)
RankBoost	0.5067 (5)	0.6989 (5)	0.7211 (5)	0.7428 (5)	0.6251 (5)

Table 13. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on NP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	*0.5527* (2)	**0.7603** (1)	*0.7842* (2)	**0.8159** (1)	**0.6809** (1)
AdaRank-MAP	0.4800 (4)	0.6979 (4)	0.7310 (4)	0.7497 (4)	0.6220 (4)
RankSVM-Primal	**0.5600** (1)	0.7236 (3)	0.7719 (3)	0.7950 (3)	*0.6755* (2)
ListNet	0.5333 (3)	*0.7587* (2)	**0.7965** (1)	*0.8128* (2)	0.6720 (3)
RankBoost	0.4267 (5)	0.6274 (5)	0.6512 (5)	0.6914 (5)	0.5640 (5)

Table 14. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on OHSUMED (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	*0.5459* (2)	**0.4959** (1)	**0.4785** (1)	**0.4584** (1)	**0.4491** (1)
AdaRank-MAP	0.5388 (3)	0.4682 (4)	0.4613 (3)	0.4429 (3)	*0.4487* (2)
RankSVM-Primal	**0.5460** (1)	*0.4855* (2)	*0.4689* (2)	*0.4504* (2)	0.4446 (4)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
ListNet	0.5326 (4)	0.4732 (3)	0.4432 (5)	0.4410 (4)	0.4457 (3)
RankBoost	0.4632 (5)	0.4555 (5)	0.4494 (4)	0.4302 (5)	0.4411 (5)

Table 15. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on MQ2008 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.3692 (5)	**0.4375** (1)	*0.4770* (2)	**0.2318** (1)	**0.4776** (1)
AdaRank-MAP	*0.3754* (2)	*0.4370* (2)	**0.4794** (1)	0.2288 (4)	0.4764 (4)
RankSVM-Primal	0.3725 (4)	0.4333 (3)	0.4765 (3)	*0.2309* (2)	0.4744 (5)
ListNet	*0.3754* (2)	0.4324 (4)	0.4747 (4)	0.2303 (3)	*0.4775* (2)
RankBoost	**0.3856** (1)	0.4288 (5)	0.4666 (5)	0.2255 (5)	*0.4775* (2)

Table 16. Unified ranking of methods (the higher Rank value, the better)

	$R_r = 1$	$R_r = 2$	$R_r = 3$	$R_r = 4$	$R_r = 5$	Rank
FS-SCPR	11	8	0	0	1	17.6
AdaRank-MAP	5	4	3	8	0	13.2
RankSVM-Primal	2	5	5	7	1	12
ListNet	1	4	10	4	1	12
RankBoost	1	1	0	1	17	5.6

On the different datasets, FS-SCPR ranks either first or second in various measures with the only exception being that it is ranked fifth in NDCG@1 on MQ2008. For example, for OHSUMED, FS-SCPR is the best performer for NDCG@3, NDCG@5, NDCG@10, and MAP, and is ranked second for NDCG@1. We briefly highlight some statistics. In terms of NDCG@10, FS-SCPR performs the best on NP2004, OHSUMED, and MQ2008, and is the second best performer on HP2004. On NP2004, it performs 0.38% higher compared to the second best method (ListNet) and 2.63% higher compared to the third best method (RankSVM-Primal). On OHSUMED, it outperforms the second best method (RankSVM-Primal) by 1.78% and outperforms the third best method (AdaRank-MAP) by 3.50%. On MQ2008, it performs better than the second best method (RankSVM-Primal) with a 0.39% improvement and performs better than the third best method (ListNet) by 0.65%. On HP2004, it outperforms the third best method (ListNet) with a 4.26% improvement.

Regarding MAP, FS-SCPR is ranked first on NP2004, OHSUMED, and MQ2008, and second on HP2004. On NP2004, it is superior to the second best method (RankSVM-Primal) and the third best method (ListNet) by 0.8% and 1.32%, respectively. On OHSUMED, it performs 0.09% better than the second best method (AdaRank-MAP) and performs 0.76% better than the third best method (ListNet). On MQ2008, it outperforms the second best methods (ListNet and RankBoost) by 0.02%. On HP2004, it outperforms the third best method (ListNet) by 4.59%.

Overall, the comparison results indicate that FS-SCPR performs very competitively and has stable performance on ranking on different datasets compared to other baselines. Table 16 demonstrates the following unified ranking of the methods: FS-

SCPR $\succ$ AdaRank-MAP $\succ$ RankSVM-Primal $=$ ListNet $\succ$ RankBoost, in which the proposed FS-SCPR ranks first.

5. Conclusion and Future Work

This paper addresses the feature selection problem in LTR. We proposed a graph-based filter feature selection method, FS-SCPR (see Fig. 3). FS-SCPR selects a subset of features that have minimum redundancy with each other and have maximum relevance to the ranking problem. In practice, FS-SCPR models feature relationships as a feature similarity graph. Based on such a graph model, FS-SCPR selects features using spectral clustering for redundancy minimization and biased PageRank for relevance analysis. Furthermore, we developed a new LTR for IR approach that integrates FS-SCPR and Ranking SVM (see Fig. 2). This approach exploits FS-SCPR as a preprocessor to determine discriminative and useful features and utilizes Ranking SVM to derive a ranking model with the selected features. We evaluated the proposed approach using four LETOR datasets (namely, HP2004, NP2004, OHSUMED, and MQ2008) and found that it performed well with competitive results. We presented the performance gains of the proposed approach compared to representative feature selection methods (namely, GAS, FSMSVM, and FSMRank) and state-of-the-art LTR methods (namely, AdaRank, Ranking SVM, ListNet, and RankBoost). The experimental results showed that (1) FS-SCPR can significantly boost the ranking performance; (2) FS-SCPR has superior performance to GAS (a filter method) and FSMSVM (a wrapper method), and is competitive to FSMRank (an embedded method) in a few cases; and (3) FS-SCPR performs very competitively compared to several LTR baselines and has stable performance on ranking on different datasets.

It is also worth noting that similar to other filter methods, this study tries to find a feature subset with minimum total similarity and maximum total relevance. However, the graph-based selection strategy makes this work quite distinct from the existing studies. Our approach is the first graph-based feature selection technique that uses spectral clustering for redundancy minimization and biased PageRank for relevance analysis. To the best of our knowledge, there was little graph-based attempt to tackle feature selection for LTR and this research contributes to this gap in the literature.

There remains room for improvement. First, it would be valuable to study whether improving relationships between features in the feature similarity graph will directly profit FS-SCPR. Other measures of ordinal association, such as Spearman's *rho* (ρ), are worth exploring to evaluate feature relationships. Second, methods of evaluating the goodness of a clustering can be utilized to help automatically decide the number of feature clusters. Another interesting issue to investigate is what kinds of ranking performance of features besides MAP contribute to FS-SCPR regarding biasing the PageRank computation. FS-SCPR could also be integrated with other learning methods, e.g., AdaRank-MAP. Finally, verifying the effectiveness of FS-SCPR using additional datasets would be beneficial.

References

1. Akaike, H.: Information Theory and an Extension of the Maximum Likelihood Principle. In Proceedings of the 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, 267–281. (1973)
2. Albuquerque, A., Amador, T., Ferreira, R., Veloso, A., Ziviani, N.: Learning to Rank with Deep Autoencoder Features. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN 2018), Rio de Janeiro, Brazil. (2018)
3. Aliguliyev, R. M.: Performance Evaluation of Density-based Clustering Methods. Information Sciences, Vol. 179, No. 20, 3583–3602. (2009)
4. Allvi, M. W., Hasan, M., Rayan, L., Shahabuddin, M., Khan, M. M., Ibrahim, M.: Feature Selection for Learning-to-Rank Using Simulated Annealing. International Journal of Advanced Computer Science and Applications, Vol. 11, No. 3, 699–705. (2020)
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley. (1999)
6. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems, Vol. 30, No. 1–7, 107–117. (1998)
7. Burges, C. J. C., Ragno, R., Le, Q. V.: Learning to Rank with Nonsmooth Cost Functions. In Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS 2006), Vancouver, BC, Canada, 193–200. (2006)
8. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to Rank Using Gradient Descent. In Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany, 89–96. (2005)
9. Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H.: Learning to Rank: From Pairwise Approach to Listwise Approach. In Proceedings of the 24th International Conference on Machine Learning (ICML 2007), Corvallis, OR, 129–136. (2007)
10. Chapelle, O., Keerthi, S. S.: Efficient Algorithms for Ranking with SVMs. Information Retrieval, Vol. 13, No. 3, 201–215. (2010)
11. Cheng, F., Guo, W., Zhang, X.: MOFSRank: A Multiobjective Evolutionary Algorithm for Feature Selection in Learning to Rank. Complexity, Vol. 2018, Article: 7837696. (2018)
12. Chung, F. R. K.: Spectral Graph Theory. American Mathematical Society. (1997)
13. Cossock, D., Zhang, T.: Subset Ranking Using Regression. In Proceedings of the 19th Annual Conference on Learning Theory (COLT 2006), Pittsburgh, PA, 605–619. (2006)
14. Crammer, K., Singer, Y.: Pranking with Ranking. In Proceedings of the 15th Annual Conference on Neural Information Processing Systems (NIPS 2001), Vancouver, BC, Canada, 641–647. (2001)
15. Dang, V., Croft, W. B.: Feature Selection for Document Ranking Using Best First Search and Coordinate Ascent. In Proceedings of the SIGIR 2010 Workshop on Feature Generation and Selection for Information Retrieval, Geneva, Switzerland, 28–31. (2010)
16. de Sousa, D. X., Canuto, S. D., Rosa, T. C., Martins, W. S., Gonçalves, M. A.: Incorporating Risk-Sensitiveness into Feature Selection for Learning to Rank. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016), Indianapolis, IN, 257–266. (2016)
17. Dhake, N., Raut, S., Rahangdale, A.: Identification of Efficient Algorithms for Web Search through Implementation of Learning-to-Rank Algorithms. Sādhanā, Vol. 44, No. 4, Article: 97. (2019)
18. Du, L., Pan, Y., Ding, J., Lai, H., Huang, C.: EGRank: An Exponentiated Gradient Algorithm for Sparse Learning-to-Rank. Information Sciences, Vol. 467, 342–356. (2018)
19. Du, D., Zhou, F., Xiong, W.: Cost-Sensitive ListMLE Ranking Approach Based on Sparse Representation. Journal of Information Science and Engineering, Vol. 35, No. 1, 1–22. (2019)
20. Duh, K., Kirchhoff, K.: Learning to Rank with Partially-Labeled Data. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, 251–258. (2008)

21. Freund, Y., Iyer, R., Schapire, R. E., Singer, Y.: An Efficient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research, Vol. 4, 933–969. (2003)
22. Geng, X., Liu, T.-Y., Qin, T., Li, H.: Feature Selection for Ranking. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 407–414. (2007)
23. Gigli, A., Lucchese, C., Nardini, F. M., Perego, R.: Fast Feature Selection for Learning to Rank. In Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR 2016), Newark, DE, 167–170. (2016)
24. Gupta, P., Rosso, P.: Expected Divergence Based Feature Selection for Learning to Rank. In Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), Mumbai, MH, India, 431–439. (2012)
25. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, Vol. 3, 1157–1182. (2003)
26. Haveliwala, T. H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 4, 784–796. (2003)
27. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Smola, A. J., Bartlett, P. L., Schölkopf, B., Schuurmans, D. (eds.): Advances in Large Margin Classifiers. The MIT Press, 115–132. (2000)
28. Hua, G., Zhang, M., Liu, Y., Ma, S., Ru, L.: Hierarchical Feature Selection for Ranking. In Proceedings of the 19th International Conference on World Wide Web (WWW 2010), Raleigh, NC, 1113–1114. (2010)
29. Järvelin, K., Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems, Vol. 20, No. 4, 422–446. (2002)
30. Joachims, T.: Optimizing Search Engines Using Clickthrough Data. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002), Edmonton, AB, Canada, 133–142. (2002)
31. Kendall, M. G.: A New Measure of Rank Correlation. Biometrika, Vol. 30, No. 1–2, 81–93. (1938)
32. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In Proceedings of the 7th European Conference on Machine Learning (ECML 1994), Catania, Italy, 171–182. (1994)
33. Krasotkina, O., Mottl, V.: A Bayesian Approach to Sparse Learning-to-Rank for Search Engine Optimization. In Proceedings of the 11th International Conference on Machine Learning and Data Mining (MLDM 2015), Hamburg, Germany, 382–394. (2015)
34. Lai, H., Pan, Y., Liu, C., Lin, L., Wu, J.: Sparse Learning-to-Rank via an Efficient Primal-Dual Algorithm. IEEE Transactions on Computers, Vol. 62, No. 6, 1221–1233. (2013)
35. Lai, H., Pan, Y., Tang, Y., Liu, N.: Efficient Gradient Descent Algorithm for Sparse Models with Application in Learning-to-Rank. Knowledge-Based Systems, Vol. 49, 190–198. (2013)
36. Lai, H.-J., Pan, Y., Tang, Y., Yu, R.: FSMRank: Feature Selection Algorithm for Learning to Rank. IEEE Transactions on Neural Networks and Learning Systems, Vol. 24, No. 6, 940–952. (2013)
37. Laporte, L., Flamary, R., Canu, S., Déjean, S., Mothe, J.: Nonconvex Regularizations for Feature Selection in Ranking with Sparse SVM. IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, No. 6, 1118–1130. (2014)
38. Li, P., Burges, C. J. C., Wu, Q.: McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS 2007), Vancouver, BC, Canada, 897–904. (2007)
39. Lin, Y., Lin, H., Xu, K., Sun, X.: Learning to Rank Using Smoothing Methods for Language Modeling. Journal of the American Society for Information Science and Technology, Vol. 64, No. 4, 818–828. (2013)
40. Liu, T.-Y.: Learning to Rank for Information Retrieval. Springer. (2011)

41. Lu, M., Xie, M., Wang, Y., Liu, J., Huang, Y.: Cost-Sensitive Listwise Ranking Approach. In Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010), Hyderabad, India, 358–366. (2010)
42. Naini, K. D., Altingovde, I. S.: Exploiting Result Diversification Methods for Feature Selection in Learning to Rank. In Proceedings of the 36th European Conference on Information Retrieval (ECIR 2014), Amsterdam, The Netherlands, 455–461. (2014)
43. Nallapati, R.: Discriminative Models for Information Retrieval. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), Sheffield, South Yorkshire, UK, 64–71. (2004)
44. Ng, A. Y., Jordan, M. I., Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. In Proceedings of the 15th Annual Conference on Neural Information Processing Systems (NIPS 2001), Vancouver, BC, Canada, 849–856. (2001)
45. Pahikkala, T., Airola, A., Naula, P., Salakoski, T.: Greedy RankRLS: A Linear Time Algorithm for Learning Sparse Ranking Models. In Proceedings of the SIGIR 2010 Workshop on Feature Generation and Selection for Information Retrieval, Geneva, Switzerland, 11–18. (2010)
46. Pahikkala, T., Tsivtsivadze, E., Airola, A., Järvinen, J., Boberg, J.: An Efficient Algorithm for Learning to Rank from Preference Graphs. Machine Learning, Vol. 75, No. 1, 129–165. (2009)
47. Pan, F., Converse, T., Ahn, D., Salvetti, F., Donato, G.: Feature Selection for Ranking Using Boosted Trees. In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, China, 2025–2028. (2009)
48. Pandey, G., Ren, Z., Wang, S., Veijalainen, J., de Rijke, M.: Linear Feature Extraction for Ranking. Information Retrieval Journal, Vol. 21, No. 6, 481–506. (2018)
49. Purpura, A., Buchner, K., Silvello, G., Susto, G. A.: Neural Feature Selection for Learning to Rank. In Proceedings of the 43rd European Conference on Information Retrieval (ECIR 2021), 342–349. (2021)
50. Qin, T., Liu, T.-Y.: Introducing LETOR 4.0 Datasets. arXiv preprint (arXiv:1306.2597) (2013). [Online]. Available: https://arxiv.org/abs/1306.2597 (current May 2021)
51. Qin, T., Liu, T.-Y., Xu, J., Li, H.: LETOR: A Benchmark Collection for Research on Learning to Rank for Information Retrieval. Information Retrieval, Vol. 13, No. 4, 346–374. (2010)
52. Qin, T., Zhang, X.-D., Tsai, M.-F., Wang, D.-S., Liu, T.-Y., Li, H.: Query-Level Loss Functions for Information Retrieval. Information Processing & Management, Vol. 44, No. 2, 838–855. (2008)
53. Rahangdale, A., Raut, S.: Deep Neural Network Regularization for Feature Selection in Learning-to-Rank. IEEE Access, Vol. 7, 53988–54006. (2019)
54. Robertson, S. E.: Overview of the Okapi Projects. Journal of Documentation, Vol. 53, No. 1, 3–7. (1997)
55. Shashua, A., Levin, A.: Ranking with Large Margin Principle: Two Approaches. In Proceedings of the 16th Annual Conference on Neural Information Processing Systems (NIPS 2002), Vancouver, BC, Canada, 961–968. (2002)
56. Shirzad, M. B., Keyvanpour, M. R.: A Feature Selection Method Based on Minimum Redundancy Maximum Relevance for Learning to Rank. In Proceedings of the 5th Conference on Artificial Intelligence and Robotics (2015 AI & Robotics), Qazvin, Iran. (2015)
57. Spearman, C.: The Proof and Measurement of Association Between Two Things. The American Journal of Psychology, Vol. 15, No. 1, 72–101. (1904)
58. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In Proceedings of the KDD 2000 Workshop on Text Mining, Boston, MA, 109–110. (2000)
59. Sun, Z., Qin, T., Tao, Q., Wang, J.: Robust Sparse Rank Learning for Non-Smooth Ranking Measures. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009), Boston, MA, 259–266. (2009)

60. Tan, P.-N., Steinbach, M., Karpatne, A., Kumar, V.: Introduction to Data Mining (2nd edition). Pearson. (2019)
61. Taylor, M., Guiver, J., Robertson, S., Minka, T.: SoftRank: Optimizing Non-Smooth Rank Metrics. In Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM 2008), Palo Alto, CA, 77–86. (2008)
62. Tsai, M.-F., Liu, T.-Y., Qin, T., Chen, H.-H., Ma, W.-Y.: FRank: A Ranking Method with Fidelity Loss. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 383–390. (2007)
63. Volkovs, M. N., Zemel, R. S.: BoltzRank: Learning to Maximize Expected Ranking Gain. In Proceedings of the 26th International Conference on Machine Learning (ICML 2009), Montreal, QC, Canada, 1089–1096. (2009)
64. von Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing, Vol. 17, No. 4, 395–416. (2007)
65. Wierzchoń, S. T., Kłopotek, M. A.: Modern Algorithms of Cluster Analysis. Springer. (2018)
66. Xia, F., Liu, T.-Y., Wang, J., Zhang, W., Li, H.: Listwise Approach to Learning to Rank - Theory and Algorithm. In Proceedings of the 25th International Conference on Machine Learning (ICML 2008), Helsinki, Finland, 1192–1199. (2008)
67. Xu, J., Li, H.: AdaRank: A Boosting Algorithm for Information Retrieval. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 391–398. (2007)
68. Xu, J., Liu, T.-Y., Lu, M., Li, H., Ma, W.-Y.: Directly Optimizing Evaluation Measures in Learning to Rank. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008), Singapore, 107–114. (2008)
69. Yeh, J.-Y., Lin, J.-Y., Ke, H.-R., Yang, W.-P.: Learning to Rank for Information Retrieval Using Genetic Programming. In Proceedings of the SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007), Amsterdam, The Netherlands, 41–48. (2007)
70. Yeh, J.-Y., Tsai, C.-J.: Graph-based Feature Selection Method for Learning to Rank. In Proceedings of the 6th International Conference on Communication and Information Processing (ICCIP 2020), Tokyo, Japan, 70–73. (2020)
71. Yu, H., Oh, J., Han, W.-S.: Efficient Feature Weighting Methods for Ranking. In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, China, 1157–1166. (2009)
72. Yue, Y., Finley, T., Radlinski, F., Joachims, T.: A Support Vector Method for Optimizing Average Precision. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 271–278. (2007)
73. Zhai, C., Lafferty, J.: A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001), New Orleans, LA, 2001, 334–342. (2001)

Jen-Yuan Yeh is currently an Associate Researcher of the Dept. of Operation, Visitor Service, Collection and Information Management at the National Museum of Natural Science. His research interests include text mining and summarization, information retrieval and extraction, digital libraries and museums, and natural language processing.

Cheng-Jung Tsai is currently a professor in the Graduate Institute of Statistics and Information Science at National Changhua University of Education, Chang-Hua,

Taiwan, R.O.C. His research interests include data mining, big data analysis, information security, e-learning, and digital image processing.

Received: December 20, 2020; Accepted: July 25, 2021.

Computer Science and Information Systems 19(1):165–184 https://doi.org/10.2298/CSIS200424038K

Deep RNN-Based Network Traffic Classification Scheme in Edge Computing System

Kwihoon Kim[1], Joohyung Lee[2], Hyun-Kyo Lim[3], Se Won Oh[4], and Youn-Hee Han[5⋆]

[1] Department of Artificial Intelligence Convergence Education,
Korea National University of Education, Cheongju 28173, South Korea
kimkh@knue.ac.kr
[2] Department of Software, Gachon University,
Seongnam 13120, South Korea
j17.lee@gachon.ac.kr
[3] Interdisciplinary Program in Creative Engineering,
Korea University of Technology and Education, Cheonan 31253, South Korea
glenn89@koreatech.ac.kr
[4] Electronics and Telecommunications Research Institute,
Daejeon 34129, South Korea
sewonoh@etri.re.kr
[5] Future Convergence Engineering, Korea University of Technology and Education,
Cheonan 31253, South Korea
yhhan@koreatech.ac.kr

Abstract. This paper proposes a deep recurrent neural network (RNN)-based traffic classification scheme (deep RNN-TCS) for classifying applications from traffic patterns in a hybrid edge computing and cloud computing architecture. We can also classify traffic from a cloud server, but there will be a time delay when packets transfer to the server. Therefore, the traffic classification is possible almost in real-time when it performed on edge computing nodes. However, training takes a lot of time and needs a lot of computing resources to learn traffic patterns. Therefore, it is efficient to perform training on cloud server and to perform serving on edge computing node. Here, a cloud server collects and stores output labels corresponding to the application packets. Then, it trains those data and generates inferred functions. An edge computation node receives the inferred functions and executes classification. Compared to deep packet inspection (DPI), which requires the periodic verification of existing signatures and updated application information (e.g., versions adding new features), the proposed scheme can classify the applications in an automated manner. Also, deep learning can automatically make classifiers for traffic classification when there is enough data. Specifically, input features and output labels are defined for classification as traffic packets and target applications, respectively, which are created as two-dimensional images. As our training data, traffic packets measured at Universitat Politecnica de Catalunya Barcelonatech were utilized. Accordingly, the proposed deep RNN-TCS is implemented using a deep long short-term memory system. Through extensive simulation-based experiments, it is verified that the proposed deep RNN-TCS achieves almost 5% improvement in accuracy (96% accuracy) while operating 500 times faster (elapsed time) compared to the conventional scheme.

Keywords: RNN, Traffic Classification, Edge Computing, Cloud Computing.

⋆ Corresponding author

1. Introduction

To realize advanced network management, user service, and security functions according to various application traffic, service providers are required to design effective ways to inspect and identify their application traffic. Aiming for this goal, the deep packet inspection (DPI), which is a type of network packet filtering technique, has been a widely deployed approach for many years; it examines packet payloads to identify application traffic. Specifically, the DPI technique utilizes unique byte patterns (e.g., headers, data protocol structures and the payload of the message) as signatures to detect the application type. Because most recent applications are frequently updated to add new features with different versions, an accurate DPI system must periodically verify and update existing signatures, which sometimes requires much human intervention. Further, the manual task of application traffic generation and verification on multiple platforms and updated applications is highly tedious and error-prone [39,11,3,27].

This limitation has recently motivated research to establish lightweight and automated methods for classifying application traffic [35,2]. Recently, owing to the breakthroughs made by deep learning technique in various typical algorithms including deep multilayer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), and long short-term memory (LSTM) [28,36,20,43], there have been broad use cases in the area of image classification (e.g., almost 98% accuracy achieved in image classification). According to Lecun et al. [19], the deep learning technology performed better in image classification than classical machine learning algorithms such as support vector machines (SVM) and Random Forest (RF). Despite its practical popularity in deep learning techniques, there has been only a limited number of research works on the automated classification of application traffic. In particular, the authors of [43] first applied and proposed a deep CNN-based traffic classification system in order to detect malware applications. However, because of the CNN's own characteristics, which are generally used to handle batch data and not for streaming data (i.e., time-series analysis), there is still room to improve its accuracy by considering time-varying packet payload patterns depending on the type of application, which inspired our work [17,14,5,15,32,16,33].

In this paper, a deep RNN-based traffic classification scheme (deep RNN-TCS) is proposed by adopting the RNN technique, which is suitable for training on streaming data. CNN is good at classifying the image data, especially such as data with shift invariant characteristics. On the other hand, RNN is good at classifying the time series data. Because network traffic flows through time series, it is appropriate to classify using RNN. The proposed deep RNN-TCS provides lightweight and automated classification of application traffic. Specifically, a novel learning platform is designed suited for detecting time-varying application traffic where input features and output labels are mapped to traffic packets and target applications, respectively. A hybrid edge computing and cloud computing architecture is considered. Here, a cloud server collects and stores output labels corresponding to the application packets. Then, it trains those data and generates inferred functions. An edge computation node receives the inferred functions and executes classification. From input features, multiple two-dimensional square matrices for sequential flows in preprocessing are created and trained in a stacked RNN model. As our training data, traffic packets measured at Universitat Politecnica de Catalunya Barcelonatech were utilized. Accordingly, the proposed deep RNN-TCS is implemented using a deep long short-term memory (LSTM) system. Through extensive simulation-based experiments, it

is revealed that the proposed scheme improves accuracy by almost 5% (96% accuracy) while operating 500 times faster (elapsed time) compared to the conventional scheme over five types of applications from the produced inferred function. In this study, network traffic classification is performed using only the payload excluding header information among network packet information. Recently, many IoT and mobile devices use private or dynamic IP addresses and changeable port numbers. So the classification of network traffic based on packet header information is no longer accurate [45]. The payload-based network traffic classification can solve the problem. The contributions of this paper can be summarized as follows.

- In our scheme, by applying the RNN mechanism to the traffic classification problem, we design new input features of the traffic payload data as the image data with a two-dimensional fixed-size matrix, so that only payload of packets, excluding TCP/IP headers, are used for training and inference data.
- We deploy the proposed deep RNN-TCS by considering a hybrid edge computing and cloud computing architecture is considered. Here, the cloud computing acts as a learner, which collects and stores output labels corresponding to the application packets received from PC clients and creates inferred functions for classification through a deep-learning process. Then, those inferred functions (i.e., deep learning model) are delivered to the edge computing. Correspondingly, the edge computing performs classification of application packets without output labels by using the deep-learning model delivered from the cloud computing.
- Through extensive simulation-based experiments, it is verified that the proposed deep RNN-TCS achieves almost 5% improvement in accuracy (96% accuracy) while operating 500 times faster (elapsed time) compared to the conventional scheme.
- Our research is not limited to this vanilla RNN. There are RNNs that have been modified recently, and it is easy to apply to reflect modified RNNs. Later, it will be the future work to apply the revised RNN to improve performance.

We can thus develop a practical deep recurrent neural network-based traffic classification scheme. Section II explains the related work of network traffic classification problem. Section III presents an overview of the proposed system model. In Section IV, we formulate the problem as a deep learning model and present the novel deep RNN-TCS. Numerical results and performance analysis are explained in Section V. We summarize and conclude this work in Section VI.

2. Related work

Recently, to solve each problem in various domains, such as smart homes, airport gate assignments, and the traveling salesmen, rule-based algorithms such as daily activation recognition, and classical machine learning algorithms such as the Support Vector Machine (SVM) and the Random Forest (RF) are being studied [24,7,6]. Classical algorithms such as the Principal Component Analysis (PCA), the Broad Learning System (BLS) techniques, a new performance degradation prediction method, and a genetic and ant colony adaptive collaborative optimization are being studied to address abnormal detection issues in manufacturing areas such as the Fault Diagnosis and the Prognostic and Health Management (PHM) [49,48,8]. For the network domain case, many researches have focused

on network traffic classification methods. Existing researches include rule-based network traffic classification method.

Recently, researches on network traffic classification method using good performance deep learning models have been actively performed [13,45]. Network traffic classification using deep learning is a method of automatically classifying packets without human intervention. Existing rule-based network traffic classification is a method of classifying packets having the network according to predefined rules [21,29,40,30]. For example, the classification methods use the header of the network packets. Therefore, rule-based network traffic classification is conducted on the basis of IP address and port number of the packet header. Li et al. proposed an approach to reduce the dependency on packet header information [21,22,38]. Using this approach, they found that their packet-shaping device uses the HTTP and TLS-handshake fields in their matching rules but only for the first packet in each direction. If there is similar information in the header information of the incoming packet compared to the header information found in the first packet, the incoming packet is classified as a packet of the same type. Although there is less dependence on the IP and port number of the packet, the method of classifying subsequent packets using the header information of the first packet still depends on the header information. However, since the rule-based network traffic classification method is highly dependent on the header information (Source IP / Port number, Destination IP / Port number), network traffic classification methods such as Correlation-based and payload-based methods have been studied. Correlation-based network classification classifies datasets by selecting packets with high correlation between traffic packets considering correlation between network traffic [47,18,9]. Zhang et al have shown that there is a strong correlation between flow size and rate [46]. The flow of application used by the user has a certain size and rate [47]. Also, user behavior might have an effect on large flows. Erman et al. consider the problem of traffic classification in the core network [9].

The packet classification at the core network is challenging because only partial header information about the flow are available. So, they use only unidirectional flow records. Specifically, they propose and evaluate a clustering-based framework for classifying network traffic using only unidirectional flow statistics. And their work is facilitated by recent full-payload Internet packet traces [22]. As the research on payload-based network traffic classification is studied, network traffic classification methods using machine learning and deep learning are being studied variously [43,9,31,44,12,37,25,26,10]. Haffner et al. used a variety of traditional machine learning techniques to compare and analyze packet payloads [12]. It reduced the amount of computation required when generating payload-based datasets. Toward this end, they used only the first few bytes of unidirectional traffic data and unencrypted TCP data. Specifically, they used NB, AdaBoost, and MaxEnt for traffic classification. AdaBoost outperforms NB and MaxEnt, yielding an overall precision of 99% with an error rate within 0.5%. Shafiq et al. used to classify network traffic by various machine learning algorithms using different kinds of datasets [37]. They used the three machine learning algorithms, multi-layer perceptron (MLP), C4.5 decision tree, and support vector machine (SVM).

However, recent developments in computing resources have led to a significant advance in deep learning fields that can be applied to network traffic classification. Especially, as the CNN and RNN models in the deep learning model are developed, they can be easily applied to the classification of network traffic. Wang et al. classified malware traf-

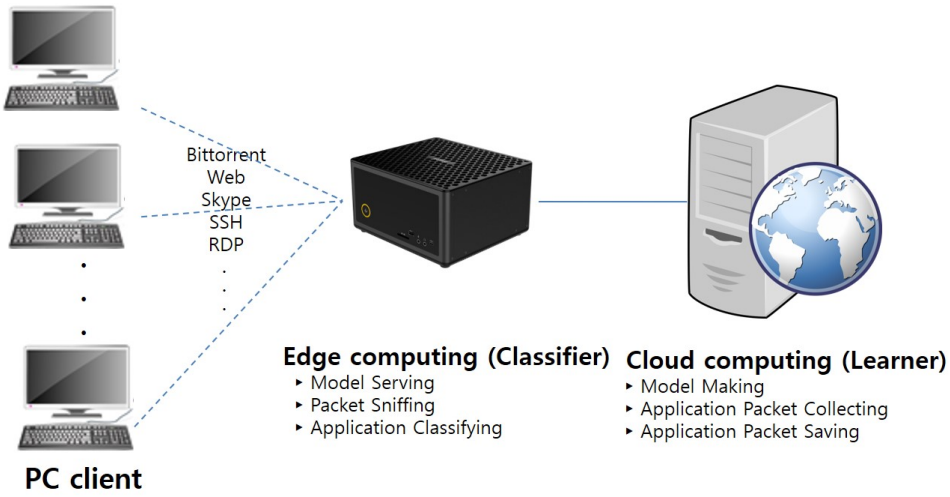


Fig. 1. Proposed system model for applying the proposed scheme

fic and normal traffic by using CNN [43]. To generate training set, header information of packet was extracted using DPI tool. Based on the 5-tuple (source IP / port number, destination IP / port number, protocol) of the extracted header information, flow-based dataset and session-based dataset is generated. The generated flow and session-based datasets again generate 28x28 training sets for each packet through an imaging process suitable for the CNN model. The trained CNN model using flow-based and session-based datasets is 100% accurate for malware traffic and normal traffic classification. Lopez-Martin et al. used to classify the network traffic using a combined CNN and LSTM [25]. The dataset is extracted from the packet headers of the network traffic and learns the dataset using a model that combines the single-layer CNN with LSTM, CNN, and LSTM. However, most network traffic classification methods that use in deep learning use the IP, port number, and MAC address of the packet header information as a feature of the training set. In this paper, a preprocessing process that extracts only the payload of packets from network traffic is implemented, and detailed comparison and analysis of the layers of CNN and LSTM are provided. Aceto et al. and Wang et al. utilizes various models of deep learning (CNN, LSTM, SAE, MLP) to classify application of network traffic using payload data as well as header information. Since the IP and port numbers, which are some information in the TCP/IP headers, are changed dynamically, they have the bad effects when they are used for classifying packets [41,1].

3. System Model

Multiple personal computer (PC) clients are considered that generate traffic while executing an application where they are connected to a cloud computing via an intermediate node called an “edge computing”. Accordingly, a hybrid edge computing and cloud computing architecture is considered. Here, the cloud computing acts as a learner, which collects and stores output labels corresponding to the application packets received from

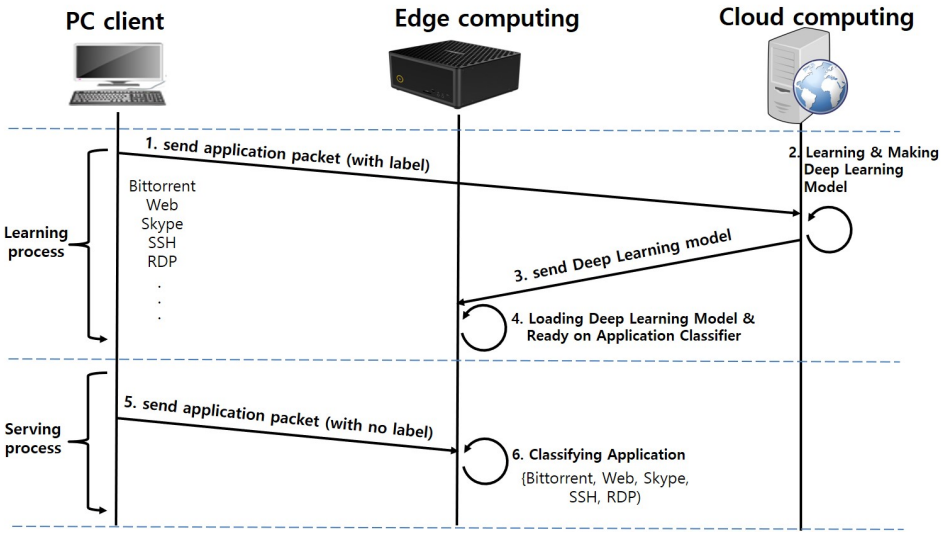
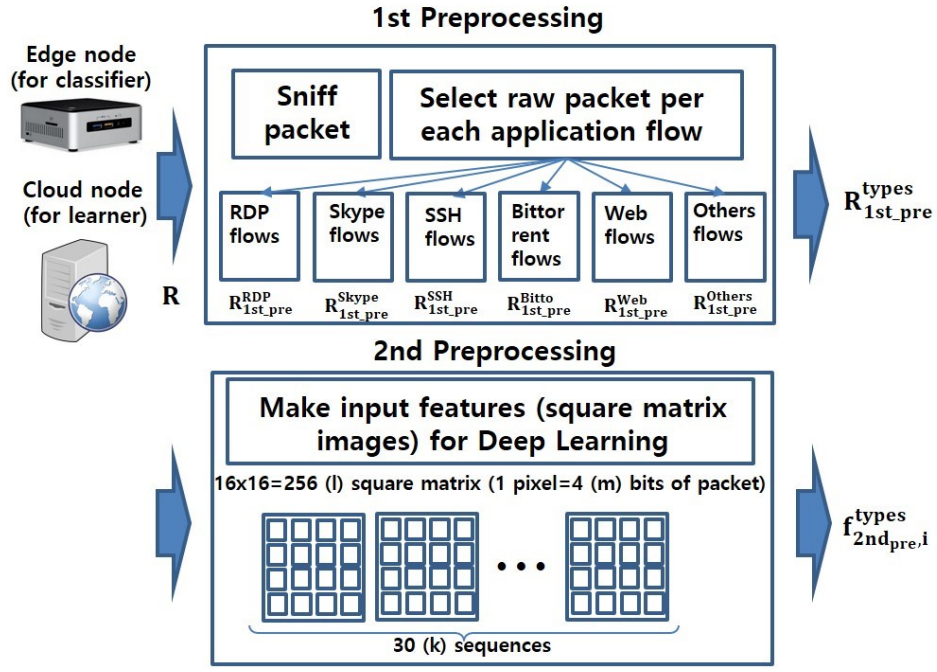


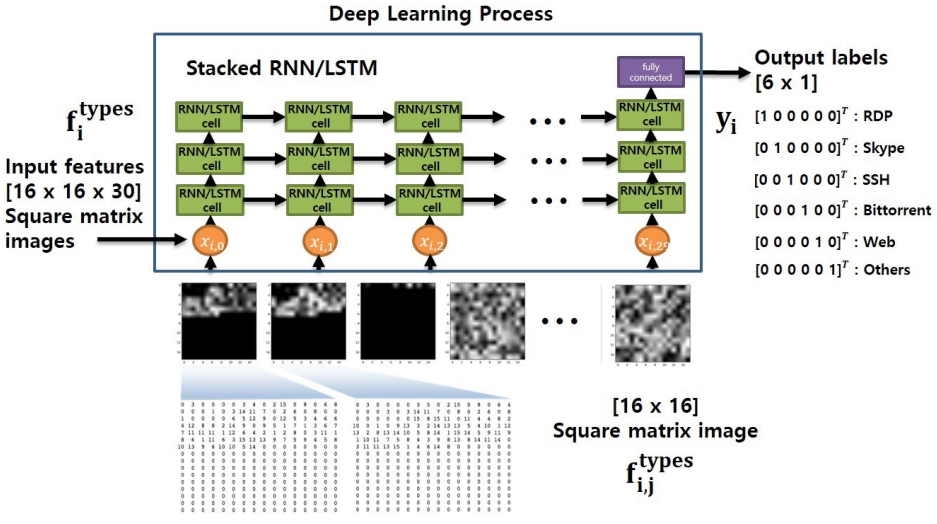
Fig. 2. Proposed procedures for applying the proposed scheme

PC clients and creates inferred functions for classification through a deep-learning process. Then, those inferred functions (i.e., deep learning model) are delivered to the edge computing. Correspondingly, the edge computing performs classification of application packets without output labels by using the deep-learning model delivered from the cloud computing. Fig. 1 shows a detailed system model and procedures of the proposed deep RNN-TCS.

As Fig. 2 is shown, detailed procedures consist of two processes: a learning process and a serving process. In the learning process, PC clients send the application packets with a corresponding output label to the cloud computing via the edge computing. In this case, five types of applications (e.g., BitTorrent, web service, Skype, secure shell (SSH), and remote desktop protocol (RDP)) are considered as classification candidates. Here, it should be noted that the proposed scheme is not limited to classifying those five applications and can be easily extended to classify different types. In the cloud computing, first the collected packets are preprocessed, and through the learning process based on the output labels corresponding to the collected packets, an appropriate deep-learning model is created. Afterwards, the result of the deep-learning model is sent to the edge computing. The edge computing loads the received deep-learning model to act as an application classifier. In the serving process, the PC clients send the application packets without the corresponding output labels. Then, the edge computing examines or sniffs the application packets and classifies the application, which would be mapped into one of the five candidates. On the basis of this automated classification function, it is expected that a service provider can effectively perform the desired network management according to various application traffic.



(a)



(b)

Fig. 3. Proposed deep RNN-based traffic classification scheme for the (a) data preprocessing and (b) deep learning process (Stacked RNN/LSTM)

Table 1. Parameters explanations

Parameters	Explanations
$\boldsymbol{R}$	The raw data set for the first preprocessing procedure
$\boldsymbol{R}_{1st\_pre}^{type}$	The first preprocessing result with a certain type, which is a subset of $\boldsymbol{R}$
$\boldsymbol{N}^{flow}$	The data size of one sequential flow
$\boldsymbol{f}_{2nd\_pre,i}^{types}$	The second preprocessing data of the flow i
$\boldsymbol{f}_{i,0}^{types}$	The sequence 0 of flow i with a certain type

4. Proposed Deep RNN-Based Traffic Classification Scheme

In this section, the detailed process of the proposed scheme is explained as shown in Fig. 3. The entire process is divided into two parts: data preprocessing and deep learning. Detailed parameters are summarized in Table 1

4.1. Proposed Data Preprocessing

The set of features entering LSTM's input is payload data for packets in each flow. The input data of the neural network changed each element of the payload to 8 bits, and then re-imaged the bitted payload data and used it as input data through the preprocessing. In the proposed scheme, preprocessing is conducted in two stages, called the first preprocessing and second preprocessing procedures. For the first preprocessing procedure, raw data with a defined set $\boldsymbol{R}$ is collected and sniffed in the cloud computing. Then, raw data is classified into a corresponding type of application. Here, because only six types of applications are considered, including "others," each flow belongs to one of six types. For convenience, $\boldsymbol{R}_{1st\_pre}^{type}$ is denoted as a subset of $\boldsymbol{R}$, which is the first preprocessing result with a certain type, i.e., $\boldsymbol{type} \in \{RDP, Skype, SSH, BitTorrent, Web, Others\}$. In this process, a conversion process of the raw data is conducted, which removes the head information that indicates the flow id, start time, and end time of each flow, and imports the data portion of the application payload in the flow unit and creates a data set. The first preprocessing result is given by

$$\boldsymbol{R} = \boldsymbol{R}_{1st\_pre}^{RDP} \cup \boldsymbol{R}_{1st\_pre}^{Skype} \cup \boldsymbol{R}_{1st\_pre}^{SSH} \cup \boldsymbol{R}_{1st\_pre}^{Bitto} \cup \boldsymbol{R}_{1st\_pre}^{Web} \cup \boldsymbol{R}_{1st\_pre}^{Others} \quad (1)$$

For the second preprocessing procedure, square matrix images are generated from the classified raw data in the first preprocessing procedure. That is, each flow corresponding to a certain type contains sequential data with time-series (e.g., streaming data). In addition, one square matrix image in each sequential datum has a user-designed size, which can be set by the user to l. The bit size of one pixel in the square matrix image has a m, and the character-type value replaced by a floating-point value is $0 \sim (2^m - 1)$. That is, the data size of one sequential flow (N^{flow}) is obtained by

$$\mathrm{N^{flow}} = \quad \mathrm{k} \quad \times \quad \mathrm{l} \quad \times \quad \mathrm{m} \quad (2)$$

where k is the number of square matrix images per one flow, l is the number of pixels per one square matrix image, and m is the size of a pixel.

For instance, in the case of 30 sequences per flow, 256($= 16 \times 16$) pixels per sequence (square matrix image), and 4 bits per pixel for RNN models detecting time-varying target applications, N^{flow} has a value 30,720 ($= 30 \times 256 \times 4$).

Fig. 3(a) shows the payload of a packet in one of the completed flows in a two-dimensional image, an element of four bits in size, and a value between 0 and 15 (floating point). In addition, because of the nature of the LSTM network structure, all flows should have the same number of packets. Because the length of a sequence is set in advance, the number of packets per flow should be the same as the predefined length. However, because application flows can have a varying number of packets, the number of packets should be processed with a defined size. Specifically, if the number of packets per flow is smaller than N^{flow}, the packet is padded by zero. Contrarily, if the number of packets per flow is greater than N^{flow}, then the packets over N^{flow} are discarded. Finally, the second preprocessing data of the flow i ($\mathrm{f}^{\mathrm{types}}_{\mathrm{2nd\_pre,i}}$) is given by

$$\mathrm{f}^{\mathrm{types}}_{\mathrm{2nd\_pre,i}} = [f^{types}_{i,0} \quad f^{types}_{i,1} \quad \cdots \quad f^{types}_{i,29}], \tag{3}$$

where $f^{types}_{i,0}$ is denoted as a sequence 0 of flow i with a certain type.

4.2. Proposed Deep-Learning Process

In the deep-learning process, these generated input features with corresponding output labels are inserted and trained to create an acceptable inferred function (i.e., deep learning model) where a stacked RNN model with three layers was utilized in this study to improve accuracy with consideration of time-varying target applications. In particular, for the existing vanilla RNN model, the length of the sequences should be short; otherwise, acceptable accuracy cannot be achieved, which is called the "long-term dependency problem." To alleviate this issue while using 30 sequences per flow, RNN with the LSTM method is additionally considered, which is composed of a memory cell, an input gate, an output gate, and a forget gate. Then, each LSTM cell takes an input and stores it for some period of time to solve the "long-term dependency problem." The output labels used for the train, validation, and test tasks were designed as one-hot vectors as shown in Table 2, where the size of the one-hot vectors is $[5 \times 1]$ because of the five application types. LSTM [48, 49] is a type of the RNN model. It is useful for training datasets with long-term dependency, so that it is commonly used to train speech and text dataset. In LSTM, the previous learning data is reflected in the current learning data using the cyclic structure. As a result, LSTM is suitable for the classification of the flow-based network traffic dataset with sequential feature. As explained in Section II, collection of the application packets and learning for the deep learning model is conducted in the cloud computing, and the classification of an application packet is served by the edge computing. For learning, an equal amount of data for the six application types should be prepared, which is used as an input for generating the deep learning model. As depicted in Fig. 3(b), $f^{types}_{i,j}$, with $= 16 \times 16$ square matrix images, is used as an input feature, where j is a sequence index from 0 to 29. In addition, each input feature has a corresponding output label designed as in Table 2. Finally, in the serving process, classification is conducted on the basis of the result of output labels $[5 \times 1]$ such that the largest vector can be selected as an inferred application type. For instance, if the result represents a vector with [0.01 0.12 0.76 0.04 0.07], SSH with [0 0 1 0 0] is selected.

Table 2. Output labels of applications (one-hot vector)

Application	BitTorrent	Web	Skype	SSH	RDP
Output label	$\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}$

Table 3. Deep-learning system environment

	Case	**Description**
Cloud Computing	DL Toolkit	Tensorflow 1.12
	Language	Python 3.6
	OS	Ubuntu 16.04 LTS
	RAM	32 GB
	GPU	Two NVIDIA GTX 1080Ti, 11GB
	CPU	Intel Core i9-7900X @ 3.30GHz
	Hyper parameter	Optimizer: Adam, Batch: 100, Epoch: 200
Edge Computing	DL Toolkit	Tensorflow 1.12
	Language	Python 3.6
	OS	Ubuntu 16.04 LTS
	RAM	32 GB
	GPU	One NVIDIA GTX 1080Ti, 11GB
	CPU	AMD Ryzen 5 1400 Quad-Core Processor
	Hyper parameter	Optimizer: Adam, Batch: 100, Epoch: 200

5. Performance Evaluation

In this section, the details regarding the performance evaluation of the proposed scheme is discussed. The proposed deep RNN-TCS is compared with the conventional approach, a deep CNN-traffic classification scheme (deep CNN-TCS) [43].

The experiments were conducted on Ubuntu 16.04 LTS, using 32 GB of RAM and two NVIDIA GTX 1080Ti GPUs with 11 GB for the cloud computing and using 16 GB of RAM and one NVIDIA GTX 1080Ti GPU with 11 GB for the edge computing. Tensorflow 1.12 in the Python 3.6 environment is used to configure the LSTM and CNN deep-learning models. For the experiments, 2000 flows for each application are used in the training data, and the number of packets per flow was set to 10, 30, 60, and 100. In addition, the payload size of each packet was set to 40, 80, and 160. Table 3 shows the detailed hyper-parameters used in deep RNN-TCS and deep CNN-TCS. There is a limitation of the DL-PAS in applications to a high-user-density nature owing to the factorial increase in both output nodes and multi-layer perceptions (MLP) weight. The LSTM model used in RNN-TCS consists of a single LSTM layer, and an output layer. The number of cells in the LSTM layer is 320, and we use the 'uniform' distribution for the model parameter initializer. The dropout rate is set to 0.2 in order to prevent the overfitting. The activation function is 'softmax', optimization type is 'adam', and batch size is 100.

The traffic data used to classify applications were preprocessing packet capture (PCAP) files supplied by Universitat Politecnica de Catalunya Barcelonatech (UPC) [4] suitable

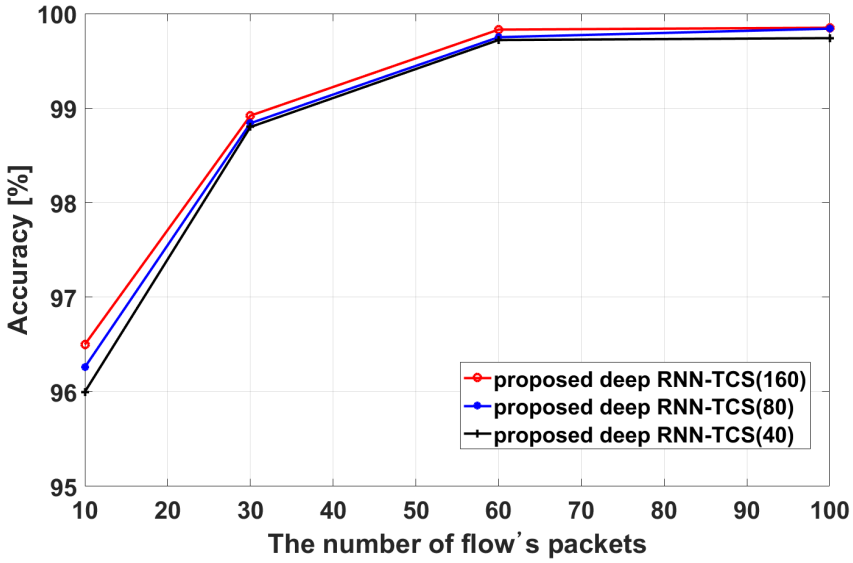


Fig. 4. Accuracy of the deep RNN-based traffic classification scheme

for RNN learning. For the data preprocessing process, six types of applications are considered, where web applications included HTTP-Facebook-Google, HTTP-Web, HTTP-Wiki, and HTTP-Youtube. The learning data set included flight data with 8,750 flows, 1,250 validation data, and 3,000 test data. For five applications, the overall accuracy of the deep RNN-TCS with respect to the number of packets per flow are examined. In this investigation, the deep RNN-TCS is tested with different payload sizes, i.e., 40, 80, or 160 bytes, denoted by proposed deep RNN-TCS(40), deep RNN-TCS(80), and deep RNN-TCS(160), respectively. As shown in Fig. 4, as the number of packets per flow increases, the accuracy increases. Similarly, the accuracy also increases as the payload size increases, because a lot of packets and big payload size provides more information to the classifier to make an accurate decision. Specifically, the proposed scheme achieves accuracy of 96.00% - 99.85% with varying conditions.

Fig. 5 shows the accuracy of the proposed scheme for each five applications. As shown in the figure, the accuracy of each application increases as the number of packets per flow increases. In particular, SSH, RDP and Skype usually consist of text or control data with fewer bits, thus requiring fewer packets per flow. Correspondingly, as the number of packets per flow and payload size increase in the all applications, the accuracy of RNN-TCS also reaches about 99%. However, if the dataset size is small (i.e., 10 packets per flow with 40 payload size), especially in case of Web and BitTorrent, the accuracy is not as high (88.97% and 95.65%, respectively). Because BitTorrent and Web are usually composed of image or video data with a large amount of data, our scheme requires more packets per flow to classify these applications accurately.

A clearly presenting the predictions of the deep RNN-TCS model is to use a confusion matrix. Table 4 shows the confusion matrix generated by the test process of the proposed deep RNN-TCS as a flow-based dataset. For example, the deep RNN-TCS ac-

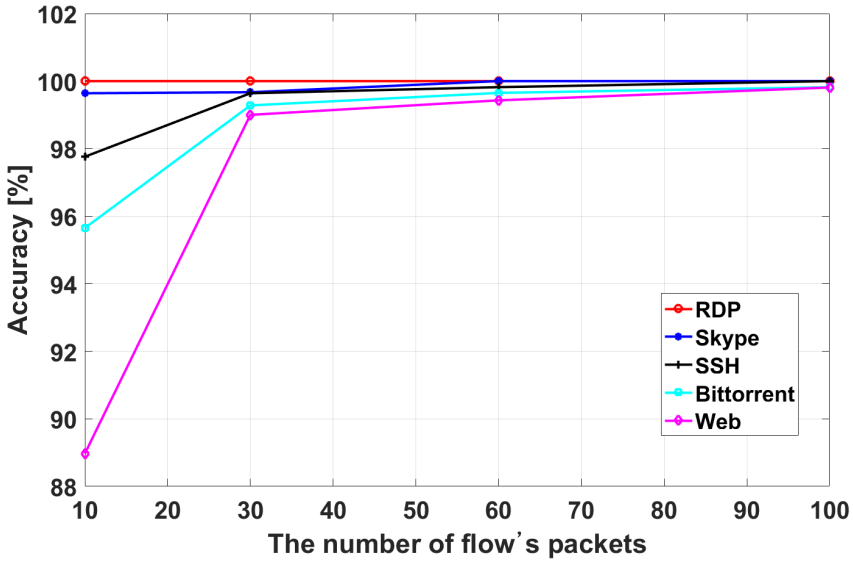


Fig. 5. Accuracy of deep RNN-TCS for each target application

curately predicts 574 BitTorrents, 593 Web, 578 Skype, 594 SSH and 603 RDP. It also incorrectly predicted 58 cases (all cases except the diagonal position in Table 4) from the total number of all estimates. In addition, the F1 score of each application label remains high with strong robustness, demonstrating that the deep RNN-TCS model can effectively and reliably predict network applications. If the number of application labels is different, the accuracy indicator may be misrepresented. The model predicts most applications for all predictions and achieves high classification accuracy, but the model may not be useful for problem areas. Therefore, we replace the confusion matrix, which is the prediction result for all application labels, with the binary confusion matrix for each label. Table 5 is an example of a binary confusion matrix of BitTorrent labels created based on Table 4. In the binary confusion matrix in Table 5, the deep RNN-TCS predicts 600 (= 574 + 26) of the total 3000 test datasets as BitTorrent and 2400 (= 19 + 2381) as the remainder. In fact, 593 (= 574 + 19) of the test dataset is bit torrent and 2407 (= 26 + 2381) is the remainder. The TP represents the cases in which the actual label is positive (BitTorrent) and the prediction result is also positive correctly. The FN represents the cases in which the actual label is positive, but the prediction result is negative incorrectly. The FP represents the cases in which the actual label is negative (that is, not BitTorrent), but prediction result is positive incorrectly. Lastly, the TN represents the cases in which the actual label is negative, and the prediction result is also negative correctly.

To solve the reliability issue of accuracy, the deep RNN-TCS is evaluated by calculating the F1-score in this paper. Using a binary confusion matrix, it is defined according to the following equation:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Table 4. The all applications confusion matrix by the proposed deep RNN-TCS test with 100 packets per flow and 160 pixels.

Applications		Predicted						
		BitTorrent	Web	Skype	SSH	RDP	SUM	F1 score
Actual	BitTorrent	574	0	2	17	0	593	0.99
	Web	6	593	1	1	0	601	1.00
	Skype	19	1	578	1	0	599	1.00
	SSH	1	3	0	594	5	603	0.99
	RDP	0	0	0	1	603	604	1.00
Total F1-score								0.99

Table 5. The Binary confusion matrix of BitTorrent application labels in deep RNN-TCS tests.

n=3000		Prediction	
		Positive	**Negative**
Actual	Positive	574 (True Positive: TP)	19 (False Negative: FN)
	Negative	26 (False Positive: FP)	2381 (True Negative: TN)

$$\mathrm{F1-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

where Recall = TP/(TP + FP) and Precision = TP/(TP + FN).

The F1-score expresses the harmonic mean of precision and recall, and shows the predicted results performance of a deep learning model accurately. In this paper, we compute the accuracy, recall, precision, and F1-score values of all five application labels.

Finally, Fig. 6, Fig. 7 and Fig. 8 represent that the performance of the accuracy and elapsed time for the different schemes (the proposed deep RNN-TCS and the traditional deep CNN-TCS). The CNN model used in CNN-TCS consists of two convolutional layers, a maxpooling layer, and finally an output layer. Like RNN-TCS, we use the 'uniform' distribution for the model parameter initializer. The number of CNN model filters used in the experiment is the same as the payload size of the dataset used for training. Also, the size of the kernel is $\mathbf{3} \times \mathbf{3}$ and the output size of the convolutional layers maintains as the input size by using the 'same' padding method. The activation function is 'softmax', the optimization type is 'adam', and the batch size is 100. It should be noted that the elapsed time is the sum of the preprocessing time for generating the appropriate data for each model and the time taken to train the CNN and LSTM model. As shown in Fig. 6 and Fig. 7, as the number of applications increases, the proposed scheme, which utilizes streaming training data, achieves almost 96%-99% accuracy and 0.995-0.998 F1-score, whereas the conventional deep CNN-TCS is 91%-96% accurate and 0.926-0.948 F1-score, for a nearly 5% accuracy gap and 0.06 F1-score gap. The elements of the application types according to the number of applications is summarized in Table 6. Furthermore, with regard to elapsed time, as depicted in Fig. 8, the proposed scheme is almost 500 times faster than the conventional scheme, because the conventional scheme reads arbitrary packets of each application flow in the data preprocessing step. On the other hand, the proposed scheme

Table 6. Elements of application types with the number of applications

Number of apps.	Set of apps.
2	{BitTorrent, Web}
3	{BitTorrent, Web, Skype}
4	{BitTorrent, Web, Skype, SSH}
5	{BitTorrent, Web, Skype, SSH, RDP}

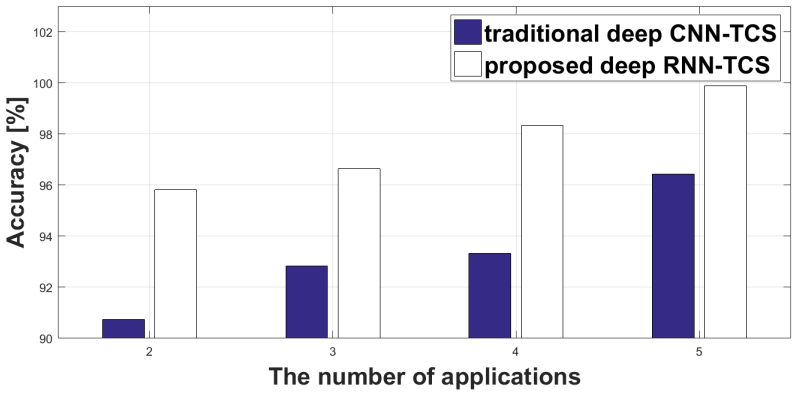


Fig. 6. Performance evaluations for accuracy of the proposed deep RNN-TCS vs. the traditional deep CNN-TCS

reads and generates data in each application flow unit. In particular, the elapsed time of RNN-TCS is faster than that of CNN-TCS because CNN-TCS in the pre-processing process converts the payload into an image after reading all data sets of network traffic. On the other hand, in the case of RNN-TCS, the network traffic dataset is extracted as the number of packets per flow, and the payload is converted to an image only for the extracted dataset. Here, the overall tendency of both schemes is that as the number of applications increases, because of the increased training data for classification, the overall accuracy is improved at the cost of more elapsed time.

In addition, recently, network traffic classification schemes using various machine learning methods have been actively studied. Parsaei et al. [31] described a method of classifying network traffic using the four neural network models (Feedforward Neural Network, Multi-layer Perceptron, Levenberg-Marquardt, and Naive Bayes) in Software-Defined Networking. The four neural network models show accuracy of 95.6%, 97%, 97% and 97.6%, respectively. Parsaei's scheme performed the classification based on the header information as well as the payload one. But, our RNN-TCS performs it based only on the packet payload. It achieves a 99% accuracy, so that it is better than the above four neural network models.

Wang et al. [42] worked on classifying malware traffic and normal traffic using a CNN model. They preprocessed all of the header and payload information of the network traffic into images for input data from the CNN model. The CNN model is trained from the imaging dataset, and the accuracy of malware traffic and normal traffic classification was almost 100%. But, the binary classification of the Wang's scheme is the simplest kind

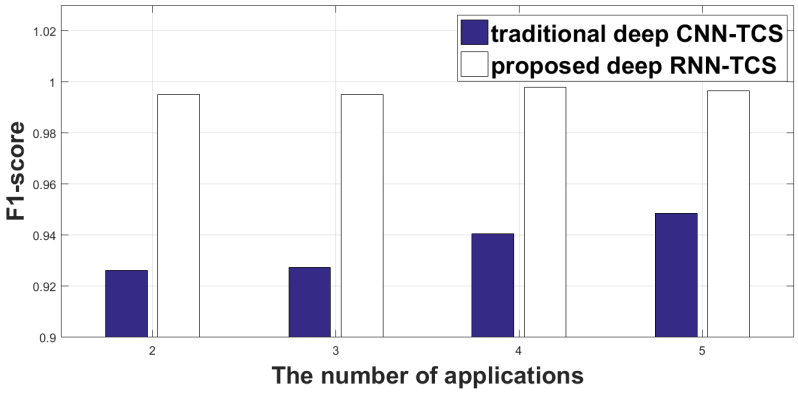


Fig. 7. Performance evaluations for F1-score of the proposed deep RNN-TCS vs. the traditional deep CNN-TCS

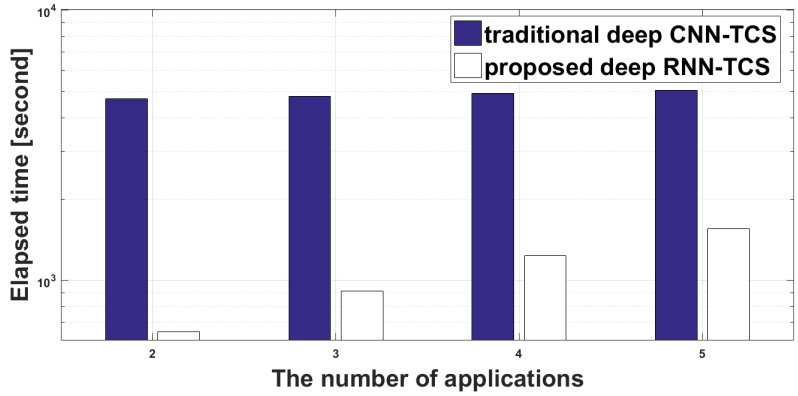


Fig. 8. Performance evaluations for preprocessing elapsed time of the proposed deep RNN-TCS vs. the traditional deep CNN-TCS

of machine learning problem. On the other hand, our proposed RNN-TCS is not binary classification, but multiple classification.

In our previous work [23], we collected payload-based network traffic which excluded the TCP/UDP headers, and used CNN and ResNet models for network traffic classification method. According to the network traffic classification results, the F1-score values for CNN and ResNet models are 0.948 and 0.969, respectively. The proposed scheme in this paper is also payload-based network traffic classification, but it uses the RNN model unlike previous work. We experimented with a comparison of our RNN-based scheme with the previous one. As a result, our RNN-based scheme outperforms the previous one.

6. Discussion

When new traffic data are generated from a new application (or protocol), the proposed traffic classification scheme does not classify them correctly, since the new data were not used to train the model. Even with traditional methods including DPI, however, it is impossible to know new applications in advance and the traffic from the new application will be unclassified with the methods. Rather, the proposed method can better respond to traffic generated by a new application, since a manager only creates a new label for the new application data, and retrain the model with the new training data augmented with the new traffic images and label. The DPI-based traditional methods are more difficult because it must come up with new rules (e.g., a specific values or structure on headers or payload) for the new application data.

From a system perspective, the proposed scheme includes the ability to collect data about the new application with a cloud computing system in a centralized manner, update the model through retraining with the new application data, and apply the results to edge computing systems in a distributed manner.

Regarding the classification of encrypted packets, much works have been already studied [34]. In case the packets are encrypted, as we can see from the previous studies, it should be possible measures 1) to add metadata to encrypted packets in the preprocessing phase, and utilize them in the classification phase, 2) to utilize the various data interpolation (Nearest value based, Bi-linear Interpolation based, Bicubic based) schemes to process data packets with various lengths to a fixed size, or 3) to extract the features through the packet header and encrypted payload using deep packet toolkit with deep learning. Therefore, by using such extended measures, classification of encrypted packets can be performed with the deep learning-based scheme proposed in this paper.

7. Conclusion

In this paper, a RNN-based traffic classification scheme (RNN-TCS) is proposed to classify applications from traffic patterns. RNN-TCS is a payload-based network classification method, so that it can classify network traffic well even though dynamic IP addresses or port numbers are used on the packet header. In addition, the architecture of the proposed technique is a hybrid edge computing system capable of parallel execution by performing data preprocessing and model training in the cloud and classifying network traffic with the trained model in edge computing. For evaluation, traffic packets measured at Universitat Politecnica de Catalunya Barcelonatech for our training data are utilized and the proposed scheme is implemented using a deep LSTM system. Finally, it is shown that the proposed deep RNN-TCS achieves almost 96%-99% accuracy and 0.995-0.998 F1-score with low elapsed time. In the future, we plan to classify more than 100 services based on deep RNN-TCS in edge computing studied in this paper. Also, based on each classified service, we will conduct a research to control Quality of Service using Software-Defined Networking.

Acknowledgments. This research was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIP) (No. CRC-15-05-ETRI).

References

1. Aceto, G., Ciuonzo, D., Montieri, A., Pescapé, A.: Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. IEEE Transactions on Network and Service Management 16(2), 445–458 (2019)
2. Bernaille, L., Teixeira, R., Salamatian, K.: Early application identification. in Proceedings of the 2006 ACM CoNEXT Conference pp. 1–12 (2006)
3. Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrad-solano, F., Caicedo, O.M.: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications (2018)
4. Carela-Español, V., Bujlow, T., Barlet-Ros, P.: Is Our Ground-Truth for Traffic Classification Reliable? In Proc. of the Passive and Active Measurements Conference (PAM'14) (Mar 2014)
5. Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks pp. 240–254 (1994)
6. Deng, W., Xui, J., Zhao, H.: An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem. IEEE Access pp. 20281–20292 (2019)
7. Deng, W., Zhao, H., Yang, X., Xiong, J., Sun, M., Li, B.: Study on an improved adaptive PSO algorithm for solvingmulti-objective gate assignment. Applied Soft Computing pp. 288–302 (2017)
8. Deng, W., Zhao, H., Zou, L., Li, G., Yang, X., Wu, D.: A novel collaborative optimization algorithm in solving complex optimization problems. Soft Computing pp. 4387–4398 (2017)
9. Erman, J., Mahanti, A., Arlitt, M., Williamson, C.: Identifying and discriminating between web and peer-to-peer traffic in the network core. in Proceedings of the 16th International Conference on World Wide Web pp. 883–892 (2007)
10. Finsterbusch, M., Richter, C., Rocha, E., Muller, J.A., Hanssgen, K.: A survey of payload-based traffic classification approaches. IEEE Communications Surveys & Tutorials pp. 1135–1156 (2014)
11. Gupta, P., McKeown, N.: Algorithms for packet classification. IEEE Network: The Magazine of Global Internetworking pp. 24–32 (2001)
12. Haffner, P., Sen, S., Spatscheck, O., Wang, D.: automated construction of application signatures. in MineNet (2005)
13. Hatcher, W.G., Yu, W.: A survey of deep learning: platforms, applications and emerging research trends. IEEE Access. pp. 24411–24432 (2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition 2016 pp. 770–778 (2016)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation pp. 1735–1780 (1997)
16. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. in Proceedings of the 14th International Joint Conference on Artificial Intelligence pp. 1137–1143 (1995)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (NIPS 2012) pp. 1097–1105 (2012)
18. Lan, K., Heidemann, J.: On the correlation of Internet flow characteristics. Technical Report ISI-TR-574, USC/Information Sciences Institute (2003)
19. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. in Proceedings of the IEEE vol. 86(11), 2278–2324 (Nov 1998)
20. LeCun, Y., Bengio, Y., Hinton, G.: Deep Learning. Nature International Journal of science pp. 436–444 (May 2015)
21. Li, F., Kakhki, A.M., Choffnes, D., Gill, P., Mislove, A.: Classifiers unclassified: An efficient approach to revealing IP traffic classification rules. in proceedings of the 2016 Internet Measurement Conference pp. 239–245 (2016)

22. Li, L., Kianmehr, K.: Internet traffic classification based on associative classifiers. IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) pp. 263–268 (2012)
23. Lim, H., Kim, J., Heo, J., Kim, K., Hong, Y., Han, Y.: Packet-based Network Traffic Classification Using Deep Learning. In Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) pp. 46–51 (2019)
24. Liu, Y., Wang, X., Zhai, Z., Chen, R., Zhang, B., Jiang, Y.: Timely daily activity recognition from headmost sensor events. ISA Transactions pp. 379–390 (2019)
25. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J.: Network traffic classifier with convolutional and recurrent neural networks for internet of things. IEEE Access. pp. 18042–18050 (2017)
26. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow clustering using machine learning techniques. in Passive and Active Network Measurement. Berlin, Heidelberg: Springer Berlin Heidelberg pp. 205–214 (2004)
27. Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y.P.: A machine learning approach for IoT device identification based on network traffic analysis. in proceedings of the Symposium on Applied Computing pp. 506–509 (2017)
28. Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., Khudanpur, S.: Extensions of recurrent neural network language model. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP) pp. 5528–5531 (2011)
29. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys Tutorials pp. 56–76 (2008)
30. Park, J.: Statistics signiture based application traffic classification. Korea Communication Journal 34, 1234–1244 (Nov 2009)
31. Parsaei, M.R., Sobouti, M.J., Khayami, S.R., Javidan, R.: Network traffic classification using machine learning techniques over software defined networks. International Journal of Advanced Computer Science and Applications (2017)
32. Pedregosa, F., Varoquaux, G., Gramfort, A.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research pp. 2825–2830 (2011)
33. Powers, D.M.W.: Evaluation: From precision, recall and f-measure to ROC, informedness, markedness & correlation. Journal of Machine Learning Technologies pp. 37–63 (2011)
34. Rezaeij, S., Liu, X.: Deep learning for encrypted traffic classification: An overview. IEEE Communications Magazine pp. 76–81 (2019)
35. Risso, F., Baldi, M., Morandi, O., Baldini, A., Monclus, P.L.: Lightweight, payload-based traffic classification: An experimental evaluation. IEEE International Conference on Communications pp. 5869–5875 (2008)
36. Sak, H., Senior, A., Beaufays, F.: Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Proceedings of the Annual Conference of the International Speech Communication Association(INTERSPEECH) pp. 338–342 (Jan 2014)
37. Shafiq, M., Yu, X., Wang, D.: Network traffic classification using machine learning algorithms. Advances in Intelligent Systems and Computing pp. 621–627 (2018)
38. Singh, H.: Performance analysis of unsupervised machine learning techniques for network traffic classification. 2015 Fifth International Conference on Advanced Computing & Communication Technologies pp. 401–404 (2015)
39. Trivedi, U., Patel, M.: A fully automated deep packet inspection verification system with machine learning. 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (Nov 2016)
40. Udrea, O., Lumezanu, C., Foster, J.S.: Rule-based static analysis of network protocol implementations. Information and Computation p. 130–157 (2008)
41. Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., Zhu, M.: Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access 6, 1792–1806 (2018)

42. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware trac classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN) pp. 712–717 (2017)
43. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware Traffic Classification Using Convolutional Neural Network for Representation Learning. IEEE ICOIN 2017 (2017)
44. Yu, C., Lan, J., Xie, J., Hu, Y.: QoS-aware traffic classification architecture using machine learning and deep packet inspection in SDNs. Procedia Computer Science pp. 1209–1216 (2018)
45. Zander, S., Nguyen, T., Armitage, G.: Automated traffic classification and application identification using machine learning. The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) pp. 250–257 (2005)
46. Zhang, J., Xiang, Y., Wang, Y., Zhou, W., Xiang, Y., Guan, Y.: Network traffic classification using correlation information. IEEE Transactions on Parallel and Distributed Systems pp. 104–117 (2013)
47. Zhang, Y., Breslau, L., Paxson, V., Shenker, S.: On the characteristics and origins of internet flow rates. SIGCOMM 02 Proceedings of the 2002 conference on Applications pp. 309–322 (2002)
48. Zhao, H., Liu, H., Xu, J., Deng, W.: Performance Prediction Using High-Order Differential Mathematical Morphology Gradient Spectrum Entropy and Extreme Learning Machine. IEEE Transactions on Instrumentation and Measurement pp. 379–390 (2019)
49. Zhao, H., Zheng, J., Xu, J., Deng, W.: Fault Diagnosis Method Based on Principal Component Analysis and Broad Learning System. IEEE Access pp. 99263–99272 (2019)

Kwihoon Kim is currently a professor in the Department of Artificial Intelligence Convergence Education, Korea National University of Education (KNUE), South Korea. He received the B.S, M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea in 1998, 2000 and 2019, respectively. He worked in LG DACOM 2000 2005. From 2005 to 2020, he was a Principle Researcher with Electronics and Telecommunications Research Institute (ETRI). He is an editor and rapporteur of ITU-T SG11 since 2006. His interested fields are Fog/edge computing, Internet of Things, 5G/IMT2020, deep learning, machine learning, reinforcement learning, GAN and knowledge-converged intelligent service.

Joohyung Lee (S'09–M'14–SM'19) is currently an Assistant Professor in the School of Computing, Gachon University, South Korea. He received the B.S, M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2008, 2010 and 2014, respectively. From 2012 to 2013, he was a Visiting Researcher with the Information Engineering Group, Department of Electronic Engineering, City University of Hong Kong, Hong Kong. From 2014 to 2017, he was a Senior Engineer with Samsung Electronics. He has contributed several articles to the International Telecommunication Union Telecommunication (ITU-T) and the 3rd Generation Partnership Project (3GPP). His research work is resource management at the intersection of mobile systems and machine learning focusing on edge computing architectures to optimize the trade-off between latency, energy, bandwidth and accuracy for data analytics.

Hyun-Kyo Lim received the B.S. degree in computer science and engineering and the M.S. degree in computer science engineering from the Korea University of Technology

and Education, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Interdisciplinary Program in Creative Engineering. He studied mobility management during his master course and he especially researched distributed mobility management in software-defined networking. He is studying deep learning and reinforcement learning during his doctoral studies. He is also exploring ways to apply deep learning and reinforcement learning to the network and is working on applying deep learning and reinforcement learning to a variety of applications.

Se Won Oh received the B.S.(1999) and the M.S. degrees(2001) from Pohang University of Science and Technology (POSTECH), and received the Ph.D.(2018) from Chungnam National University (CNU), Daejeon, South Korea, respectively. Since joining Electronics and Telecommunications Research Institute (ETRI) in 2001, he is currently a Principal Researcher working for Knowledge-converged Super Brain (KSB) Convergence Research Department in ETRI, Daejeon, South Korea. He has been involved in several research projects on software platform (such as RFID Event Management, USN Middleware, Internet of Things Platform) which integrates legacy applications with various data resources and sensors. He has made several contributions in international standardization activities, particularly on automatic identification and data capture techniques of JTC 1/SC 31. His recent interested areas include machine learning application, anomaly detection, and knowledge-converged intelligent service solutions.

Youn-Hee Han received the B.S. degree in mathematics and the M.S. and Ph.D. degrees in computer science and engineering from Korea University, Seoul, South Korea, in 1996, 1998, and 2002, respectively. From 2002 to 2006, he was a Senior Researcher with the Next Generation Network Group, Sam- sung Advanced Institute of Technology. Since 2006, he has been a Professor with the School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan, South Korea. Since 2002, his activities have been focusing on mobility management, media independent handover, and cross-layer optimization for efficient mobility support. He has published approximately 250 research articles on the theory and application of mobile computing and has filed 40 patents on information and communication technology domain. His current research interests include theory and application of computer networks, including protocol design and math- ematical analysis, mobile sensor/actuator networks, social network analysis, machine learning, deep learning, and reinforcement learning. He has made several contributions in IETF and IEEE standardization. He has served as the Co-Chair for working group in the Korea TTA IPv6 Project Group. He has been serving as an Editor for the Journal of Information Processing Systems since 2011.

Received: April 24, 2020; Accepted: July 25, 2021.

Computer Science and Information Systems 19(1):185–204 https://doi.org/10.2298/CSIS210201047J

Building of Online Evaluation System Based on Socket Protocol

Peng Jiang[1], Kexin Yan[2,*], Haijian Chen[3] and Hai Sun[4]

[1] Jingan Branch Campus, Shanghai Open University,
Shanghai 200040, China
jzhpmail@163.com

[2] Management School, Shanghai University of International Business and Economics,
Shanghai 201620, China
ykx980219@163.com

[3] Shanghai Academic Credit Transfer and Accumulation Bank for Lifelong Education,
Shanghai 200433, China
xochj@sou.edu.cn

[4] School of Management, Fudan University
Shanghai 200433, China
sunhai@fudan.edu.cn

Abstract. As an important part of the evaluation reform, online evaluation system can effectively improve the efficiency of evaluation work, which has been paid attention by teaching institutions. The online evaluation system needs to support the safe and stable transmission of information between the client and the server, and socket protocol establishes the connection through the listening port, which can easily carry out the message transmission and process control. Because it can well meet the construction requirements of online evaluation system, it is applied in our study. The building of online evaluation system based on socket protocol includes the function design of students and teachers, data flow design, evaluation difficulty grading design and system implementation. The system uses Java language and MVC mode for development, which has good scalability and platform-independence. It realizes the paperless examination process and greatly reduces the workload of teachers. The contribution of this paper is mainly reflected in two aspects. One is to explore the construction of an online evaluation system based on the socket protocol, and it provide an Asynchronous IO technical solution for the network communication between the student and the server, which provides a reference for the development of similar systems. The second is to give the realization method of the difficulty classification of the evaluation, and classify the difficulty of the test questions, which lays the foundation for carrying out personalized testing and evaluation.

Keywords: online evaluation system; socket protocol; MVC mode; Asynchronous IO.

* Corresponding author

1. Introduction

With the promotion of the Internet and the development of educational information technology, online education has emerged. This new teaching method has the advantage of unlimited time and place [21], and it is welcomed by more and more learners. At the same time, online evaluation system has been rapidly promoted in many teaching institutions because of its convenient test method and paperless characteristics [15]. The research of online evaluation system has also become one of the hot spots of online education research, which has attracted the attention of many scholars [1][2][10][17]. However, because the online evaluation system needs to consider not only the concurrent access of users, but also the stability of data transmission and the security of the system [16], its construction presents a certain degree of difficulty, which is higher than that of the common management information system.

In order to meet the requirements of concurrency and security of online evaluation system, this paper presents a construction method of online evaluation system based on socket communication protocol [3][14]. The system completes the communication between terminal and server through asynchronous IO mode, which effectively ensures the safe transmission of data. In addition, the system is realized by adopting the MVC design pattern based on the J2EE framework [13][22][23], and it has good cross-platform and adaptability.

2. Related Research

With the advancement of teaching reform and the development of Internet, online evaluation system came into being. For example, the University of Valladolid system, which was born in 1995, it undertook the service of ACM International Undergraduate Program Design Competition [19]. Since then, similar systems have been emerging and their functions have become more and more perfect. In 2003, the blackboard platform launched for the teaching of Chinese colleges and universities has the functions of uploading courseware, correcting homework, testing and scoring. It plays an important role in the teaching of colleges and universities and is welcomed by many colleges and universities. Since then, the online evaluation system has obtained a good opportunity for development, and many online education testing platforms have appeared at home and abroad, such as Coursera, Udacity, edX and so on [6].

Because online evaluation system has many advantages, some colleges and universities have developed their own online examination system, which can realize the random combination of test papers and real-time performance presentation. For example, Tsinghua University and East China Normal University began to try to use Internet technology to change students' learning and examination methods. However, the original technology of the evaluation system is not perfect, and the function of the system is limited to simple operation and examination. Since then, more and more scholars began to study the design of online evaluation system, and they think about how to make this system better meet the needs of users. Kang et al. (2004) used J2EE technology to design the evaluation system. Their research gave the method of selecting test questions to form a test and showed the advantages of online classroom. The

practice results show that J2EE can support distributed applications and improve the evaluation efficiency [7]. Wang (2014) developed an online examination system based on IDC, which realized the real paperless online examination [20]. Although the above system has promoted the research and development of online examination to a certain extent, it is still not complete in function and lack of strong universality. In order to solve these problems, scholars have carried out more in-depth research on online evaluation system. Chen (2020) took the design of "data structure" course as an example. Their research is based on B/S mode, and used the SpringBoot framework of J2EE to develop the online evaluation system. The system has passed the stress test and stability test, which proves that the system can effectively improve the efficiency of online evaluation [4]. Zhong et al. (2020) analyzed the requirements of online evaluation system, which including front desk service module, background management module and evaluation module, and their study introduced the whole process of online evaluation system [29]. Zhang et al. (2018) designed an online evaluation system based on struts and Ajax technology to solve the problem of low openness of the current online evaluation system. The evaluation system has good openness and can be used to assist course teaching [26].

3. Functional Design

The development of the system follows the idea of multi-layer architecture [18], and it uses configuration files to save general configuration information. The reading and parsing operation of information is performed by the reading tool class Config class. Users can invoke various commands in the program to change the parameters and attributes of each component. The system interface is implemented by JFrame window class, JFrame can be regarded as a container, and all components used in the interface can be put into the container [25][28]. The system is divided into many functional modules, such as teachers' questions, changing questions, arranging examinations, viewing results, analyzing results, students previewing test papers, participating in examinations, querying results and so on.

The whole server uses socket communication protocol. When the client of the terminal calls the ExamServiceAgentImpl interface to access the service ExamServiceServer, the student side will pass the command name, parameter type, parameter value and an empty Sid to the server through an encapsulation class request. At this time, the server will assign a unique Sid to each user whose Sid is empty, and call ExamServiceImpl to respond to the request. The response of the above request is realized by reflection, so an encapsulation class response is returned to the student side, which contains the return value of the command and the new Sid. After that, all requests transmitted by the user will be bound to this SID, and the server will find the corresponding function implementation class of the core business according to the SID to implement the corresponding request and return the data. The structure diagram of the server is shown in Figure 1.

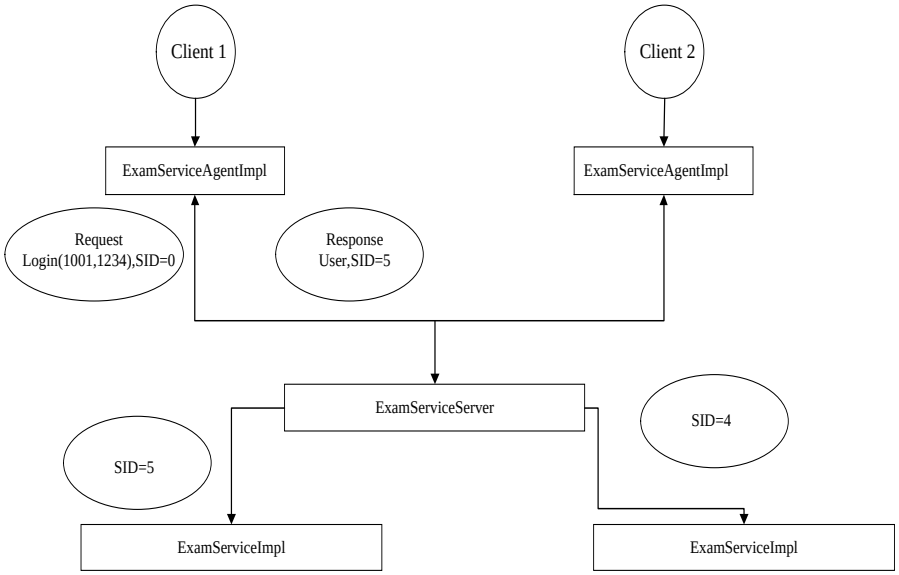


Fig. 1. The structure diagram of Server-side

3.1. Functional Design for Student

The main operations of the student side include login, preview, answer and submit. First, students start the student terminal, and input the user's name and password in the authentication interface, and enter the student terminal interface to prepare for the exam after the verification is successful. At this time, students can only preview the test paper. When the test time arrives, students can click the start button to answer the questions. In the whole test process, students must complete the single choice questions, multiple choice questions, judgment questions, fill in the blanks and question and answer questions within the specified time, and then submit the test paper. After the whole test, students will get the test scores. If the test paper is not submitted in the specified time, the system will force students to submit the test paper.

The entire student terminal adopts the MVC design pattern [9][27]. When the user makes a request from each Frame every time, various commands in ExamService are called through Client Context, and the function is realized by ExamServiceImpl class. The EntityContext class is the tool class of question bank and student information import, which runs automatically when the program starts. After the user logs in successfully, it reads the properties setting file through the Config class before the exam starts, and imports test questions and student information into the program. The student side structure is shown in Figure 2.

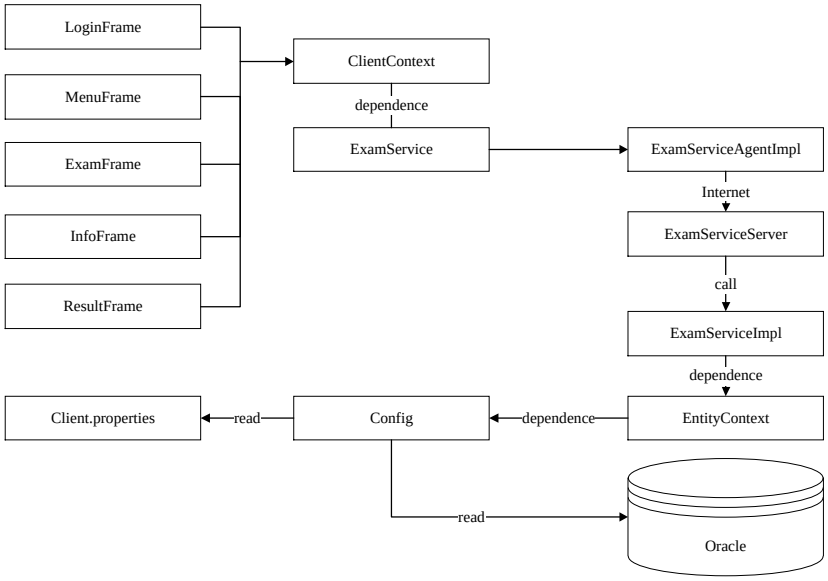


Fig. 2. The structure diagram for student

3.2. Functional Design for Teacher

The main operations of the teacher side include login, test paper making, examination arrangement and score inquiry. The teacher enters the user's name and password in the authentication interface, and enters the teacher interface once the authentication is successful. Among, test paper making includes adding test questions, importing test question set in batch and modifying test questions. Through the above functions to complete the production of the whole set of test papers and add the test questions to the test database.

Examination arrangement can set examination information such as test paper name, test time, the start time, number of questions and total score of test paper. It can save this information into the property's configuration file. After setting up the examination information, teachers will enter the interface of adding student information and add the student information involved in the examination into the user table of the database.

Score inquiry allows teachers to query the results of all students. Teachers can also filter the query results according to the specified conditions if further requirements are required. In addition, teachers can enter the interface of grade analysis to check the correct situation of all students. In this interface, teachers can check the correct rate of each type and difficulty of questions to improve the teaching plan in the future. The basic flow chart is shown in Figure. 3.

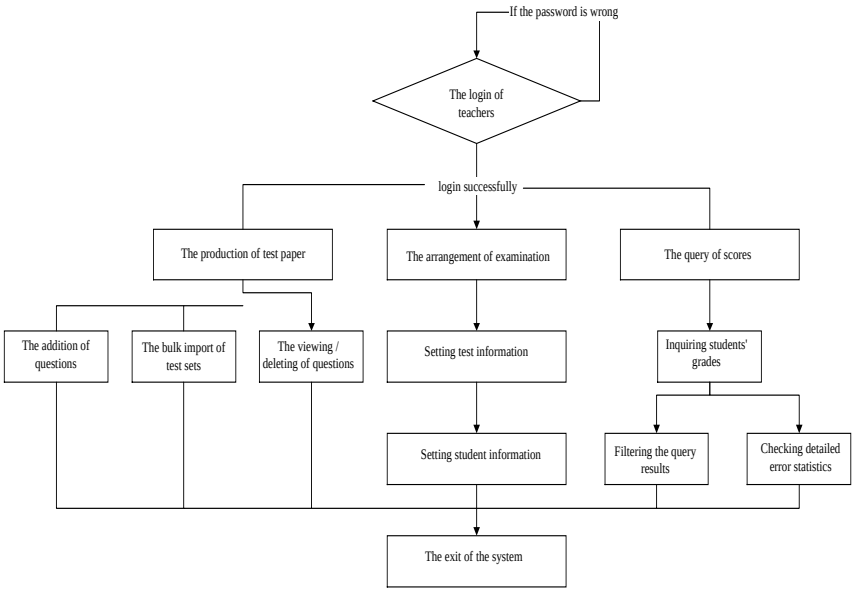


Fig. 3. The basic flow chart for teacher

The teacher side mainly completes the reading and editing of the database through the interface, and sets the examination information, database IP address and port, so it needs to have permission to operate the database and properties configuration file.

4. Analysis of Data Flow and Asynchronous IO

4.1. Analysis of Data Flow

The data flow chart is a tool to describe the data flow of the system, which uses the structured analysis method of top-down, layer-by-layer decomposition and step-by-step refinement [5][8][11]. At the same time, it uses a hierarchical DFD diagram to represent the transmission process between various data. The DFD diagram, also known as data flow diagram, is a tool to explain a series of processes such as system data input, output, storage and processing, which can help programmers to achieve effective cooperation between various modules [12][24].

In order to show the details of teachers and students in the whole system. On the basis of the top-level diagram, we decompose it from top to bottom and get the classification data flow chart of the online examination system. The classification data flow chart of the system is shown in Figure. 4.

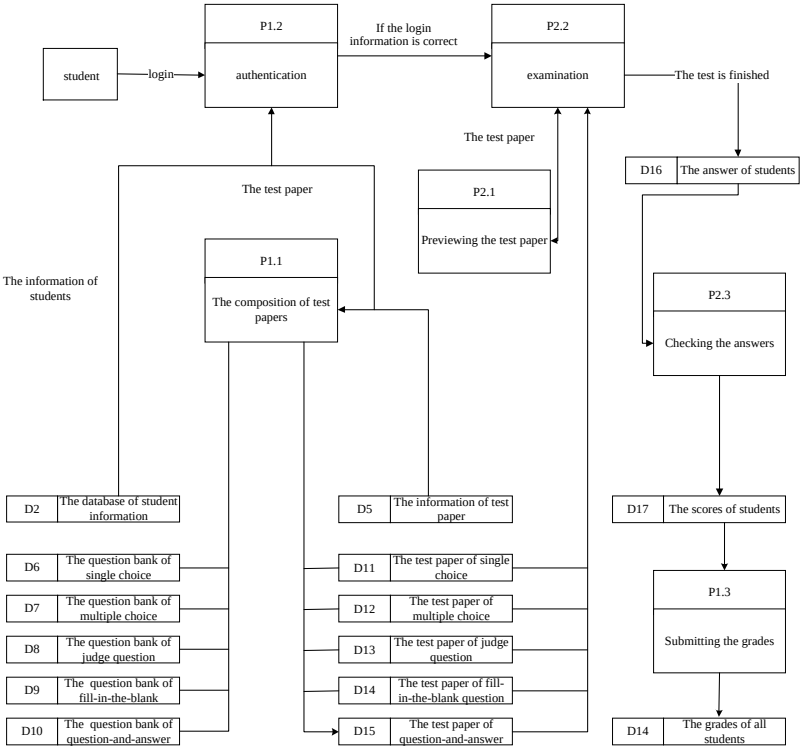


Fig. 4. The detailed flow chart for student and server

It can be seen from Figure. 4 that the online evaluation system is mainly divided into three sub-modules. That is, the student-side module, the server-side module and the teacher-side module.

4.2. Analysis of Asynchronous IO

Asynchronous IO is relative to synchronous io. The difference between asynchronous IO and synchronous IO is that when a thread performs IO operation, the operating system does not suspend the current thread operation. Instead, after the Input/Output instruction is executed, the operating system continues to let the current thread execute the next instruction. When the Input/Output operation is completed, the Input/Output thread will be notified through an event, and the thread will process the response event after receiving the notification. Specifically, the synchronous type allows multiple tasks to be completed through multiple threads, while the asynchronous type uses one thread to complete multiple tasks. When it encounters an I/O operation, it still lets the thread continue to execute other instructions, and notifies the thread to schedule a response event after the I/O is completed. The process of asynchronous IO in our study is shown in Figure 5.

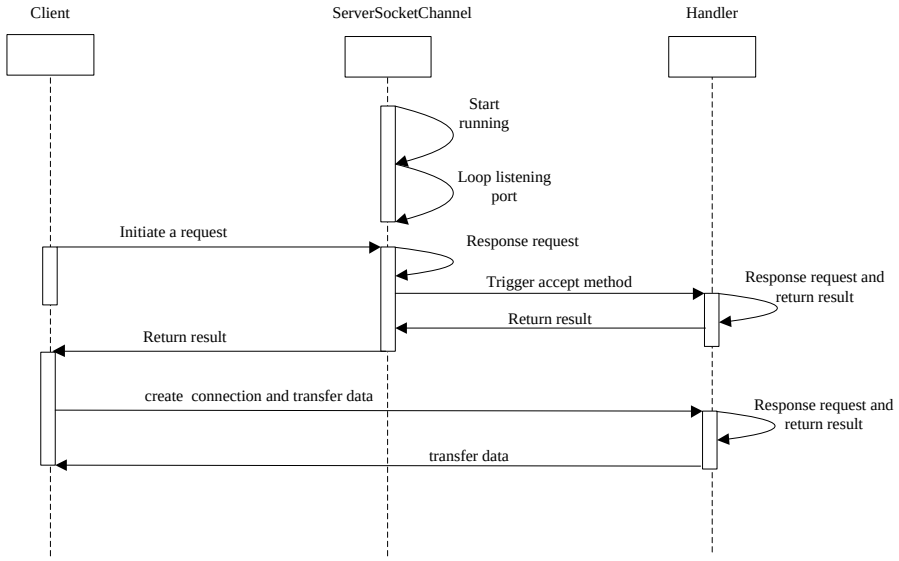


Fig. 5. The process of asynchronous IO

5. System Implementation

5.1. Implementation of Login Function

When the program starts, the system automatically starts the loadUsers method in the EntityContext to read all user names and passwords from the database. After the user entering the user's name and password in the login interface and clicking the login button, the system will call the GetId and GetPwd methods of the ClientContext (interface controller) to obtain the user's name and password entered by the user. Then, the user's name and password are introduced into getUserin the ExamServiceand compared with the set of users read from the database by the preprocessing utility class. If the User name and password are correct, the User data is returned. Finally, the User data is passed into the MenuFrame (student or teacher interface) and the interface is updated. At this point, the login of user is successful and the next operation can start. The whole process is shown in Figure. 6.

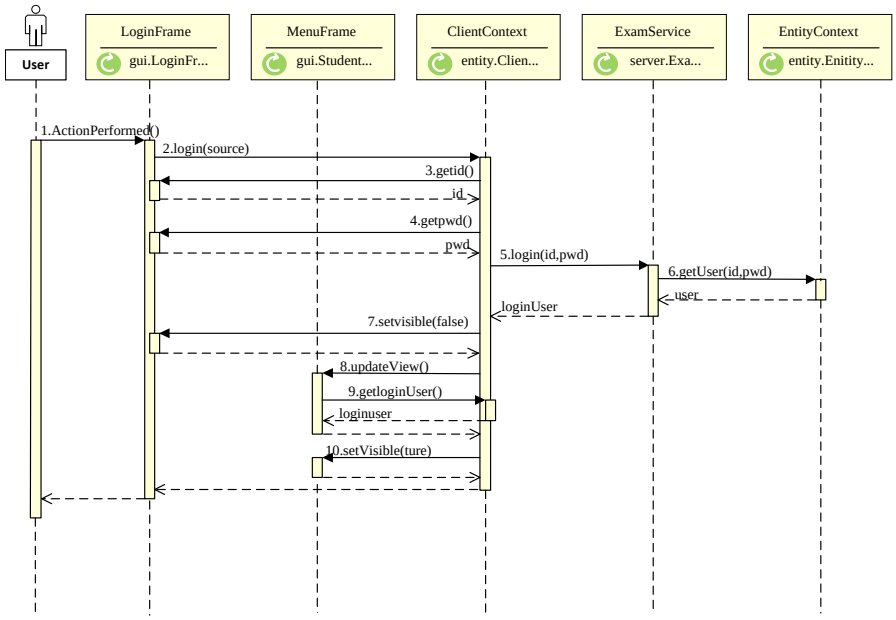


Fig. 6. The process of login function

Taking the login of student as an example, the procedure is executed as follows:

First, the EntityContext class is responsible for reading the user's name and password from the database and adding them into a collection. The steps are as follows:

- Step 1: Read student information from the database and add it to the array.
- Step 2: Read the information in Properties configuration file through the Config class and connect to the database.
- Step 3: Create SQL statement and pass it to the database to execute.
- Step 4: Get the query results of SQL.
- Step 5: Search all user information through SQL query statements and convert it to user data.
- Step 6: Add User data to student user's collection list.
- Step 7: Disconnect from database.

After that, the program obtains and transmits the user's name and password entered by the student through the ClientContext class, and the sample of key codes are follows.

```
public void login (JFrame source) {
     int id = loginFrame.getId();
     // Get the user's name from the login interface
     String pwd = loginFrame.getPwd();
     // Get the password from the login interface loginUser =
     service.studentlogin(id, pwd);
     // Check the user's name and password correctly through the
     relevant function of the core program
     studentMenuFrame.updateView();
     // Update user information on the student interface if successful
     studentMenuFrame.setVisible(true);
     // Enter the student interface
}
```

Finally, the ExamService implementation class is responsible for verifying that if the user's name password is correct. The steps are as follows:

- Step 1: Read the user's name entered by clientcontext.
- Step 2: The dialog box saying "no user" will be popped if the user's name is not found in the user collection.
- Step 3: If the user's name and password are input correctly, the user will be set as a valid user and the test paper generation operation is started.
- Step 4: Otherwise, it displays the dialog box with information of "the password is incorrect".

5.2. Implementation of Generating Test Paper Function

After successfully logging in, students will enter the main interface of students. Various loadQuestions methods of the EntityContext (preprocessor utility class) are automatically executed by the system. All questions are read from the question bank in the database, and then are added into a HashMap with Level or Score as the Key and the collection of this type of questions as Value according to the test paper requirements. They then call the buildPaper function in ExamServiceImpl (the core business implementation class) to establish the test paper for each type of topic, and use the getQuestions function to select the question from the HashMap and add it to a QuestionInfo. Each QusetionInfo contains all the attributes and the user's answer of the question.

Finally, all QuestionInfo were formed into a set "Paper" to form the test paper, which contains the information of the QuestionInfo and its number in the set (the index of the question in the set). The whole process is shown in Figure. 7.

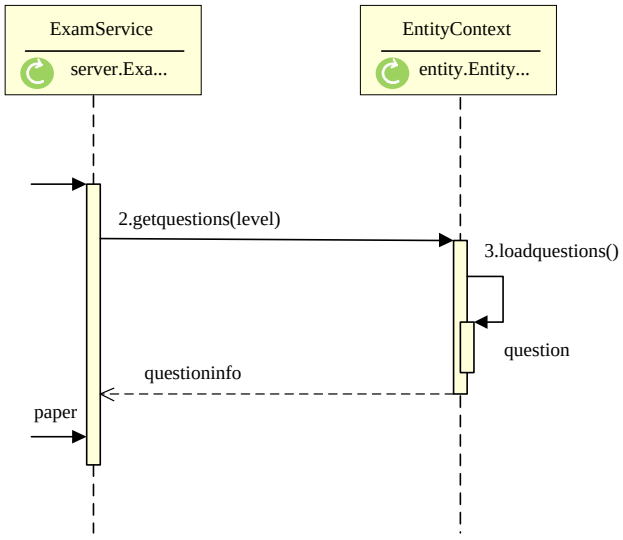


Fig. 7. The process of the function of reading question papers

Take the production of multiple-choice test papers as an example, the procedure is executed as follows:

First of all, to read the information of each question in the question bank through the EntityContext class. The steps are as follows:

- Step 1: Create a set of topics "Choice Questions" for multiple choice questions.
- Step 2: Read the information in the Properties profile through the Config class and connect it to the database.
- Step 3: Create SQL query statements and pass them to the database for execution.
- Step 4: Get the query result and convert it to an entity with multiple choice question.
- Step 5: The question is added to the choice levels (a HashMap based on the difficulty of the question), which will be used to complete the retrieval operation according to the requirements of the test paper in the future.
- Step 6: Close the connection to the database.

After reading the title, we form the test paper through the implementation class in the ExamService, and the sample of key codes are follows.

```
private List<ChoiceQuestionInfo> choicepaper = new
ArrayList<ChoiceQuestionInfo>();
// Create a set of multiple-choice questions with ChoiceQuestionInfo
private void buildChoicePaper()  {
// Read the information in Properties settings file through the Config class
and connect it to the database
  for (int level = 1; level <= 10; level++) {
  // Through the outer cycle, each difficult topic can be selected
    for (int k = 1; k <= count; k++) {
       ...// Establish a random number and get the questions from the
       HashMap, It is through the inner loop to ensure a specified number of
       questions can be obtained for each difficult topic, and then add the
       question to the test set
    }
  }
…     // Close the connection to the database
```

5.3. Implementation of Previewing Test Paper Function

When the user clicks the "preview test paper" button of the MenuFrame (student interface), the system sends the command of previewing test paper to the ClientContext (interface controller). Subsequently, the controller acquires all the test questions from the ExamService (business core logic) and converts them into strings. Finally, the string is passed into a TextArea of the preview interface (SearchQuestionFrame) and displayed for candidates to preview the test paper. The process is shown in Figure. 8.

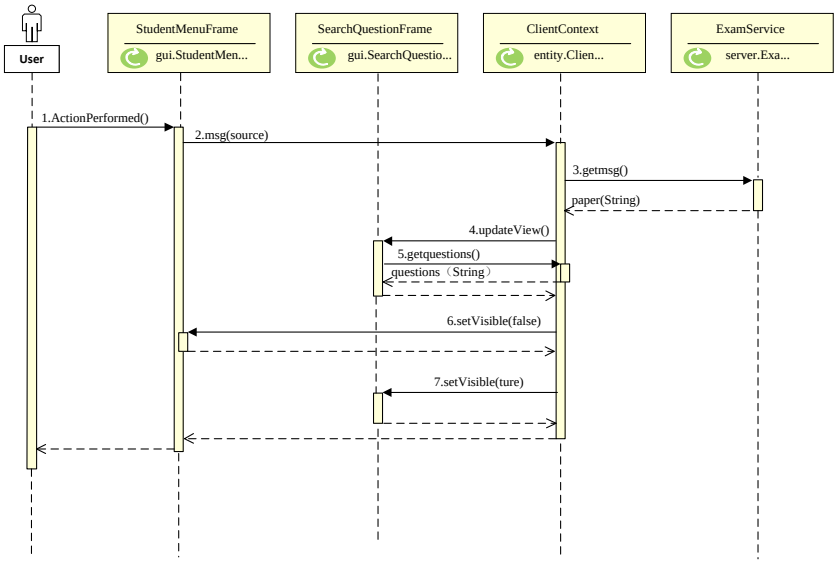


Fig. 8. The process of previewing test papers

5.4. Implementation of Examination Function

The whole examination is divided into five types: single-choice questions, multiple-choice questions, judgment questions, fill-in questions and question-answer questions. Although the type of data transmitted is different from the source of the information sent, the principle of starting the examination is the same.

When user clicks the button of starting test in the MenuFrame (the student interface), the system sends the command of starting test to ClientContext (the interface controller). If the specified test time is reached, the system sends the “start” command to ExamService (the business core program) to start the test. At this time, the timer starts, and the controller sends the getExamInfo command to the business core program to get the exam information. This information includes examination times, examination subjects and the number of questions.

Then the command of “getSingleChoiceQuestions” is sent to the business core program to get the test information of the first topic of the single topic. Finally, the first question and the examination information will be introduced into the single-choice examination interface and the examination interface will be updated, and the students will enter the single-choice interface to start answering questions.

The interface of the five types of questions has a common timer thread. When the timer is 0, the user's current information is saved and the user's test paper is forced to be submitted. The process is shown in Figure. 9.

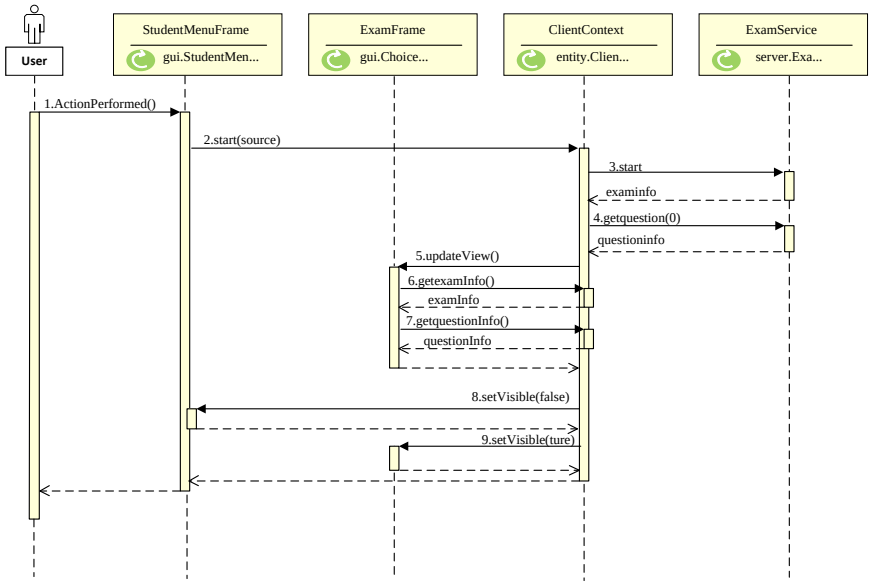


Fig. 9. The process of the function of starting examination

Taking the "start the single-topic examination" as an example, it will send a request to the core business by the controller, and the sample of key codes are follows.

```
public void startsinglechoice (JFrame source) {
       service.startsinglechoice();
       // Send a request of starting test
       singlechoiceQuestionInfo=service.getSingleChoiceQuestion(0);
       // Get the title information of the first single topic
       singlechoiceExamFrame.updateView();
       // Update the Test Interface of Single Topic
       ...
           // Leave the student interface and display the exam interface
       startTimer(); // Start timer thread
}
```

The steps of the timer thread are follows.

- Step 1: Get the exam time, the present time and the end time of the exam.
- Step 2: Start the thread to start the timer, calculate the remaining time of the exam and display it on the interface.
- Step 3: If the timer is timed out, the related operation is performed.

In the process of students participating in the online test, the system provides five types of test questions for students to choose according to the difficulty of the test questions. The difficulty classification of examination questions adopts fuzzy comprehensive evaluation method. In the fuzzy comprehensive evaluation, the weight will have a great influence on the final evaluation result. We use the expert estimation method to determine the weight. The steps of fuzzy comprehensive evaluation are as follows.

- Step 1: Determining the factor domain of evaluation object, $U = \{u_1, u_2, \ldots u_m\}$. there are m evaluation indexes, which indicate from which aspects we can judge and describe the evaluated object.

- Step 2: Determine the set of rating levels. The evaluation grade set is a set composed of various total evaluation results that the evaluator may make on the evaluated object, which is represented by V as follows. $V = \{v_1, v_2, \ldots v_n\}$. In fact, it is a division of the change interval of the evaluated object. Where v_i represents the ith evaluation result and n is the total number of evaluation results. The specific level can be described according to the test questions. For example, the difficulty degree of the test questions can be expressed by V, V = {very difficult, relatively difficult, medium, relatively easy, very easy}.
- Step 3: The single factor evaluation was carried out and the fuzzy relation matrix R was established. After constructing the fuzzy subset of the grade, the evaluated object should be quantified from each factor $u_i(i = 1, 2, \ldots, m)$ that is to say, the membership degree of the evaluated object to each fuzzy subset of each grade from the single factor is determined, and then the fuzzy relation matrix is obtained as follows.

$$R = \begin{pmatrix} r_{11} & r_{12} & \Lambda & r_{1n} \\ r_{21} & r_{22} & \Lambda & r_{2n} \\ M & M & O & M \\ r_{m1} & r_{m2} & \Lambda & r_{mn} \end{pmatrix} \quad (1)$$

Where $r_{ij}(i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$ represents the membership degree of an evaluated object to the V_j-level fuzzy subset from the perspective of factor u_i. The performance of an evaluated object in a factor u_i is described by fuzzy vector $r_i = (r_{i1}, r_{i2}, \ldots r_{im})$. r_i is called single factor evaluation matrix, which can be regarded as a fuzzy relationship between factor set U and evaluation set V.

- Step 4: determine the fuzzy weight vector of evaluation factors. In order to reflect the importance of each factor, each factor u should be assigned a corresponding weight $a_i(i = 1, 2, \ldots, m)$. a_i is usually required to conform to $a_i \geq 0$, $\sum a_i = 1$, and a_i represents the weight of the ith factor.

5.5. Implementation of Browsing the next Question Function

It is a process of submitting the answer and obtaining a new question that the student clicks on “previous question or next question”. For example, when a student clicks “the next question” button in the ExamFrame (exam interface), the program will save the answer of candidate into the test paper List of the ExamService (program core business) through the ClientContext (interface controller). At the same time, through the interface controller, the QuestionInfo of the next question will be returned to the test interface and the interface will be updated (if the question has been completed, the interface will be updated according to the answer selected by the user). At this point, the user can continue to complete the next question until the end of the exam. The whole process is shown in Figure. 10.

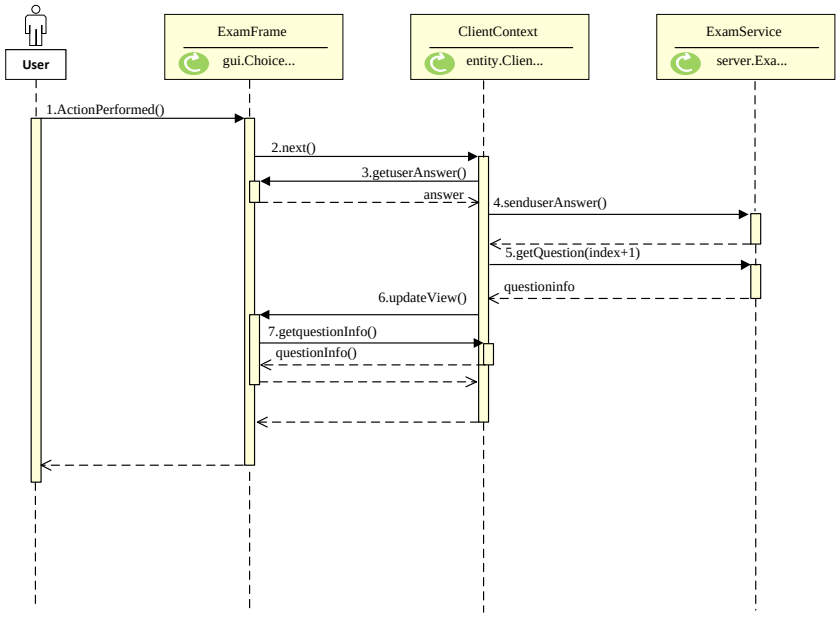


Fig. 10. The process of the function of the upper/lower topic

Next, taking the judgment question as an example, the process is shown as follows.

Firstly, the information of the next topic is obtained by "the next" command in the clientcontext class, and the sample of key codes are follows.

```
public void TFnext (JFrame source) {
       int index = tfQuestionInfo.getQuestionIndex();
             //Access to current title
       if (index + 1 == examInfo.getTFQuestionCount()) {
       return;}
       //If the title number is greater than the total title, no action is
       performed
       int answers = tfExamFrame.getUserAnswer();
             //Access to user answers
       service.sendUserTFAnswers(index, answers);
            //Transmit question numbers and user answers to business core
       programs
       tfQuestionInfo = service.getTFQuestion(index + 1);  //Access to
       information of the next question
       tfExamFrame.updateView();
       //Update interface}
}
```

After that, the interface will be updated through the ExamMenu with the following steps.

- Step 1: Update the topic information according to the current topic information.
- Step 2: If the user has done this question, the answer data will be updated according to the user's answer.
- Step 3: The update buttons. if the title is the first question, you cannot use the previous button, if the title is the last question, you cannot use the next button.

5.6. Implementation of Submitting Answer's Function

When the student press "the submit test paper" button or the timer returns to zero, the ExamService (the program's core business) will execute the commit command to calculate the candidate's scores of each type of question and save the result into the score table of the database. Next, let's take the judgment topic as example. The execution process of the whole program is as follows:

First of all, the score is calculated by the score settlement function of commit command. The steps are as follows.

- Step 1: The result of judging questions will be calculated by cyclic calculation.
- Step 2: The corresponding bonus operation will be performed according to the consistence of the user answer and the correct answer.
- Then, it submits the results through the results submission function in the commit command, and the steps are as follows.
- Step 1: Calculate the total score of the examination through adding the scores of each type of question.
- Step 2: Read the information in the Properties profile through the Config class and connect it to the database.
- Step 3: By using SQL insert command, the test results are stored in the database and then disconnected from the database.
- Step 4: Set the test status to complete.

5.7. Implementation of Server Response

The data transmission between the student side and the server side uses a simple Socket transmission protocol, in which the reflection mechanism is used to execute the corresponding command. The whole reposing process is shown in Figure. 11.

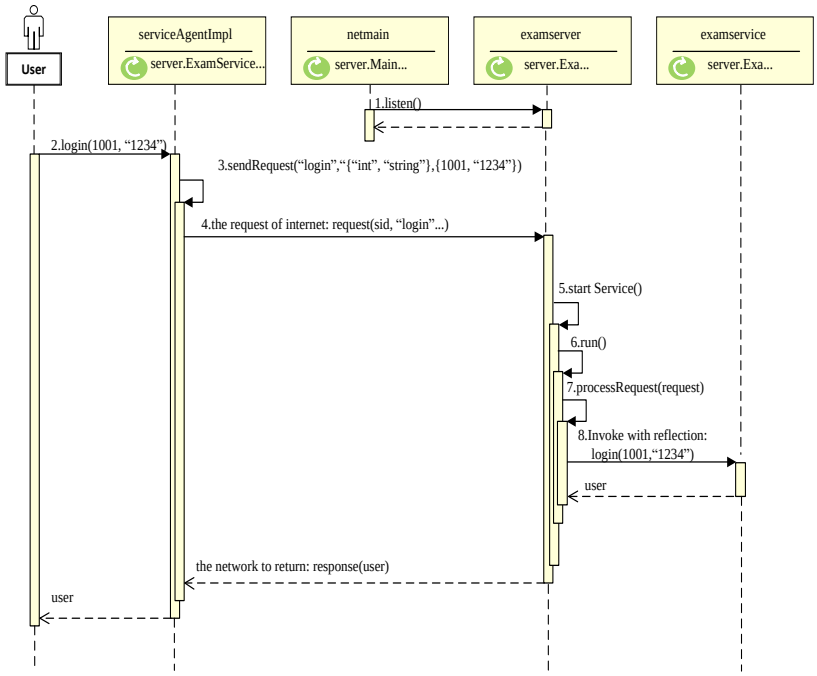


Fig. 11. The whole reposing process of the server

Before the exam, the Socket connection is established by starting the server with the netmain class (server-side main program). Then the listen thread is executed to start listening requirements. When the user passes the login (1001, '1234') command to the serviceAgentImpl (the network function implementation class of core business), the class encapsulates the login request as a request class, which contains an empty sid, the command name (login) of the command, the parameter type (the first parameter type is int, the second parameter type is String), and the parameter value (the first parameter value is 1001, the second parameter value is '1234'). Then the request is sent to the Server (the server side), which gives a sid. Then, take the sid as the key, an ExamServiceImpl (the business core function implementation class) as the Value to join a HashMap. After, the login (1001, '1234') method in the realization class of the core business function is called through the reflection mechanism, and a user parameter is returned to be encapsulated into a response return, which is parsed into a user parameter.

At this point, the transmission of the command is ended. When the user submits the request command again, the corresponding business core function implementation class can be found from the HashMap according to SID, and the corresponding command can be executed through reflection mechanism.

6. Conclusion

With the promotion of online education, online evaluation system has attracted more and more attention. This paper introduces the function design, process analysis and implementation of online evaluation system, especially the process of file transmission based on socket protocol. Practice shows that the system has good stability and security and it can meet the needs of online evaluation. In the future research, the personalized test of online test can be further improved. In the case of collecting more data, the difficulty of the test can be divided more reasonably. In general, our study provides a reference for the design of online evaluation system.

Acknowledgment. This work is supported by the project of Shanghai Philosophy and Social Sciences Plan (No.2016BGL004), National Natural Science Foundation of China (No.71971066) and Project of key education and scientific research in Jingan District of Shanghai (zs202102).

References

1. Alkhafaji, S., Sriram, B.: Instructor's Performance: A Proposed Model for Online Evaluation. International Journal of Information Engineering and Electronic Business 5(4), 34-40 (2013)
2. Asare, S., Daniel, B.K.: Factors Influencing Response Rates in Online Student Evaluation Systems: A Systematic Review Approach. Journal of Interactive Learning Research 29(2), 133-143 (2018)
3. Castillo, I., Pascual, V.: The WebSocket Protocol as a Transport for the Session Initiation Protocol (SIP). Journal of Biosciences 33(3), 309-311 (2013)
4. Chen, J.: Analysis and Study of an Online Assessment System for the Data Structures Course. Intelligent Computer and Applications 10(06), 264-267 (2020)
5. Cormier, S.M., Zheng, L., Hill, R.A., Nova, R.M., Flaherty, C.M.: A flow-chart for developing water quality criteria from two field- based methods. The Science of the Total Environment 633(15), 1647-1656 (2018)
6. José, L.P.L., Augusto, C.P., Rocío, A.M.: Analysis of the Academic Management and Assessment of External Placements from the University of Valladolid. Procedia Social and Behavioral Sciences 139(2014), 487-495 (2014)
7. Kang, H.Y., Fan, X.Z., Tang, S.P.: Research and Design of Online Test-evaluating System Based on J2EE. Computer Engineering 13, 169-171 (2004)
8. Kurt, H.S., Doan, Z.: Pre-Service Science Teachers' Skills to Express The Algorithms Used in Solving Physics Problems with Flowcharts (An Example From Turkey). Jurnal Pendidikan Fisika Indonesia 16(1), 24-33 (2020)
9. Li, Y., Yang, G.B., Ding X.L., Zhu, Y.P.: Early DIRECT Mode Decision for MVC Using MB Mode Homogeneity and RD Cost Correlation. IEEE Transactions on Broadcasting 62(3), 700-708 (2016)
10. Lorentz, J., Sorana-Daniela, B.: Auto-calibrated Online Evaluation: Database Design and Implementation. Leonardo Electronic Journal of Practices and Technologies 5(9), 201-204 (2006)
11. Mo, Z., Zhang, A., Yang, Z.: A new parallel algorithm for vertex priorities of data flow acyclic digraphs. Journal of Supercomputing 68(1), 49-64 (2014)

12. Moghadam, N., Li, H.: A New Wireless Multicast Queuing Design Using Network Coding and Data-Flow Model. IEEE Communications Letters 20(8), 1603-1606 (2016)
13. Monika., Upadhyaya, S.: Secure Communication Using DNA Cryptography with Secure Socket Layer (SSL) Protocol in Wireless Sensor Networks. Procedia Computer Science 70, 808-813 (2015)
14. Moskal, A.C.M., Stein, S.J., Golding, C.: Can you increase teacher engagement with evaluation simply by improving the evaluation system?. Assessment and Evaluation in Higher Education 41(2), 286-300 (2016)
15. Murugan, P.V., Queen, V.M.: MOOCs as a Digital Learning Platform. International Journal of Multidisciplinary Research Review 3(1), 28-37 (2020)
16. Rienties, Bart.: Understanding academics' resistance towards (online) student evaluation. Assessment and Evaluation in Higher Education 39(8), 987-1001 (2014)
17. Tucker, B., Jones, S., Straker, L.: Online student evaluation improves Course Experience Questionnaire results in a physiotherapy program. Higher Education Research and Development 27(3), 281-296 (2008)
18. Wang, D.M., Ding, L., Li, G.J.: Research on OA System Development Platform Architecture of MVC Mode. Applied Mechanics and Materials 421, 690-693 (2013)
19. Wang, M., Yan, Z., Wang, X.: Design and Implementation of Home Heating Intelligent Management Application on iOS Mobile Platform. Guide of Science and Education 17(10), 1-12 (2015)
20. Wang, S.M.: On-line Examination System Based on Browser/Server Mode. Computer Technology and Development 1, 59-60 (2014)
21. Wang, Z.X.: Summary of the Development of Internet Online Education. Creative Education Studies 3(4), 164-167 (2015)
22. Wojciechowski, J., Sakowicz, B., Dura, K., Napieralski, A.: MVC model, struts framework and file upload issues in web applications based on J2EE platform. Modern Problems of Radio Engineering, IEEE International Conference Telecommunications and Computer Science 342-345, (2004)
23. Wu, H.L., Cheng, Y.H.: Design of the Logistics Management System Based on J2EE and MVC. Advanced Materials Research 765-767, 1419-1422 (2013)
24. Yviquel, H., Boutellier, J., Raulet, M., Casseau, E.: Automated design of networks of Transport-Triggered Architecture processors using Dynamic Dataflow Programs. Signal Processing Image Communication 28(10), 1295-1302 (2012)
25. Zhang, J., Pu, X., Zhang, Z.: Design and Implement of Teaching Resources Management Network Platform Based on MVC. Applied Mechanics and Materials 631-632(2), 999-1002 (2014)
26. Zhang, L.Q., Li, Y.: Design and Implementation of College Program Online Evaluation System Based on B/S. Communication and Information Technology 4, 33-36 (2018)
27. Zhang, W.S., Chen, H.: The Research and Application of Modular Mobile Phone Web Front-end Based on MVCS Mode. International Journal of Future Generation Communication and Networking 8(5), 97-106 (2015)
28. Zhang, Y.F., Ke, C.Y.: Applied Technology in an Interactive Design for a Web-Based Language Teaching System. Advanced Materials Research 886, 621-624 (2014)
29. Zhong, Y.Z., Gui, Q.: The Design and Implementation of the ACM Competition Online Evaluation System. Wireless Internet Technology 17(18), 42-44 (2020)

Peng Jiang is an associate professor at Jingan Branch Campus, Shanghai Open University, China. His current research interests include Educational technology and Management Information System. Contact him at jzhpmail@163.com.

Kexin Yan is the corresponding author of this paper, she is a master at the Management School, Shanghai University of International Business and Economics, China. Her current research interest is the Data Science and Management Information System. Contact her at ykx980219@163.com.

Haijian Chen is a professor at the Shanghai Academic Credit Transfer and Accumulation Bank for Lifelong Education, China. He received his Ph.D. in Management Science and Engineering from Shanghai University of Finance and Economics, China in 2015. His current research interests include Educational technology and cloud computing. Contact him at xochj@sou.edu.cn.

Han Sun is a lecture at management school, Fudan University, China. He received his Ph.D. in Management Science and Engineering from Tongji University, China in 2002. His current research interests include Management Information System and big data analysis. Contact him at sunhai@fudan.edu.cn.

Received: February 01, 2021; Accepted: July 30, 2021.

Computer Science and Information Systems 19(1):205–227 https://doi.org/10.2298/CSIS210104046S

Applied Machine Learning in Recognition of DGA Domain Names

Miroslav Štampar[1] and Krešimir Fertalj[2]

[1] SekuriPy LLC, Mirka Račkog 10,
10360 Zagreb, Croatia
miroslav.stampar@sekuripy.hr
[2] Faculty of Electrical Engineering and Computing, Unska 3,
10000 Zagreb, Croatia
kresimir.fertalj@fer.hr

Abstract. Recognition of domain names generated by domain generation algorithms (DGAs) is the essential part of malware detection by inspection of network traffic. Besides basic heuristics (HE) and limited detection based on blacklists, the most promising course seems to be machine learning (ML). There is a lack of studies that extensively compare different ML models in the field of DGA binary classification, including both conventional and deep learning (DL) representatives. Also, those few that exist are either focused on a small set of models, use a poor set of features in ML models or fail to secure unbiased independence between training and evaluation samples. To overcome these limitations, we engineered a robust feature set, and accordingly trained and evaluated 14 ML, 9 DL, and 2 comparative models on two independent datasets. Results show that if ML features are properly engineered, there is a marginal difference in overall score between top ML and DL representatives. This paper represents the first attempt to neutrally compare the performance of many different models for the recognition of DGA domain names, where the best models perform as well as the top representatives from the literature.

Keywords: domain generation algorithm, binary classification, supervised machine learning, deep learning, blind evaluation.

1. Introduction

When attempting to establish a connection with *command and control* (C&C) server(s), a certain type of malicious programs (malware) create numerous *domain name system* (DNS) queries for domain names generated in a pseudo-random way, from which the majority never was and will never be registered. With the same initial value (i.e. *random seed*), usually associated with the run-time environment (e.g. current time), attackers can blindly share the same fresh list of domain names with infected hosts, without taking any intermediary steps. Thus, in case of a need for exchanging data with infected hosts, attackers have to register only a few domains from the current list and point them to the C&C server's IP address.

The family of algorithms intended for the described algorithmic creation of domain names is called *domain generation algorithms* (DGAs). Such algorithms can

pseudo-randomly generate a large number of *algorithmic* (or simply DGA) domain names (Table 1) to bypass potential security mechanisms used for detection and blocking of malicious network traffic [1]. Depending on the set of elements used in the pseudo-random generation, DGAs further split to *regular* (R) – using characters, and *dictionary* (D) – using a predefined set of words, where the latter represents the less common, but harder to detect class. In further text, DGA will refer to regular class if not explicitly declared.

Table 1. Examples of DGA domain names

DGA	Type	Example domain names
Bobax	R	*qrwxktojqz.yi.org, ttcwzadqxpm.dynserv.com*
Banjori	D	*earnestnessbiophysicalohax.com*
Cryptolocker	R	*bqwqeiswupyny.org, oocevdwyruhdi.co.uk*
Conficker	R	*ntpyocx.info, kfoqmgax.com, eiwzqeaosf.info*
Dyre	R	*aa1442a1beba3793bbde2582b4127b66ae.cc*
Locky	R	*hrgcmmihpxth.in, cbkmotlvy.yt, ecsiequ.pm*
Necurs	R	*vyguwpynyxaxld.in, caxadsjuygrem.ac*
Pushdo	R	*qaqicvofe.com, cumocuwupjo.com, cumocuwu.kz*
TinyBanker	R	*ghefvfkkxtgg.ru, mqsqytogddne.ru, hosgnecdevwt.ru*

Malware writers choose DGA to create resilient botnet infrastructures [2]. Resilience is primarily assured by disrupting the ability to block malware-related C&C communication, such as in the case of using blacklists of known malicious domain names [3]. Namely, if we consider the situation where each malware family uses its variant of DGA, we can conclude that the whole process of collection, distribution, and usage of blacklist(s) for all up-to-date DGA domains quickly becomes impractical. For example, malware from the Conficker family can generate 250 to 50,000 domain names per day, depending on the variant, while only one of these domains has to be registered by attackers to propagate new instructions to infected hosts [4].

Domain name is a sequence of labels split by dots (e.g. *www.example.com*), with *chosen prefix* (e.g. *example*) and *public suffix* (e.g. *.com*, *.co.uk*) as most distinguished parts. As a public suffix – also known as a *top-level domain* (TLD) – can contain more than one label (e.g. *.co.uk*), the term *effective TLD* (eTLD) is the more correct one [2]. Along with the use of regular registration of domain names (e.g. *3b580fa7.com*), there are cases where malware (e.g. Bobax, Corebot, Symmi, etc.) uses DGA to generate domain names with dynamic domain providers (e.g. *zqjotkxwrq.dyndns.org*). Thus, in this research, we will analyze only properties for the chosen prefix, which represents the arbitrarily selected label for the registered domain.

One important case which perfectly illustrates the necessity to inspect only the chosen prefix, while ignoring other labels, is the usage of *DNS Blacklist Lookup* (DNSBL) services. Such services provide a quick way to lookup entries in centralized databases through the usage of DNS protocol, where server response contains the lookup results. Entries can be anything requiring the additional check, such as suspicious domain names, IP addresses, file hashes, e-mail addresses, etc. Generally, as DNS labels have a

restricted set of ASCII characters that can be used, entries are specially encoded (e.g. Base32 format) and prepended to the domain name used for such service (e.g. *ff572stfjvxezcp5ueuzxsttvebqeaqbaeaq.a.e.e5.sk* – for Avast blacklist lookup). As a result, related domain names can appear to be DGA generated.

As the whole point of DGA is the evasion of potential security mechanisms, malware authors should be able to register any of the generated domain names, while in the case of DNSBL services main domain name is always the same. Otherwise, security providers could simply blacklist the main (i.e. common) domain name to deal with the DGA malware. Hence the necessity to analyze only the chosen prefix of the inspected domain name as it represents the part that can be registered within the eTLD or dynamic DNS registrar.

Thus, network security analysts should be able to programmatically detect C&C communication attempts toward DGA domains and neutralize infected hosts inside their organization(s). Related DNS traffic is generally "noisy" and should be relatively easy to detect with manual inspection, as queried domain names generally do not look like they have been generated by a human. Nevertheless, the real challenge is the automatic recognition of DGA domain names, with as much accuracy as possible, which is the main topic of our research.

2. Brief Introduction to Recognition of DGA Domain Names

One of the essential features of malware that uses DGA to communicate with C&C server(s) is the noticeable amount of failed DNS queries (Note: response code *NXDOMAIN – no such name*) [5]. As such behavior appears as an anomaly when compared to regular traffic, network security analyst should be able to fairly easily recognize artificially created DGA domain names (e.g. *mkhjbvxvuqznmcjmy.com*) – at least for regular DGAs – by manually inspecting the DNS log entries. Nevertheless, in this research, we are trying to find the best method to perform the recognition in an automated way.

In a general case, groups of DGA domain names can be discovered based on a list of responses for failed DNS queries generated by an infected host. Domain names in such groups usually share two or more common attributes, such as length, TLD, client IP address, high Shannon-entropy score [6], similar frequency of occurrences for different types of characters (e.g. vowels, consonants, numbers, etc.), and temporal proximity of associated DNS traffic. Such groups with shared attributes property are also referred to as *clusters* [7][8].

Heuristics (HE) are fast decision-making strategies based on limited information, yet frequently correct [8]. A *simplistic* HE method for identifying DGA domain names may include the search for all failed DNS queries generated per client for domain names having at least 8 characters long chosen prefix (e.g. *bsfwptsyobt.com*) and percentage of vowel occurrences of less than 10%. These conditions are based on general observation where DGA domain names should be sufficiently large to cover a significant number of combinations, while the percentage of vowel usage should be distinctively lower than in any spoken human language.

By processing entries in *Alexa Top 1 Million Sites* (ALEXA1M) [9], list of most popular domain names used in similar research, we found that the mean length of regular domain's chosen prefix is 10.35, percentage of vowel usage is 36.96%, while percentage of chosen prefixes that could trigger false-positive (FP) identification in proposed HE method is 0.08%. Additionally, if the condition where the corresponding DNS query has to result with the lookup failure is applied, the probability of FP identification effectively becomes negligible.

Running such HE method for 24 hours inside the *Class B* production network environment resulted in the detection of three infected hosts which generated queries for different clusters of non-existent domains. By running additional checks with help of specialized service *DGArchive* [10], we found that recognized clusters of DGA domain names are specific to the malware families Conficker, Necurs, and Nymaim. Thus, if the final goal is simply the detection of infected network hosts, running the described HE method should be sufficient in a general case.

While the proposed HE method could be used for a quick check of potential DGA malware presence in network traffic, it is not suitable for usage in systems where it is crucial to perform classifications with high *accuracy* (ACC), i.e. *Intrusion Detection System* (IDS) and *Intrusion Prevention System* (IPS). Hence, as found in related work, some form of supervised *machine learning* (ML) is commonly used to detect DGA domains.

In ML, a *feature* is an individual measurable property or characteristic of a phenomenon being observed [11], where the main challenge is finding the right set of features for learning purposes. In such a case, input data is described appropriately, so the underlying algorithm could find an optimal parametric model connecting input feature vectors with the expected results. In the case of DGA recognition, potential features include the length of chosen prefix and the percentage of vowel occurrences, as described in the simplistic HE method.

Deep learning (DL) is a special class of relatively new ML algorithms, based on *artificial neural networks* (ANN), where feature extraction is automatized [12]. This means that compared to conventional ML algorithms, where human engineer – with a considerable amount of engineering skills and domain expertise – has to choose what features (e.g. length, digit ratio, etc.) to include in the model, DL algorithms can automatically select "critical" features during the learning process. Since DL represents a distinguished field of ML, with key differences in methodology and philosophy when compared to conventional ML algorithms, the corresponding class of algorithms will be referred to as DL, while conventional ML algorithms will be referred to with just ML in further text.

Antonakakis et al. [7] used the statistical *Hidden Markov Model* (HMM) for classification, analyzing features: length and entropy of each label within a domain name, hierarchical level, n-gram distribution, etc. Positive class representatives used for learning were based on 59,144 DGA generated domains, collected inside the virtual environment for a relatively small set of malware families: Conficker, Murofet, Bobax, and Sinowal, while negative class representatives were based on the top 10,000 domain names from ALEXA1M. While authors achieved almost perfect results during the regular learning process, with a *true-positive rate* (TPR) of 99.72% and *false-positive rate* (FPR) of 0.1%, during the evaluation on real-life network traffic authors achieved TPR of 91% and FPR of 3%. To conclude on this point, in cases when research is

focused on severely limited datasets, with a narrow list of DGAs, experimental results are far from those achievable in real-life. Thus, in our research, we included representatives for the majority of known DGAs, which resulted in consistent performance between different datasets and better results in the production environment.

Ahluwalia et al. [13] used *Random Forest* (RF) for classification, analyzing features: number of consonants, number of vowels, number of digits, 3-gram distribution, and total length. Positive class representatives used in the training process were based on the analysis of 100,000 DGA domains for the following malware families: Cryptolocker, Zeus, Conficker, Tinba, Ramdo, Matsnu, Rovnx, and GameOver Zeus. Negative class representatives were based on the analysis of the top 100,000 domains from ALEXA1M. As a result, authors achieved results of TPR 98.96 % and FPR 2.1%. One of the conclusions was that the total length of the domain name is a key feature for identifying DGA domains and that FPR drastically rises for cases below 8 characters. During the feature selection process, we came to the same conclusion that length indeed represents one of the most important features.

Wang and Chen [14] used RF, *Support Vector Machine* (SVM), and *Naive Bayes* (NB) for classification, analyzing features: length and entropy of *second-level domain* (SLD), together with appearance probabilities of contained n-grams (3, 4, and 5), based on the probability lists calculated from most commonly used English terms and ALEXA1M domain names. The best results were achieved with RF – TPR 97.53% and FPR 0.20%, similar to our results for the standard dataset. It should be noted that while authors included statistical analysis for 5-grams as a feature too, we found in experiments that it does not add any additional value to the ML model in terms of performance.

Yu et al. [15] used DL for classification, particularly models based on *Long Short-Term Memory* (LSTM) and *Convolutional Neural Network* (CNN) neural networks, along with comparative ML algorithm RF. For regular training purposes, authors used domain names obtained from real-traffic, for both positive and negative class representatives, while using an additional *gold* set, consisting of ALEXA1M and DGA domain names retrieved from DGArchive, for "ground truth" validation. Final results of ground truth validation were: LSTM – TPR 74.05% and FPR 0.54%, CNN – TPR 72.89% and FPR 0.31%, RF – TPR 71.28% and FPR 1.33%. During the evaluation, we showed that the proposed feature set is inferior to ours. The main reason is the English-bias, resulting in worse performance and inconsistent behavior between different datasets.

Tran et al. [16] did research on the multi-class imbalance in DGA classification, where the distribution of dataset samples for different DGA families is not uniform. Trained models based on ML algorithm *Random Undersampling Boosting* (RUB) and DL algorithm *Long Short-Term Memory* (LSTM) had similar performance in binary classification task – F1 0.98, which is in pair with our results got for the standard dataset, while used DL models scored considerably better in a multi-class classification task. Even though it is not part of our research, based on related work, we assume that DL models are superior in the multi-class classification, mainly because of their intrinsic ability to memorize the prolonged lexicographical patterns of observed DGA domains [17].

Yu et al. [18][19] compared simple *Endgame* (single LSTM layer) and complex DL architectures: *Invincea* (parallel CNN layers), CMU (forward LSTM layer + backward

LSTM layer), MIT (stacked CNN layers + single LSTM layer), NYU (stacked CNN layers). One of the conclusions was that there is surprisingly little difference between evaluated DL models in terms of accuracy, prompting a preference for the simpler architectures, as they are faster to train and score, while less prone to overfitting. Additionally, the authors pointed that an interesting direction for future work would be to test the trained DL models more extensively on domain names generated by new and previously unseen malware families. In our research, we performed such evaluation, where we showed that ML models generally perform better than DL models, particularly in case of unseen regular DGAs.

Pereira et al. [20] used graph-based method *WordGraph* for extracting dictionaries used by dictionary DGAs. The proposed method is completely agnostic to the dictionary used by the DGA and should learn it by itself. The main proposition is that once these dictionaries are known, it should become straightforward to construct a domain name classifier based on them. For training purposes, authors used *ground truth data* based on samples collected from ALEXA1M and DGArchive, similar as in [15]. While results look promising, with an almost perfect score in all tested cases, a couple of questions arise. Particularly, authors state that they were able to extract 81 dictionaries in five days of real traffic, with 15 validated through service DGArchive, while claiming that they manually verified the remaining 66 dictionaries and confirmed they were malicious, without providing any details whatsoever. At the end, they classified those new dictionaries as generated by unknown malware.

At the beginning of our research, the main focus was on ML models. Based on expertise acquired in everyday network analysis and literature review, we chose an initial set of features, which we heuristically adapted during experimental runs. By analyzing the results for each attempt, we quickly concluded that feature engineering represents the critical part of ML modeling. Nevertheless, as DL became more popular in recent studies, mainly because of its flexibility and a lack of requirement for explicit feature declaration, we gradually extended the scope of our study. Based on initial findings, it became clear that DL modeling is the step forward in this field.

In this paper, we present the results of our study, where we objectively and extensively compare the performance of different ML and DL models for the DGA binary classification. For such a task, we engineered a robust feature set and created two independent datasets, including samples for the majority of known regular and dictionary DGAs. In the end, we evaluated the models, where the most interesting findings are related to the performance of best ML and DL models on samples representing previously unseen (i.e. untrained) DGAs, and the usability validation on historical one-year DNS logs collected from the production environment.

3. Methodology

To find the best model for recognition of algorithmic domains, binary classifiers based on the following ML algorithms were trained in a supervised manner and finally evaluated: NB, *Multilayer Perceptron* (MLP), *Linear Discriminant Analysis* (LDA), *Quadratic Discriminant Analysis* (QDA), *k-Nearest Neighbors* (KNN), SVM, *Decision Tree* (DT), *Extra Trees* (ET), RF, *Bagging* (BAG), *Gradient Boosting* (GB), *Extreme*

Gradient Boosting (XGB), *Adaptive Boosting* (AB) and RUB; along with simple DL models: *Simple Recurrent Neural Network* (SRNN), *Gated Recurrent Unit* (GRU), CNN and LSTM (*Endgame*), and complex DL models: *Invincea*, C2W, CMU, MIT, NYU.

Implementation was done using the programming language Python and third-party programming libraries *scikit-learn* (*sklearn*) [21], *keras* [22], *xgboost* [23], and *imbalanced-learn* (*imblearn*) [24], where each represents the de facto standard in its field of operation. Parameter values used during the instantiation of ML and simple DL models can be found in Table 2.

Table 2. Parameter values used in ML and simple DL model instantiations

Model	Library	Classifier / Base layer	Parameter values
NB	sklearn	*GaussianNB*	-
MLP	sklearn	*MLPClassifier*	*hidden_layer_sizes=(128,)*
LDA	sklearn	*LinearDiscriminantAnalysis*	*solver='SVD'*
QDA	sklearn	*QuadraticDiscriminantAnalysis*	-
KNN	sklearn	*KNeighborsClassifier*	*n_neighbors=15*
SVM	sklearn	*SVC*	*kernel='linear'*
DT	sklearn	*DecisionTreeClassifier*	*max_depth=None*
ET	sklearn	*ExtraTreesClassifier*	*n_estimators=128*
RF	sklearn	*RandomForestClassifier*	*n_estimators=128*
BAG	sklearn	*BaggingClassifier*	*n_estimators=128*
GB	sklearn	*GradientBoostingClassifier*	*n_estimators=128*
XGB	xgboost	*XGBClassifier*	-
AB	sklearn	*AdaBoostClassifier*	*n_estimators=128*
RUB	imblearn	*RUSBoostClassifier*	*n_estimators=128*
SRNN	keras	*SimpleRNN*	*units=128*
GRU	keras	*GRU*	*units=128*
LSTM	keras	*LSTM*	*units=128*
CNN	keras	*Conv1D*	*filters=128, kernel_size=4*

After extensive initial tests and analysis of obtained results, we decided to use default parameter values where changes did not result in noticeably better results. To avoid the potential issue with the different number of *estimators* used in ensemble [25] type of ML models and *units* or *filters* in DL models, we chose to set respective parameter values of *n_estimators*, *hidden_layer_size*, *units*, and *filters* to 128 where applicable. Even though it seems to be the popular choice in related work (e.g. [15] [16][18][19][26]), we experimentally confirmed the assumption that higher value should not yield with significantly better results in evaluated classifiers.

For such a task, we used sklearn's *GridSearchCV*, a specialized tool for an exhaustive search for best values over the specified list of parameters. It resulted in the (inconsistent) best case accuracy improvements of less than 0.08%, compared to experimental results got with our chosen parameter values, in case of all models except KNN. In that case, the change of initial value for *n_neighbors* from 5 to *GridSearchCV* suggested 15 resulted in the accuracy improvement of 0.16%. Furthermore, it should be

noted that, in the case of MLP, adding more hidden layers did not result in any improvements.

Based on expertise acquired in everyday network analysis and literature review, we chose the following self-explanatory features for ML modeling purposes: length (I), character (Shannon) entropy (II), (decimal) digit ratio (III), length of the longest vowelless sequence (IV), length of the longest common prefix that can be found in at least two ALEXA1M chosen prefixes (V), length of the longest common suffix that can be found in at least two ALEXA1M chosen prefixes (VI), mean positional distance of nearby vowels (VII), mean ASCII distance of adjacent characters (VIII), number of occurrences of numerical sequences (IX), mean frequency indices of 2-grams (X), 3-grams (XI) and 4-grams (XII) given the previously calculated lists of all possible n-grams found within ALEXA1M chosen prefixes sorted by the number of occurrences. It should be noted that the n-grams not appearing in the ALEXA1M have been treated as the last elements of notable lists, thus avoiding additional penalization.

The first chosen prefix (*example*) in the given example (Table 3) represents the regular domain name, the second chosen prefix (*spiderwjzbmsmu7y*) represents the Tor (anonymity network) domain name – included only for comparative purposes, while the last chosen prefix (*wxkjzdbmowq*) represents the DGA domain name. At first look, features based on ALEXA1M n-grams (X, XI, XII) seem to be the most promising for the recognition of DGA domain names, as there appears to be a significant difference in calculated values between different classes of domain names.

Calculated distances and sequence (run) lengths were specifically inspired by *Diehard* [27] and *FIPS PUB 140-2* [28], specialized batteries of statistical tests for testing the quality of *pseudo-random number generators* (PRNG). The main proposition was that as DGA domain names are generated in a pseudo-random way, they should have better statistical results when tested for pseudo-randomness, compared to regular domain names. While mentioned batteries require significantly larger binary sequences, we derived a couple of simplified tests – namely IV, VII, VIII, IX – to perform similar statistical analysis tests on chosen prefixes for domain names.

Table 3. Example of calculated feature values for different chosen prefixes

Feature	*example (.com)*	*spiderwjzbmsmu7y (.onion)*	*wxkjzdbmowq (.info)*
Length (I)	7	16	11
Character (Shannon) entropy (II)	2.52	3.75	3.28
Digit ratio (III)	0	0.06	0
Length of longest vowelless sequence (IV)	3 (*mpl*)	8 (*wjzbmsmu*)	8 (*wxkjzdbm*)
Length of longest (ALEXA1M) common prefix (V)	7 (*example*)	7 (*spiderw*)	2 (*wx*)
Length of longest (ALEXA1M) common suffix (VI)	7 (*example*)	2 (*7y*)	3 (*owq*)
Mean positional distance of nearby vowels (VII)	3	4	5.5
Mean ASCII distance of adjacent characters (VIII)	11.33	16.4	8.2
Number of occurrences of numerical sequences (IX)	0	1	0
Mean frequency index (ALEXA1M) of 2-grams (X)	142	424	570
Mean frequency index (ALEXA1M) of 3-grams (XI)	1,037	13,443	17,644
Mean frequency index (ALEXA1M) of 4-grams (XII)	4,935	147,773	230,265

As part of the data preparation phase, calculated feature vectors were standardized to normally distributed data, with the sklearn's preprocessing utility class *StandardScaler*. The main reason is the requirement of specific ML algorithms, such as KNN and SVM,

which assume that all features are centered on zero and have variance in the same order, while that same transformation does not affect the performance of other ML algorithms.

Because of DL algorithms' ability to recognize and learn patterns in long sequences, such as in text or images, in DGA classification task samples represent raw numerical representation of chosen prefixes for domain names. Hence, instead of using feature vectors as in the case with ML models, in featureless DL models, chosen prefixes are transformed to integer representations of contained characters. If we know that the maximum length of a label in the domain name is 63, each chosen prefix is transformed to a zero-padded vector of length 63, with elements representing character indices inside the lookup table of valid DNS characters. For example, chosen prefix *google* becomes a numerical vector [7, 15, 15, 7, 12, 5, 0, 0, 0 … 0].

Simple DL models – SRNN, GRU, LSTM, and CNN – are based on *Endgame* [29], mostly because of its simplicity, wide acceptance, and generally good performance, where LSTM can be considered as the *Endgame* itself. Pseudo-code for the creation of simple DL models can be found in the continuation, where italicized identifiers represent utilized keras-specific layers (i.e. classes):

```
// Valid DNS label characters
alphabet := "abcdefghijklmnopqrstuvwxyz0123456789-"
// Maximum length of DNS label
max_label_length := 63
// Length of embedding vector
embedding_vector_length := 128
// Dropout threshold value
threshold := 0.5

function CreateDLModel(main_layer_class)
  m := Sequential()
  m.add(Embedding(input_dim := LENGTH(alphabet)+1, output_dim
:= \
    embedding_vector_length, input_length := max_label_length))
  if main_layer_class = Conv1D then
    m.add(main_layer_class(filters := embedding_vector_length,
\
      kernel_size := 4))
    m.add(GlobalMaxPooling1D())
  else
    m.add(main_layer_class(units := embedding_vector_length))
  endif
  m.add(Dropout(threshold))
  m.add(Dense(1))
  m.add(Activation("sigmoid"))
  m.compile(loss := "binary_crossentropy", optimizer := "adam")
  return m
end function


srnn := CreateDLModel(SimpleRNN)
gru := CreateDLModel(GRU)
lstm := CreateDLModel(LSTM)
cnn := CreateDLModel(Conv1D)
```

Complex DL models – *Invincea*, C2W, CMU, MIT, and NYU – were implemented with the minimum modifications compared to the original work. In case that there were no details regarding certain aspects of the model in the original paper, we chose the preferred solution based on our initial findings and other related work. The basic architecture information can be found in Table 4.

Table 4. Architectures used in complex DL models

Model	Architecture	Reference
Invincea	Parallel CNN layers	[30]
C2W	Forward LSTM + backward LSTM layer	[31]
CMU	Forward GRU + backward GRU layer	[32]
MIT	Stacked CNN layers + single LSTM layer	[33]
NYU	Stacked CNN layers	[34]

It should be noted that we decided to use the name C2W for the LSTM-based model, inaccurately referred to as CMU by Yu et al. [19], while reestablishing the name CMU for the GRU-based model. The reasoning is based on the original (CMU) research [32], where Dhingra et al. explicitly state that the proposed GRU-based architecture "*uses a similar structure to the C2W model in* [31]*, with LSTM units replaced with GRU units*". The same inaccuracy can be found in other derived work (e.g. [35]).

While the process of setting up and training ML models in sklearn was straightforward, in the case of DL models we had to fine-tune certain aspects. Most notably, to prevent *overfitting* in DL models, and thus avoid poorer prediction performance on new datasets (e.g. blind dataset), we used *early stopping*. In this method, the *validation loss* – performance measure function used for cross-validation on a fraction of the training data (10% of training samples) – is calculated after each epoch. In case of no improvement, the whole training process is interrupted. Additionally, to avoid the *local extrema entrapment*, a *patience level* of 5 is used, thus training stops only in case of 5 consecutive epochs without training improvement, while the best model is preserved between epochs for future use.

To evaluate ML and DL models in a "blind trial", two independent datasets were used. Namely, along with the *standard* set, which is generally considered sufficient in similar DGA research, an additional (independent) *blind* set was used. While in the majority of related work standard set is the only one used and split in different ways (e.g. holdout, k-fold cross-validation, random subsampling, etc.) for different purposes, in our research we also used blind set created from unrelated sources and by using different filtering methods. Thus, while the standard set was used for regular training and standard evaluation, with 70% of samples for training and 30% for testing purposes, the blind set was used for additional unbiased blind evaluation of trained models.

Standard set is the primary (regular) dataset that was used for training and basic performance evaluation of corresponding models. Set is balanced, with 739,377 positive samples and the same number of negative samples. Positive samples consist of synthetic

chosen prefixes, representing uniformly distributed 51 regular[1] and 4 dictionary[2] DGAs, while negative samples consist of filtered ALEXA1M chosen prefixes.

Basic filtering of ALEXA1M chosen prefixes was done by excluding all entries, where length (i.e. 4 or less) or the character set used (i.e. non-alphanumeric) could in no way be associated with known DGA. Additionally, "problematic" chosen prefixes that could not be classified manually by network security analyst as non-algorithmic were also excluded, such as *132770*(.com) (decimal digits), *6f76b4c82656094f26*(.com) (hexadecimal digits) or *gtplkcbpl*(.com) (consonant-only). This way we reduced the possibility of introducing undesired noise into the standard set that could potentially affect the performance of trained models.

Synthetic chosen prefixes for regular DGAs were artificially generated based on descriptive regular expressions obtained from DGArchive, with a generalization that the distribution of pseudo-randomly chosen characters is uniform within the predefined character set. For example, Bamital DGA described with the regular expression *[0-9a-f]{32}\.(org|info|co\.cc|cz.cc)$*, resulted in the function $BAMITAL_{DGA}$, returning the string of length 32, pseudo-randomly chosen from a pool of hexadecimal digits. In this way, quality, quantity, and variety of positive classification data dramatically increased compared to other related work, although they were not generated by any existing DGA. The main proposition was that this way we could artificially generate any number of samples for positive classification, without a way to easily differentiate when compared to real DGA runs. As an example, chosen prefix *bde15d38ecc65c8 01a6ab50a59cea738* generated by real Bamital DGA does not have any distinct property – such as length, character domain, or character (Shannon) entropy – when compared to synthetic chosen prefix *f5c087c1905b38e110e30d5a2743469e* generated by synthetic function $BAMITAL_{DGA}$. Pseudo-code for the whole process of creation of synthetic chosen prefixes for regular DGAs is as follows:

```
letters := "abcdefghijklmnopqrstuvwxyz"
digits := "0123456789"


function Generate(alphabet, min_length, max_length)
  a := RandomInteger(min := min_length, max := max_length)
  r := RandomString(pool := alphabet, length := a)
  return r
end function

// Generates sample for BAMITAL DGA
function BAMITAL_DGA()
  a := CONCAT(digits, "abcdef")
  r := GENERATE(alphabet := a, min_length := 32, max_length :=
32)
  return r
end function
// ... functions for 49 more DGA algorithms ...
```

[1] Bamital, Bedep, Blackhole, Bobax, Conficker, Corebot, Cryptolocker, DNS Changer, DirCrypt, Dyre, EKforward, Emotet, Feodo, Fobber, Gameover, Gameover P2P, Gspy, Hesper, Locky, MadMax, Modpack, Murofet, Necurs, Nymain, Oderoor, PadCrypt, Proslikefan, Pushdo, Pushdotid, Pykspa, Pykspa 2, Qadars, Qakbot, Ramdo, Ramnit, Ranbyus, Rovnix, Shifu, Simda, Sisron, Sphinx, Sutra, Symmi, Szribi, Tempedreve, TinyBanker, Torpig, Urlzone, Virut, VolatileCedar, XxHex

[2] Banjori, Gozi, Matsnu, Suppobox

```
// Main procedure
procedure Main()
  a := {BAMITAL_DGA, ...}
  b := Input("# of samples to generate:")
  for i in {0..b} do
    c := RandomSample(pool := a, min_length := 1, max_length :=
1)[0]
    d := c()
    Print(d)
  end for
end procedure

Main()
```

In the case of dictionary DGAs, synthetic chosen prefixes were generated based on reverse-engineered algorithms found in public code repositories[3]. To eliminate the problem related to the usage of the same seed words, trait manifested with repetition of identical patterns across all related samples inside the time-constrained blacklists, we used different seeds found in the "wild". Therefore, in the example of Banjori DGA, we uniformly utilized 37 different characteristic seeds[4] in the process of generation, along with different pseudo-randomly chosen dates.

This way, we eliminated the potential bias specific for related research, where the training and evaluation of models are based on arguable dictionary DGA samples extracted from daily DGA blacklists, with evident excessive repetition of elongated patterns (e.g. *nvpnestnessbiophysicalohax*, *nxzmestnessbiophysicalohax*, *eoyoestnessbi ophysicalohax*, etc.). In our opinion, such unreasonable usage of excessive repetitions in datasets gives an unfair advantage to DL models, having the well-known ability to memorize prolonged lexicographical patterns [17] – while those same patterns usually turn out as useless outside the evaluation environment, mostly because of the narrow period of validity.

Blind set is the control dataset, independent of standard, created to provide the support for unbiased blind evaluation of trained ML models. Set contains 170,045 positive samples and the same number of negative samples. Negative samples consist of 170,045 valid (i.e. non-NXDOMAIN) chosen prefixes collected in the *Class B* production network environment during one month. Positive samples consist of chosen prefixes for real algorithmic domains collected from one week (5-11 July 2020) of DGA blacklist source DGArchive, for 81 DGAs – 76 regular and 5 dictionary DGAs, with 43 regular DGAs and 4 dictionary DGAs appearing in the standard dataset too. The discrepancy is preserved principally for dataset independence preservation, along with the opportunity to analyze model behavior in the expected case of the appearance of previously unseen DGA.

[3] https://github.com/baderj/domain_generation_algorithms and https://github.com/andrewaeva/DGA

[4] *abehmsigotg, alitydevonianizuwb, amentalistfanchonut, anarianaqh, ancorml, anerraticallyqozaw, ardenslavetusul, byplaywobb, ellefrictionlessv, enhancedysb, epictom, ererwyatanb, erionirkutskagl, estnessbiophysicalohax, fordlinnetavox, idablyhoosieraw, inaaforementionedagf, inalcentricem, iologistbikerepil, lcationgreedinessb, leasuredehydratorysagp, llaabettingk, machuslazaroqok, men, orcajanunal, orshipecmascriptivylv, partbulkyf, plefrostbitecycz, rasildeafeninguvuc, rgradienton, rsensinaix, sagabardinedazyx, satformalisticirekb, semitismgavenuteq, sikathrinezad, thoodivettewl, vinskycattederifg*

Negative samples consist of valid chosen prefixes for 437 different TLDs: *.com* (54.46%), *.net* (7.35%), *.hr* (5.61%), *.org* (3.97%), *.uk* (2.23%), *.de* (1.83%), *.it* (1.53%), *.ru* (1.44%), *.rs* (1.33%), *.info* (1.15%), etc. The main assumption used during the collection of negative samples from real traffic was that, in the general case, resolution of DGA domain name would either fail (i.e. NXDOMAIN) or result with the sinkholed response, while the probability for a resolution to a valid non-sinkhole IP address can be effectively ignored. For exclusion of sinkholed domain names, a list of 1,330 IP addresses for known sinkholes has been used, gathered from *Maltrail – Malicious traffic detection system* [36], a specialized IDS system for tracking of malware-related network activities. This way we practically reduced the probability for the inclusion of regular DGA domain names down to zero.

It should be noted that as ALEXA1M represents the list of most popular domain names on the Internet, there is an inherent overlap of negative samples between standard and blind datasets, with a percentage of 37.37%. Although the whole process of creation of datasets is kept independent, this is the single point of sample overlap. As the removal of shared entries from the blind dataset would potentially strengthen the regional bias and move the focus of evaluation on less popular domain names, while at the same time take out the realistic aspect of DNS traffic gathered in the production environment, we decided to leave them.

While a concept *blind* set used in our research is similar to the *gold* set used by Yu et al. [15], there are a couple of crucial differences. Gold set – based on ALEXA1M and DGA domain names from DGArchive – was used for ground truth validation, while blind set – based on real-domains gathered from everyday traffic and DGA domain names from the same source – was used for blind evaluation in our research. As in other related work, during research, we found by trial-and-error that ALEXA1M is the best source for negative samples used in the training process. Additionally, we used real DGA domain names for blind evaluation and synthetically generated positive samples for standard evaluation – specifically avoided by Yu et al., because of their concern on limited availability based on the usual approach with malware runs inside the virtual environment. Therefore, instead of establishing the gold truth dataset and "losing" the possibility of training models based on ALEXA1M, while at the same time providing the independence between datasets used in the standard evaluation and blind evaluation, a blind set was created. Furthermore, conducted blind evaluation can be roughly considered as the measurement of classifier performance in the real-network traffic environment, as all blind dataset samples were either gathered in the production environment or from a daily blacklist of current DGA domain names.

4. Evaluation Results

In preparation for the evaluation, after the initial runs, we noticed that in some cases reduced set of features resulted in slightly better overall results – particularly in the case with simpler ML models such as NB and QDA. Hence, to remove the possibility of training potentially weaker ML models, we performed the *feature selection* beforehand. In such a task, the feature set is being reduced, without significant performance

degradation in the recognition system [37], where features contributing the least in the decision process are discarded.

To ease the process, for each trained ML model based on tree-based algorithms (DT, RF, ET, GB, XGB, and AB) it is possible to extract the *feature importance* (FI) list. Such lists, consisting of calculated feature scores with values ranging from 0 (irrelevant) to 1 (single most important feature), can be used to find how each feature contributes to the overall classification process of the corresponding ML model.

Even though we expected that each ML model will score individual features differently by their importance, during our research we found that some features share a similar level of importance throughout all models (where FI is available). Thus, the observed phenomenon became the basis of our feature selection process, particularly in the case with the least relevant features.

Table 5. FI for different ML models

Feature	DT	RF	ET	GB	XGB	AB	$\bar{F}$
Mean ASCII distance of adjacent characters (VIII)	0.01	0.01	0.01	0.00	0.00	0.01	0.01
Number of occurrences of numerical sequences (IX)	0.00	0.00	0.01	0.00	0.00	0.02	0.01
Character (Shannon) entropy (II)	0.01	0.03	0.02	0.00	0.00	0.02	0.01
Mean positional distance of nearby vowels (VII)	0.01	0.03	0.02	0.00	0.00	0.02	0.01
Length of longest vowelless sequence (IV)	0.01	0.02	0.01	0.00	0.00	0.06	0.02
Mean frequency index (ALEXA1M) of 2-grams (X)	0.01	0.09	0.05	0.00	0.00	0.02	0.03
Digit ratio (III)	0.01	0.01	0.01	0.01	0.03	0.04	0.02
Length (I)	0.06	0.04	0.05	0.06	0.07	0.18	0.08
Length of longest (ALEXA1M) suffix (VI)	0.03	0.09	0.15	0.03	0.06	0.09	0.08
Mean frequency index (ALEXA1M) of 3-grams (XI)	0.02	0.19	0.12	0.01	0.01	0.12	0.08
Length of longest (ALEXA1M) prefix (V)	0.03	0.15	0.12	0.04	0.07	0.12	0.09
Mean frequency index (ALEXA1M) of 4-grams (XII)	0.80	0.34	0.43	0.85	0.76	0.30	0.58

Based on obtained FI (Table 5), we concluded that the mean frequency index (ALEXA1M) of 4-grams makes the most important feature across ML models. More importantly, all features based on lexical ALEXA1M properties generally have greater importance than other features. The relevance of the 4-gram feature (XII) stands out so much compared to others that our immediate impression was that it could be solely used as a HE method for recognition of DGA domains.

Therefore, performing the training process for the “shallow” (i.e. depth set to 1) DT model, primarily chosen due to the intuitive IF-THEN-ELSE resulting structure representing the trained model, we came to the value of 90,674 above which the mean frequency index (ALEXA1M) of 4-grams would have to be valued to classify the tested chosen prefix as DGA. Finally, for comparison purposes with other evaluated models, we created a simple HE method ALEXA4G based only on that single check (Table 6).

Furthermore, as a result of FI analysis, we came to an auxiliary hypothesis that we could use a HE approach in feature selection of evaluated ML models by simply removing features that were at least in one case marked as absolutely irrelevant (i.e.

value 0.00 – highlighted with dashed border in Table 5) – similar to a voting system with right of veto. Thus, we discarded the upper half (VIII, IX, II, VII, IV, X) of features listed in Table 5, while leaving the lower half (III, I, VI, XI, V, XII). To verify the hypothesis, we conducted the training and evaluation of all ML models and compared the performance in both full and reduced sets of features. As a result, in the case of reduced feature set, we got an overall 0.1% improvement of classification performance (Note: based on $\overline{F1}$ score) in all ML models, while the training time has been reduced on average by 73% compared to the original time. Hence, we continued the evaluation with a reduced set of features.

Finally, after the successful evaluation of ML and DL models, the following values were calculated: TPR_A, FPR_A, ACC_A, $F1_A$, TPR_B, FPR_B, ACC_B, $F1_B$, and $\overline{F1}$ (Table 6). The first eight values represent the performance of the corresponding binary classifier, depending on the used dataset (A – standard, B – blind), while the last $\overline{F1}$ represents the mean value based on $F1_A$ and $F1_B$. Basic measures TPR and FPR were chosen as they represent the most basic metrics used in related work. In general, TPR has to be as high, while FPR has to be as low as possible for a model to be acceptable, as otherwise, the detection mechanism could become unusable because of too many positive-misses or too many false-alarms.

Out of all statistical performance measures available for finding the "best" model, ACC and F1 represent the two most commonly used for binary classification in related work. ACC is the measure of all the correctly identified cases and is preferred when the detection of positive and negative classes is of equal importance. For example, in the case of a passive system like IDS, detection of a DGA domain could be considered of the same importance as the detection of a non-DGA domain, as the network security engineer analyzing its report would need to invest additional time "triaging" the false detections of any kind. In comparison, in the case of an active system like IPS, false detection of regular domains would most probably be detrimental to the network-user experience, while missing true detection of DGA-domains up to a certain threshold could be considered as acceptable.

Table 6. Evaluation results for standard (A) and blind (B) datasets

Model	Type	TPR_A	FPR_A	ACC_A	$F1_A$	TPR_B	FPR_B	ACC_B	$F1_B$	$\overline{F1}$
ALEXA4G	HE	0.9216	0.0467	0.9375	0.9364	0.8916	0.0561	0.9178	0.9156	0.9260
NB	ML	0.9295	0.0508	0.9394	0.9387	0.9085	0.0538	0.9274	0.9260	0.9323
RFYu	ML	0.9136	0.0656	0.9240	0.9232	0.9502	0.0530	0.9486	0.9487	0.9359
SRNN	DL	0.9485	0.0303	0.9591	0.9587	0.8855	0.0381	0.9237	0.9206	0.9397
QDA	ML	0.9523	0.0531	0.9496	0.9497	0.9380	0.0669	0.9356	0.9357	0.9427
CNN	DL	0.9477	0.0292	0.9592	0.9588	0.9011	0.0393	0.9309	0.9288	0.9438
C2W	DL	0.9585	0.0225	0.9680	0.9677	0.8835	0.0312	0.9262	0.9229	0.9453
LDA	ML	0.9345	0.0226	0.9559	0.9549	0.9159	0.0404	0.9377	0.9363	0.9456
DT	ML	0.9616	0.0402	0.9607	0.9607	0.9331	0.0698	0.9316	0.9317	0.9462
CMU	DL	0.9604	0.0249	0.9678	0.9675	0.8890	0.0331	0.9279	0.9250	0.9463
LSTM	DL	0.9618	0.0203	0.9707	0.9705	0.8900	0.0297	0.9302	0.9272	0.9488
GRU	DL	0.9639	0.0244	0.9698	0.9696	0.8980	0.0333	0.9324	0.9300	0.9498
MIT	DL	0.9665	0.0236	0.9714	0.9713	0.8948	0.0326	0.9311	0.9285	0.9499
Invincea	DL	0.9571	0.0232	0.9670	0.9667	0.9039	0.0307	0.9366	0.9345	0.9506
NYU	DL	0.9667	0.0254	0.9707	0.9706	0.9015	0.0333	0.9341	0.9319	0.9512
SVM	ML	0.9604	0.0276	0.9664	0.9662	0.9393	0.0566	0.9413	0.9412	0.9537
AB	ML	0.9648	0.0239	0.9705	0.9703	0.9419	0.0570	0.9425	0.9424	0.9564
RUB	ML	0.9636	0.0207	0.9714	0.9712	0.9349	0.0510	0.9419	0.9415	0.9564
ET	ML	0.9659	0.0220	0.9719	0.9717	0.9377	0.0519	0.9429	0.9426	0.9572
BAG	ML	0.9664	0.0199	0.9733	0.9730	0.9369	0.0505	0.9432	0.9428	0.9579
GB	ML	0.9656	0.0192	0.9732	0.9730	0.9380	0.0484	0.9448	0.9444	0.9587
RF	ML	0.9668	0.0192	0.9738	0.9736	0.9379	0.0496	0.9441	0.9438	0.9587
XGB	ML	0.9645	0.0177	0.9734	0.9731	0.9365	0.0457	0.9454	0.9449	0.9590
KNN	ML	0.9670	0.0188	0.9741	0.9739	0.9397	0.0488	0.9454	0.9451	0.9595
MLP	ML	0.9711	0.0224	0.9743	0.9742	0.9486	0.0567	0.9459	0.9461	0.9602

Even though ACC represents one of the most intuitive and obvious measures, the inclusion of F1-score has become the de facto standard in recent related work. One of the main reasons is the ongoing criticism, with claims that ACC solely cannot be considered as a reliable measure anymore, because it provides an over-optimistic estimation of the classifier ability on the majority class [38]. Nevertheless, as ACC and F1 represent the performance of the model distinctly, while at the same time making results comparable to other research, we included both as part of evaluation results and chose the mean value $\overline{F1}$ as the final score for each model.

Along with models described in the methodology part, the following comparative models were also included: ALEXA4G – representing the HE method based solely on the ALEXA1M 4-gram feature (XII) and RFYu – representing the ML model from related work (Yu et al. [15]), with English-biased set of features.

From the evaluation results, it is clear that in most cases there is a consistency in performance between datasets, where better models generally scored better in both

datasets. Thus, the blind evaluation could be considered as the verification of standard evaluation results, where comparative and simpler (probabilistic) ML models scored the worst, ensemble and more "powerful" ML models scored the best, while DL models were in the middle.

Only evident inconsistency in performance can be found in the case of the comparative RFYu. While in the case of the blind dataset it scored almost the same as correlative model RF, its performance has been remarkably poor in the case of the standard dataset, where even the comparative HE method ALEXA4G based on a single feature had better results. With close inspection of false recognitions, particularly FPs, we found that RFYu has a problem with regular domain names containing Internet characteristic non-English n-grams, such as *xpose360*, *mp3koka*, *newxxxvideos*, *1080ip*, *c365play*, *win7dwnld*, etc., same n-grams that could be better learned by the model if only the underlying features *nl2* and *nl3* [15][39] were based on real-domain names instead of the English language.

Better scoring of model RFYu in the case of the blind dataset could be explained by the considerably smaller size of the dataset compared to the standard dataset, where negative samples, consisting of regular domain names collected in real-network traffic, have a greater ratio of the most popular English-language oriented domains. To verify this, we calculated the mean of means for frequency indices of English 3-grams for negative samples, and got a value of 1,788.04 in the case of the standard dataset and a lower value of 1,602.55 in the case of the blind dataset, proving that blind dataset is indeed more English-oriented.

Thus, the results of Yu et al. [15] – particularly performance comparison of DL models and RFYu against the truth-marked *gold set* – and the conclusion of DL superiority should be re-evaluated with a better ML feature set. Used comparative ML model RFYu is English-biased with inconsistent performance between different datasets compared to other models (Table 6), and most importantly underperforming in the case of ALEXA1M – the same set of domains used as a source for negative samples in the gold set. Hence, this can be considered as a prime example of an assertion that feature engineering represents the critical part of ML modeling.

For a detailed comparison between ML and DL classes, we chose the best performing models – MLP (ML) and NYU (DL) – and further analyzed results for DGAs that had a TPR_B less than 0.90 in at least one of those models (Table 7). In the case of dictionary DGA Suppobox, NYU had a TPR_B of 0.46, while MLP underperformed with a TPR_B of just 0.01. As a result of sample analysis, we found in both standard and blind datasets multiple usages of Suppobox characteristic suffixes such as *sherburne*, *underhill*, *electricity*, and *blackwood*, from where we concluded that DL models are superior to ML models in similar cases where the same characteristic prolonged lexicographical patterns can be found in distinct datasets.

Nevertheless, in the case of dictionary DGAs Gozi, Matsnu, and Nymaim2 both models performed almost the same. While Nymaim2 represents the case of non-trained dictionary DGA, where both models perform badly as expected, Matsnu represents the case of trained dictionary DGA where even the DL model failed. By comparison of Matsnu samples in both datasets and the DGA algorithm internals, we concluded that this DGA is particularly problematic for both training and recognition because of lack of repeating patterns, mainly due to the extensive number of combinations generated from

large internal wordlists and the way they are combined (*nouns* and *verbs*) until the predefined chosen prefix length.

In the case of regular DGAs, the MLP model generally scored better than NYU, with the substantial difference in performance for untrained DGAs Darkshell, Qhost, and Qsnatch. By inspecting related samples in the blind dataset, we concluded that NYU most probably failed to recognize those because of their shortness and consequently the lack of information to classify those as positives. In the case of Darkshell, samples had a length of 6 (e.g. *r038zy*), in the case of Qsnatch samples on an average had a length of 5 (e.g. *4xxgz*), while in the case of Qhost samples shared the same prefix *ptmr*, with numeric suffix having a mean length of 4. Nevertheless, ML representative MLP could recognize those due to lack of usage of most common n-grams.

Table 7. TPR_B for "problematic" DGAs

DGA	Type	Trained	MLP (ML)	NYU (DL)
Conficker	R	⊤	0.91	0.89
Darkshell	R	⊥	1.00	0.13
Diamondfox	R	⊥	0.91	0.82
Gozi	D	⊤	0.41	0.41
Matsnu	D	⊤	0.05	0.06
Nymaim2	D	⊥	0.04	0.04
Pitou	R	⊥	0.56	0.56
Pushdo	R	⊤	0.58	0.59
Pykspa2	R	⊤	0.83	0.88
Qhost	R	⊥	1.00	0.29
Qsnatch	R	⊥	0.91	0.34
Simda	R	⊤	0.43	0.53
Suppobox	D	⊤	0.01	0.46
Symmi	R	⊤	0.67	0.67
Szribi	R	⊥	0.85	1.00
UD3	R	⊥	1.00	0.75
UD4	R	⊥	0.93	0.79
Vawtrak	R	⊤	0.66	0.64
Virut	R	⊤	0.76	0.75
Volatilecedar	R	⊤	0.55	0.47

By comparing overall performance concerning the complexity of models, the simplest ML models (NB, QDA, LDA, and DT) scored worse than more complex ML models, while in the case of DL there was no clear distinction. Even though complex NYU, *Invincea*, and MIT scored the best among DL models, simple GRU and LSTM scored better than complex CMU and C2W. Thus, we can confirm the results from Yu et al.

[18][19] and agree with the remark that simpler DL architectures should be preferred over complex DL architectures.

Although ML models generally scored better than DL models, the difference in the final score is marginal. The only significant difference can be found in the case of blind evaluation, where the best ML model MLP performed better in cases with untrained regular DGAs, while the best DL model NYU performed better in cases with trained dictionary DGAs where identical prolonged lexicographical patterns could be found in both datasets. Thus, when considering its flexibility, powerful ability to automatically memorize lexicographical patterns, and the fact that it does not require an extra step of careful feature engineering, a critical process that creates the competitive difference between comparative RFYu and correlative RF model, we came to the conclusion that DL can be considered as the preferred choice over ML for the recognition of DGA domain names.

Even though it was not formally included in the evaluation results (Table 6), we also evaluated the comparative simplistic HE method described in the introductory Section **Error! Reference source not found.**, based only on the chosen prefix length and vowel ratio. While at first glance it scored poorly with TPR_A 0.1708, TPR_B 0.2053, and $\overline{F1}$ 0.3160, it had remarkably low FPR_A 0.0006 and FRP_B 0.0012. If we assume that DGA malware generates at least 5 DGA DNS queries per day, we can conclude that it could be used to easily detect the infected client on the same day of infection, with an exceptionally low probability for FP detection.

As part of usability validation, we chose the best ML model MLP, the best DL model NYU and the simplistic HE method, and ran those against the historical DNS logs for 850 million queries collected inside the production environment for one year (Note: 1st July 2019 to 30th June 2020). As a result, we detected 2 clusters for the following DGAs: Conficker and Dromedan, where verification and DGA type recognition were based on query service provided by DGArchive. Cluster Conficker was active for 277 days, with 9 infected clients, while cluster Dromedan was active for 189 days, with 6 infected clients. All clusters have been detected by both models and a HE method, with a daily average TPR of 0.95 for both models and a daily average TPR of 0.31 for the HE method.

Consequently, during the execution, we realized that the usability of ML and DL models is surprisingly low, at least if used against the DNS queries. No matter the generally good evaluation results (Table 6), even FPR as low as 0.01 (i.e. 1%) effectively results in useless reports. For example, if the daily expected number of queries – as in our case – is 2 million, with 20,000 unique chosen prefixes, FPR of 0.01 results with 200 chosen prefixes being misclassified as DGA – or tens of thousands of wrongfully blocked DNS queries for non-DGA domains if used in an active system like IPS. Thus, the best candidate that could be used against the DNS queries, without additional fine-tuning of relevant thresholds, was the simplistic HE method based just on chosen prefix length and vowel ratio. Its exceptionally low FPR emerges it from other models in the real-life traffic environment. In our case, only 13% of all positive detections were FPs, which means that on the daily average positive detections of 62 DGA domains, only 8 domains were not related to a known DGA. As a result of further analysis, we found that the FPs were in majority of cases related to Ad-serving related networks, where operators deliberately use DGA-alike domains (e.g. *bmkz57b79pxk.com*, *01mspmd5yalky8.com*, *wk4x5rdtoz2tn0.com*) to make them more

resilient to potential blocking from clients. A common feature that differentiates them in DNS traffic from DGA domains is that those DGA-alike domains in general case resolve to valid IP addresses, which highlights the need for inspection of failed DNS responses (i.e. NXDOMAIN) in search of DGA traffic, at least in passive systems like IDS.

5. Conclusions

In this paper, we presented the results of standard and blind evaluations for 14 ML and 9 DL models, along with 2 comparative models, for the recognition of DGA domain names. For such a task, along with the standard dataset used for training and standard evaluation, we used an additional independent "blind" dataset for blind evaluation. By choosing blacklisted DGA domain names and regular domain names collected from production network traffic in case of the blind dataset, in comparison to synthetic creation of positive samples and ALEXA1M list of most popular domain names in case of the standard dataset, we successfully ensured the independence.

Based on calculated FI from different tree-based ML models, we performed a veto-based feature selection process and concluded that the mean frequency index (ALEXA1M) of 4-grams makes the most important feature across trained ML models. To verify this finding, we created a comparative HE method ALEXA4G based on this single feature, which during the evaluation scored only 3.5% worse than the best performing model MLP. Furthermore, we concluded that features based on lexical ALEXA1M properties (i.e. n-grams, longest common prefix, and longest common suffix) are overall more important than all other used features, meaning that the domain-based features should be the focus of related ML modeling.

As a result of the evaluation, we found that ML models generally score better than DL models, although the difference in overall score is marginal. Concerning the complexity of models, the simplest ML models (NB, QDA, LDA, and DT) scored worse than more complex ML models, while in the case of DL there was no clear distinction. Nevertheless, a substantial difference between ML and DL classes could be found with untrained regular DGAs Darkshell, Qhost, and Qsnatch having short chosen prefixes, where best ML model MLP performed considerably better than best DL model NYU, and in the case with trained dictionary DGA Suppobox, where DL model scored better because identical characteristic prolonged lexicographical patterns could be found in both standard and blind datasets.

Thus, when considering its flexibility, powerful ability to automatically memorize lexicographical patterns, and the fact that it does not require an extra step of careful feature engineering, a critical process that creates the difference between good and bad same-architecture ML models, we concluded that DL can be considered as the preferred choice over ML for the general recognition of DGA domain names. Nevertheless, in the case of the creation and usage of specialized models, ML should be the preferred choice for the recognition of regular DGAs, while DL should be the preferred choice for the recognition of dictionary DGAs.

Usability validation for best performing ML model MLP and DL model NYU was done on historical one-year DNS query logs, along with the comparative simplistic HE method based on chosen prefix length and vowel ratio. As a result, we detected 2

clusters for the following DGAs: Conficker and Dromedan, with both models and a HE method. Surprisingly, we found that the best candidate for usage against the DNS queries, without additional fine-tuning of relevant thresholds, was the simplest – simplistic HE method. Its exceptionally low FPR emerges it from other models in the real-life traffic environment, resulting in a practically acceptable number of positives from the perspective of network security analysts. To improve the usability of ML and DL models, the focus of DNS traffic analysis should be moved from queries to failed responses, inadvertently losing the possibility to use such models in an IPS.

Ideas for future work include further research related to specialized recognition of dictionary DGAs and usage of *hybrid* ML and DL models, where benefits should be taken from and drawbacks alleviated of both, to increase the prediction accuracy and decrease the computational complexity.

References

1. Y. Zhou, Q. Li, Q. Miao, and K. Yim, "DGA-Based Botnet Detection Using DNS Traffic.," J. Internet Serv. Inf. Secur., vol. 3, no. 3/4, pp. 116–123, 2013.
2. S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 2014, pp. 192–211.
3. M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in International Workshop on Recent Advances in Intrusion Detection, 2014, pp. 1–21.
4. M. Thomas and A. Mohaisen, "Kindred domains: detecting and clustering botnet domains using DNS traffic," in Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 707–712.
5. T. Wang, X. Hu, J. Jang, S. Ji, M. Stoecklin, and T. Taylor, "BotMeter: Charting DGA-botnet landscapes in large networks," in IEEE 36th International Conference on Distributed Computing Systems (ICDCS), 2016, pp. 334–343.
6. S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "FANCI: Feature-based automated nxdomain classification and intelligence," in 27th USENIX Security Symposium, 2018, pp. 1165–1181.
7. M. Antonakakis et al., "From throw-away traffic to bots: detecting the rise of DGA-based malware," in Proceedings of 21st USENIX Security Symposium, 2012, pp. 491–506.
8. C. Dietrich, "Decision making: Factors that influence decision making, heuristics used, and decision outcomes," Inq. J., vol. 2, no. 02, 2010.
9. "Alexa Top 1 Million Sites," Alexa Internet, Inc. [Online]. Available: http://s3.amazonaws.com/alexa-static/top-1m.csv.zip. [Accessed: 15-Mar-2021]
10. D. Plohmann, "DGArchive," Fraunhofer FKIE. [Online]. Available: https://dgarchive.caad.fkie.fraunhofer.de/. [Accessed: 15-Mar-2021]
11. C. M. Bishop, Pattern recognition and machine learning. springer, 2006.
12. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
13. A. Ahluwalia, I. Traore, K. Ganame, and N. Agarwal, "Detecting broad length algorithmically generated domains," in International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, 2017, pp. 19–34.
14. T. Wang and L.-C. Chen, "Detecting Algorithmically Generated Domains Using Data Visualization and N-Grams Methods," in Proceedings of Student-Faculty Research Day, CSIS, Pace University, 2017, pp. D4–1.

15. B. Yu, D. L. Gray, J. Pan, M. De Cock, and A. C. A. Nascimento, “Inline DGA detection with deep networks,” in IEEE International Conference on Data Mining Workshops, ICDMW, 2017, vol. 2017-Novem, pp. 683–692, doi: 10.1109/ICDMW.2017.96.
16. D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, “A LSTM based framework for handling multiclass imbalance in DGA botnet detection,” Neurocomputing, vol. 275, pp. 2401–2413, 2018.
17. L. Sidi, A. Nadler, and A. Shabtai, “MaskDGA: A black-box evasion technique against DGA classifiers and adversarial defenses,” arXiv Prepr. arXiv1902.08909, 2019.
18. B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, “Character level based detection of DGA domain names,” in 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8.
19. B. Yu et al., “Weakly supervised deep learning for the detection of domain generation algorithms,” IEEE Access, vol. 7, pp. 51542–51556, 2019.
20. M. Pereira, S. Coleman, B. Yu, M. DeCock, and A. Nascimento, “Dictionary extraction and detection of algorithmically generated domain names in passive DNS traffic,” in International Symposium on Research in Attacks, Intrusions, and Defenses, 2018, pp. 295–314.
21. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, no. Oct, pp. 2825–2830, 2011.
22. F. Chollet, “Keras - Deep Learning for humans,” 2015. [Online]. Available: https://github.com/keras-team/keras. [Accessed: 15-Mar-2021]
23. T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
24. G. Lemaitre, F. Nogueira, and C. K. Aridas, “Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 559–563, 2017.
25. T. G. Dietterich, “Ensemble methods in machine learning,” in International workshop on multiple classifier systems, 2000, pp. 1–15.
26. R. Sivaguru, C. Choudhary, B. Yu, V. Tymchenko, A. Nascimento, and M. De Cock, “An evaluation of DGA classifiers,” in 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 5058–5067.
27. G. Marsaglia, “Diehard: battery of tests for random number generators,” CD-ROM, Department of Statistics and Supercomputer Computations Research Institute, Florida State University. 1995.
28. R. Brown and J. Burrows, “FIPS PUB 140-2 Security Requirements For Cryptographic Modules,” 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf. [Accessed: 15-Mar-2021]
29. J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, “Predicting domain generation algorithms with long short-term memory networks,” arXiv Prepr. arXiv1611.00791, 2016.
30. J. Saxe and K. Berlin, “eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys,” arXiv Prepr. arXiv1702.08568, 2017.
31. W. Ling et al., “Finding function in form: Compositional character models for open vocabulary word representation,” arXiv Prepr. arXiv1508.02096, 2015.
32. B. Dhingra, Z. Zhou, D. Fitzpatrick, M. Muehl, and W. W. Cohen, “Tweet2vec: Character-based distributed representations for social media,” arXiv Prepr. arXiv1605.03481, 2016.
33. S. Vosoughi, P. Vijayaraghavan, and D. Roy, “Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder,” in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, pp. 1041–1044.

34. X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in neural information processing systems, 2015, pp. 649–657.
35. C. Choudhary, R. Sivaguru, M. Pereira, B. Yu, A. C. Nascimento, and M. De Cock, "Algorithmically generated domain detection and malware family classification," in International Symposium on Security in Computing and Communication, 2018, pp. 640–655.
36. M. Stampar and M. Kasimov, "Maltrail - Malicious traffic detection system." 2014 [Online]. Available: https://github.com/stamparm/maltrail. [Accessed: 15-Mar-2021]
37. P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," Pattern Recognit. Lett., vol. 15, no. 11, pp. 1119–1125, 1994.
38. D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, p. 6, 2020.
39. B. Yu, L. Smith, M. Threefoot, and F. G. Olumofin, "Behavior Analysis based DNS Tunneling Detection and Classification with Big Data Technologies.," in IoTBD, 2016, pp. 284–290.

Miroslav Štampar is an information security specialist at the SekuriPy LLC, Zagreb, Croatia. His major research interests are network security, malware analysis, machine learning, and vulnerability assessment in information systems. He is author of world-renowned free and open source applications. He has given talks in a number of international expert conventions on information security and programming.

Krešimir Fertalj is a full professor at the Faculty of Electrical Engineering and Computing (FER) at the University of Zagreb, Croatia, where he lectures a couple of computing courses on undergraduate, graduate, specialist and doctoral studies. His professional and scientific interest is in automated software engineering, complex information systems, project management and in software security. He led several scientific and research projects and a few dozen of development projects. He was a mentor to students for over 250 bachelor and graduate theses, nine MSc and eleven PhD theses. He has published near 200 scientific and technical papers. He is the founder of the Laboratory for Special Purpose Information Systems and of Postgraduate Specialist Study "Project Management" at FER. He is a senior member of IEEE and a full member of Croatian Academy of Engineering (HATZ). He served as a Head of Department at FER, a Secretary of Department of Information Systems of HATZ and was one of the founders and member of management board of PMI chapter in Croatia.

Received: January 12, 2021; Accepted: July 08, 2021.

Computer Science and Information Systems 19(1):229–250 https://doi.org/10.2298/CSIS210207050M

Semantic Web Based Platform for the Harmonization of Teacher Education Curricula

Milinko Mandić

Faculty of Education, Podgorička 6,
25000 Sombor, Serbia
milinko.mandic@pef.uns.ac.rs

Abstract. This paper describes a developed semi-automatic software platform for the harmonization of the informatics curricula at all levels of education. The applied algorithms for matching ontologies are described in detail, as well as the principle of mapping informatics curricula to ontological models. The model of the selected informatics teacher education curriculum from the Republic of Serbia was created and compared to the model of the reference informatics teacher education curriculum using a software platform. The analysis of the results includes a comparison with the data obtained for other possible pairs of the created input ontological models (the secondary school ACM K12 model and the reference model, the secondary school model and the model of the selected curriculum). The research presented in this paper indicates that it is necessary to consider the improvement of teacher education curriculum as well as the application of new matching techniques.

Keywords: ontology, alignment, matching, teacher education curriculum, informatics.

1. Introduction

The rapid changes in the field of computer science (CS) require the constant improvement of the CS curricula. Primary, secondary and higher education CS curricula must be mutually aligned, but also have to be harmonized with the development of CS field. Therefore, it is necessary to present the curriculum in such a form that is computer interpretable and easy to change. Also, it is important to facilitate the determination of the curricula harmonization of different levels of education. This could be achieved by applying a software platform that would point out the missing aspects in the curriculum, provide statistical data, the possibility of improving the curriculum model and the like.

The rest of the paper is organized as follows. Section 2 shows related work. Section 3 presents the ontological models of the reference and chosen informatics teacher education curricula. Section 4 describes the architecture of our platform for curricula harmonization and ontology matching methods, applied in the platform. Section 5 gives discussion of the results following the application of the presented software to created ontological models. Section 6 contains concluding considerations, limits of the developed platform and future research aims.

2. Related work

Seitz [1] states that researchers often examined the alignments between the intended and the assessed curricula and between the enacted and the assessed curricula. Bay et al. [2] present the research with the aim to establish the factors that affect curriculum alignment. In the paper, the curriculum alignment is defined as "the compatibility between a country's centralized curriculum determined by the ministry of education and what teachers do during the teaching process". Digital curriculum mapping tool, presented in [3], was created mainly to facilitate „processes of improving curriculum alignment and visibility of learning trajectories for teachers and students". In this paper, the curriculum alignment is interpreted through the „dynamics" between the program structure and the student's learning. The tool provides students, teachers and curriculum evaluators with a quick and easy insight into how and when the acquisition of certain curriculum skills and knowledge is planned. Moreover, the purpose of the created curriculum mapping tool is to enable teachers to better understand the curriculum and the position of the course they teach within a predefined learning trajectory. The developed digital mapping tool is especially important in the accreditation process, as it enables easy access to the content related to the visualized learning trajectory, providing information about thematic areas which the educational institutions consider important. In [4] the author analyzes the possibilities of electronic curriculum mapping system e-CMS for organizing curriculum alignment initiatives. The proposed system can be applied to both internal and external alignments. The internal alignment refers to determining the compliance of the three elements of a course, teaching and learning activities, assessments, and objectives. The external alignment is used to check the consistency of the courses with one another. A research, shown in [5], presents constructive alignment with a cross-institutional study from two Australian universities. The paper emphasizes the importance of two approaches: the top-down institutional alignment implementation at one university and the bottom–up approach within the other. The top-down approach starts with a corporate strategy (higher education institution) and follows a series of sequential steps ending with individual student assessment learning items. The bottom–up approach implies the reverse direction. It can be seen from the cited literature that all analyzed papers and presented tools have in common the investigation of the curriculum alignment by determining the extent to which the teaching is synchronized with the predefined curriculum. They do not deal with the harmonization of curricula with external recommendations or other curricula.

A number of papers in current literature [6-9] testifies to the suitability of using ontologies for curriculum representation. Ontologies ensure the presentation of the curricula so that they can be interpreted by machines. The authors in [8] state that the ontological representation of curricula provides easier curricula alignment. In [9], Electrical Engineering Curriculum was represented through ontologies. The preliminary research, by using the model of a semi-automated Academic Tutor, indicated that the formal representation of the curriculum's knowledge could be shared and reused in the field of education and engineering. The authors in [10] developed a new classroom teaching model driven by the curriculum ontology. It was used in the creation of a teaching plan and verified in the course of E -Commerce. Ref. [11] states that some of the benefits of using ontologies are: sharing common understanding of the information structure, „facilitating reuse of domain knowledge", analyzing domain knowledge. A model of a semi-automatic build of the intelligent curricula based on the existing

educational resources in digital format (such as digital books, web/based tutorials or curricula), was proposed in [12].

Contemporary literature presents many ontology alignment systems that use numerous methods and techniques for ontology matching. Some of them are shown in [13-15].

From the literature review, it can be concluded that there are a number of papers dealing with curriculum alignment, curriculum ontological representation and ontology alignment. However, to our knowledge, only curricula synchronization platform, presented in [16][17], is made on the basis of ontology alignment methods. The platform for the harmonization of teacher education curriculum software, presented in this paper, uses a different input ontological model compared to those presented in [16][17]. Also, the ontology matching process uses (new) algorithms adapted to the specificity of teacher education curriculum comparisons. This is especially true of terminological and taxonomic structural algorithms.

3. Teachers' Curricula Ontological Models

The motivation for the development of the platform shown in this paper is to establish whether the graduates of informatics teacher preparation programs are competent to implement teaching in primary and secondary schools in accordance with contemporary international standards and recommendations. Therefore, the main upper class of the teachers' ontological models is the *Competence* class. Different definitions of competence are summarized in [18]. From [18] it can be seen that competence almost always implies acquiring knowledge and skills. Thus, direct subclasses of the *Competence* class are *Knowledge* and *Skills* classes that are mutually connected via *hasKnowledge/hasSkills* object properties. According to the relevant literature [19-20] (revised) Bloom's taxonomy is particularly suitable for use in computer science field. Therefore, the *Skills* class is modelled based on cognitive domain of the Revised Bloom's taxonomy, i.e., the following classes: *Evaluate*, *Create*, *Analyze*, *Apply*, *Remember-understand* are subclasses of the *Skills* class.

Computer Science Teachers Association (CSTA), the organization which promotes and supports CS teaching [21], suggests models for educating teachers relying upon the ACM model K12 curriculum of computer science. The paper suggests that any programme for preparing CS teachers must include the four main components: academic requirements in the field of computer science; academic requirements in the field of education; methodological (a methods course) and field experience; general pedagogical knowledge. National Council for Accreditation of Teacher Education (NCATE), the accreditation body in the USA for the accreditation of study programmes that educate future teachers, has developed (since 1990) a series of standards for preparing secondary CS teachers by promoting programmes based on K12 model curriculum. The proposal, shown in [22], lists knowledge and skills that CS teachers should have.

A detailed insight into the CS (informatics) teachers' curricula around the world (USA, Serbia, Israel, Estonia, Turkey, Austria, Germany, Scotland) as well as the analysis of reference CSTA/NCATE standards [21][22] and current literature [23] shows that informatics teacher curricula should cover the following content fields:

general knowledge, general educational and pedagogical knowledge, informatics domain knowledge, knowledge of teaching practice, knowledge of informatics teaching methods. These fields are mapped onto appropriate classes and are modelled as *Knowledge* subclasses (Fig. 1).

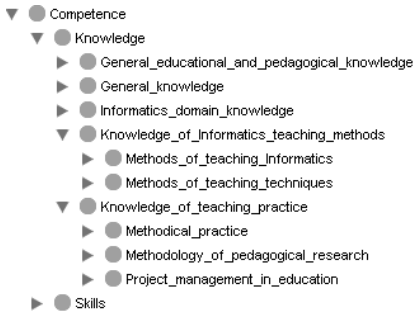


Fig. 1. Upper hierarchical structure of the *Knowledge_of_teaching_practice* class

The developed alignment software is applied in this work in order to compare ontological models of the reference to the selected curricula. NCATE/CSTA standards and curricula from different countries are mapped to the „reference" teacher education model. The study program for informatics and technology teachers is mapped to the "selected" teacher education model. The procedure of creating the *Knowledge* subclasses implied that the courses (or content fields) were mapped to direct subclasses of one of the five general knowledge areas (shown in fig. 1), while the topics contained in the courses were represented as lower subclasses. An additional description of the topics (usually shown in parentheses), if any, is mapped to the classes' labels. Figure 1 presents a part of the hierarchical structure of the *Knowledge_of_Informatics_teaching_methods* class in the selected curriculum model. The *Skills* class structure was created as follows: learning outcomes were classified based on the cognitive domain of the Revised Bloom's taxonomy and represented by lower *Skills* subclasses.

Chosen teacher education curricula model is based on the study program "Informatics and techniques in education" at the Technical faculty, Zrenjanin [24]. The method of mapping the curriculum into the ontological model is analogous to the procedure described in [17]. Also, the ontological models are given in owl format at [25].

4. Semantic Web Based Platform for Curricula Synchronization

The developed software platform is based on ontology alignment that is, in this paper, interpreted as a „set of correspondences" [26] between two ontological models of teacher education curricula curricula (O_1 and O_2). Object and datatype properties are predefined and the same in both ontologies, while instances are not included in the models. Therefore, the "set of correspondences" consists only of the classes' pairs (C_{i1},C_{j2}), similarity values (confidence degree - $conf_i$) between the classes and relations

among them (Equation 1). Possible relations between classes are: the superclass, the subclass and the equivalence.

$$\text{Alignment}(O_1, O_2)=\begin{cases} (C_{i1}, C_{j2}, \text{conf}_i, \text{relation}_i) \mid \\ C_{i1}\in O_1,\ C_{j2}\in O_2, \text{conf}_i\in[0,1], \\ \text{relation}_i\in\{=,\subseteq,\supseteq\} \end{cases} \quad (1)$$

The presented ontology alignment system predicts “one to one” and “one to more” (1:N) relationships. This means that a set of classes from one ontology can be the subclasses of a class from the other ontology. The ontology alignment is done in several steps. In the first phase, terminological similarity is determined. Similarity matrix, obtained by applying terminological matcher, represents the input for all following matchers. In the second phase, ontological matching methods are applied to determine taxonomic structural similarity, relational similarity and one-to-many similarities, respectively. The results of the application of each matcher represent the input for the next. The developed software platform provides the user with a change of the results after all matching steps, apart from after the terminological one.

A similarity matrix is created after each matching phase. It consists of the similarities of all possible the classes’ pairs of compared ontologies. The aim of using matchers is to establish “the best matched classes” i.e. to find the pairs of classes (for “1 to 1” relation) that are mostly alike (closest).

A determination of the best matched classes from the similarity matrix as well as a detailed description of each matcher is given in the following sections.

4.1. Determination of the Best Matched Classes

The problem of matching is well studied in literature dealing with graph theory [27], according to which it is possible to apply several criteria for determining the best matched pairs: maximum cardinality, maximum total "weight" and "stable marriage" Matching has maximum cardinality if it has the largest number of mappings (paired fields); matching has a maximum total weight if the sum of the weights of its mapping is the greatest; a "stable marriage" requires that there are no such combinations of paired fields (x, y) and (x1, y1) so that x more "prefers" y1 than y and y1 "prefers" x more than x1. According to [27], Greedy's choice for the matching of entities with cardinality of 1:1 can be considered as "monogamous" version of the "perfectionist egalitarian polygamy" selection metric which, according to the empirical results shown in [28] gives the best results in the matching scheme. In this paper, Greedy selection algorithm is used for determining the best classes’ pairs, because this method is frequently used in the ontology alignment systems like [29] and [30]. Also, authors in [31] state that, for example, a matching that maximizes the sum of the similarities of the selected pairs, is not an "optimal" solution for the problem of ontology alignment. As a reason for this claim, in [31] it is stated that the goal of an ontology alignment is to maximize the number of correct pairs and minimize the number of incorrect pairs. Therefore, in the context of ontology alignment consideration (assuming that the value of the similarity is directly related to the likelihood that the matching is true), selecting a pair with a high similarity (for example, over 90%) may be more correct than selecting two pairs with an average similarity value (50-60%).

The Greedy method [27], applied to a two-dimensional similarity matrix, may be described as follows [26].

1. Selecting a pair of entities $e_{m1} \in O_1$ and $e_{n2} \in O_2$ which has the highest similarity value of all entity pairs.
2. "Removing" rows and columns containing e_{m1} and e_{n2} so that e_{m1} cannot be paired with any $e_{j2} \in O_2$, $j \neq n$, and e_{n2} cannot be paired with any $e_{i1} \in O_1$, $i \neq m$.
3. Finding the greatest similarity between the remaining pairs of entities.
4. Repeating the process until one value in the similarity matrix remains.

In the system described in this paper, the obtained pairs of entities become the "best paired" if they are greater than the given threshold. Figure 2 shows an example of determining the paired entities by using the described method, with the threshold value of 0.5.

e_{11}	0.9	0.2	0.4
e_{21}	0.85	0.5	0.6
e_{31}	0.4	0.8	0.5

Fig 2. Example of determining the best pairs from the 2D similarity matrix

In the first step, the greatest possible similarity contained in the matrix is selected; that is, in the example from the image, 0.9, and the first pair of paired entities is: $\{e_{11}, e_{12}\}$. In the next step, the pair $\{e_{21}, e_{12}\}$ will not be selected, although their similarity is 0.85, since e_{12} has already been matched. The next greatest similarity among the remaining unpaired entities is, then, 0.8, therefore $\{e_{31}, e_{22}\}$ is the next best matched pair. In the last step $\{e_{21}, e_{32}\}$ are matched.

4.2. Terminological Similarity

Terminological similarity is determined using string and linguistic based method. String tokenization including string normalization methods (identification of numbers, special characters, blank spaces, uppercase to lowercase conversion, removing stop words etc.) precedes the establishing of string similarities. In this phase, strings contained in local names and class labels are taken into account. The English version of the WordNet lexical database is used for morphological linguistic normalization.

The similarity of tokens contained in the local classes' names is obtained using Lin's "information-theoretic" method [32] if both tokens are in the WordNet database. If at least one of the tokens is not in the WordNet dictionary, the similarity of the tokens is determined using the Jaro Winkler method [33], [34]. S_{ln} list, consisting of similarities of the "the best matched pairs" of tokens, is gained by applying the Greedy selection method to the matrix comprising the similarities of all tokens of the C_{i1} class with all tokens of the C_{j2} class. The total similarity of the local names for the two classes $s_{ln}(C_{i1}, C_{j2})$ is then calculated. A slightly different way is applied depending on whether the classes are subclasses of the *Knowledge* class or of the *Skills* class.

Skills subclasses represent skills/outcomes that are often described by free text. The difference in the number of words contained in the outcomes can significantly affect the different meaning of the outcome. Therefore, when calculating the similarity of the local names of the *Skills* subclasses, the number of tokens should be taken into account. Thus, $s_{ln}(C_{i1}, C_{j2})$ is calculated as follows.

$$s_{ln}(C_{i1}, C_{j2}) = \frac{2 \cdot \sum_{i=o}^{m} S_{ln}(i)}{|tok_{i1}| + |tok_{j2}|} \text{ ; } |tok_{ik}| \text{- \# of tokens in local name of } C_{ik}; \quad \textbf{(2)}$$

m-dimension of S_{ln}; $C_{ik}$is Skills subclass

It can be seen from the above formula that the similarity between the classes, described with the different number of words (tokens), is reduced.

On the other hand, for the *Knowledge* subclasses, which are usually described by a smaller number of tokens and which represent the names of topics/thematic areas, it has been experimentally shown that more accurate results are obtained if a different principle of calculating the similarity is applied. The principle of determining $s_{ln}(C_{i1}, C_{j2})$ of the *Knowledge* subclasses depends on the ratio of the difference in the number of tokens $\big||tok_{i1}| - |tok_{j2}|\big|$ and the minimum number of tokens $min(|tok_{i1}|, |tok_{j2}|)$. If the difference in the number of tokens is not less than the minimum number of tokens, the above formula (2) is applied. Otherwise, the total similarity of the local names of the two classes $s_{ln}(C_{i1}, C_{j2})$ is obtained as the average value of the elements of the list S_{ln}.

The similarity of the classes' labels $s_{lb}(C_{i1}, C_{j2})$, and the similarity between the local name of the class of one ontology and the label of the class of the compared ontology $s_{lnlb}(C_{i1}, C_{j2})$ and, inversely, $s_{lbln}(C_{i1}, C_{j2})$ is calculated in an analogous way. The total terminological similarity for classes $s_{term}(C_{i1}, C_{j2})$ is:

$$s_{term}(C_{i1}, C_{j2}) = \max(s_{ln}(C_{i1}, C_{j2}), s_{lb}(C_{i1}, C_{j2}), s_{lnlb}(C_{i1}, C_{j2}), s_{lbln}(C_{i1}, C_{j2})) \quad \textbf{(3)}$$

4.3. Taxonomic Structural Similarity

Taxonomic structural similarity includes the following stages: the determination of a parent similarity, the determination of the similarities of leaf classes and the determination of similarities of leaf classes belonging to the unmatched classes' structures. Since *Skills* part of the ontological hierarchy is less structured (classes representing a cognitive domain of Bloom taxonomy mostly have direct subclasses only), the taxonomic structural similarity is established only for the *Knowledge* subclasses. The classes in the upper classes' structure that are the same in both ontologies (like: *Knowledge*, *Competence*, *Informatics_domain_knowledge*, *General_knowledge*, etc.) are not considered in obtaining the taxonomic structural similarity.

Parent Classes' Similarity

In the first step, only the parent classes (classes that have subclasses) are compared. Parent classes' similarity is calculated based on the terminological similarity of the compared classes, the similarities of all superclasses (if they exist) and the similarities of all subclasses. Thereby, considered superclasses (parents) and subclasses (children) include direct and indirect classes in a parent/child relation. Thus, the similarity of the superclasses of the classes C_{i1} and C_{j2} $s^{sup}(C_{i1}, C_{j2})$ is determined as follows

/* Let A_{ij} be a class of an ontology| $A_{k1} \in O_1$ and $A_{l2} \in O_2$
if $\nexists A_{k1} | C_{i1} \subseteq A_{k1}$ or $\nexists A_{l2} | C_{j2} \subseteq A_{l2}$ then
$s^{sup}(C_{i1}, C_{j2})$ is not taken into account
else
Let $C_{i1} \subseteq \{A_{k1}\}$, k=1,n; n≥1 and $C_{j2} \subseteq \{A_{l2}\}$, l=1,m; m≥1
for k = 1 to n
for l = 1 to m
/* the values of the similarity of classes from the set $\{A_{11}, A_{21} \ldots A_{n1}\}$ with classes from $\{A_{12}, A_{22} \ldots A_{m2}\}$ become elements of matrix with n rows and m columns
matrix[k][l] = $s_{term}(A_{k1}, A_{l2})$
/* By applying Greedy selection method on the matrix list of the best matched superclasses' pairs S^{sup} is obtained
S^{sup} = Greedy_Selection_Method (matrix)

$$s^{sup}(C_{i1}, C_{j2}) = \frac{\sum_{i=o}^{m} S^{sup(i)}}{m}, \quad m = \text{size of } S^{sup}$$

The similarity of the subclasses $s^{sub}(C_{i1}, C_{j2})$ is calculated in an analogous way [16]. The total similarity $s_{parent}(C_{i1}, C_{j2})$ of classes C_{i1} and C_{j2} is calculated as follows

If $\exists A_{k1} | A_{k1} \subseteq C_{i1}$ and $\exists A_{l2} | A_{l2} \subseteq C_{j2}$ then

$$s_{parent}(C_{i1}, C_{j2}) = \frac{s^{sup}(C_{i1}, C_{j2}) + s^{sub}(C_{i1}, C_{j2}) + s_{term}(C_{i1}, C_{j2})}{n};$$

n=2 in case when $\nexists A_{k1} | C_1 \subseteq A_{k1}$ or $\nexists A_{l2} | C_{j2} \subseteq A_{l2}$
(when $s^{sup}(C_{i1}, C_{j2})$ is not taken into account.
Otherwise, n=3
else

$$s_{parent}(C_{i1}, C_{j2}) = 0$$

The resulting similarity matrix S_{parent} contains calculated similarities between all *Knowledge* subclasses (not including predefined classes) from ontology O_1 with all *Knowledge* subclasses of the O_2 ontology. The best matched classes from S_{parent} are achieved using Greedy selection algorithm with the given threshold.

Leaf Classes' Similarity

In the next phase, only the similarities of leaf classes are calculated. List (leaf) classes are classes that do not contain their own subclasses. The similarity values calculated by terminological matcher are allocated to the pairs of leaf classes if some of their parent

classes are matched by using previous matcher for determining parent similarity. If this is not the case, or if one of the compared classes has subclasses, their similarity is $s_{parent}(C_{i1}, C_{j2})$. In this stage, leaf classes and classes that have only leaf subclasses are also considered (by using the same principle of assigning similarity values for leaf classes). The reason for extending the algorithm to these classes is the possibility that a topics/thematic area in one of the curricula is described in more detail with leaf subclasses, although it is equivalent to the compared leaf class. This exception is used for the superclass/subclass relations described in the 1:N algorithm.

Unmatched Parents' Leaf Classes' Similarity

The classes belonging to the "unmatched classes structure" take into account the last step of the calculation of taxonomic structural similarity. The motivation for the introduction of this algorithm is the possibility that related thematic areas are represented by a different number of classes' structures but at the same hierarchical level. For example, thematic areas/courses related to the study of programming can be represented by two upper classes structure in one ontology (e.g. *Programming_languages*, *Object_oriented_ programming*), while in the other ontology related (or even the same) programming topics can be mapped onto subclasses of three or more upper classes' structure (e.g. *Programming_languages*, *Introduction_to_programming*, *Object_oriented_ programming*). In this case, it is possible that some related (or even the same) topics are mapped onto subclasses that would not be matched since their superclasses are not paired by the parent 1:1 matcher.

This algorithm uses disjoint property from the OWL (e.g. listed superclasses that represent programming concepts would not be signed as disjoint) in the following way [16]:

/* Let A_{leaf} be a list of matched classes obtained by a matcher that determines the similarity of the leaf classes of matched parents.
/* Let the following apply:
$\{\{A_{11}, A_{12}\}\ldots\{A_{n1}, A_{n2}\}\} \in A_{leaf}, C_{i1} \subseteq \{A_{11}\ldots A_{n1}\}$, $C_{j2} \subseteq \{B_{12}\ldots B_{m2}\}$, $A_{k2} \in \{A_{12}\ldots A_{n2}\}, B_{k2} \in \{B_{12}\ldots B_{m2}\}$
If C_{i1} and C_{j2} are unmatched leaf classes and $\nexists A_{k2}$, B_{k2} defined as disjoint classes and
$\nexists\{A_{11}, B_{k2}\} \mid \{A_{11}, B_{k2}\} \in A_{leaf}, A_{11} \in \{A_{11}\ldots A_{n1}\}, B_{k2} \in \{B_{12}\ldots B_{m2}\}$
then

$$s_{disj}(C_{i1}, C_{j2}) = s_{term}(C_{i1}, C_{j2})$$

else

$$s_{disj}(C_{i1}, C_{j2}) = s_{leaf}(C_{i1}, C_{j2})$$

4.4. Relational Similarity

The relational similarity is calculated only between the *Skills* subclasses as follows: if the compared classes C_{i1} and C_{j2} are connected via *hasKnowledge* object property to the

Knowledge subclasses that belong to paired classes structures (i.e. there is at least one pair of *Knowledge* subclasses $\{B_{k1}, B_{m2}\}$ obtained by the taxonomic matcher so that B_{k1} is related to C_{i1} and B_{m2} to C_{j2}), then pair $\{C_{i1}, C_{j2}\}$ gets a similarity value calculated by the terminological matcher.

4.5. 1:N matcher

1:N matcher pairs a class of one ontology with leaf classes of another ontology in "superclass/subclasss" relation, provided that the observed class of one ontology is matched with the parent class of the leaf classes of the other. It is based on the special case of the leaf matcher where a list class from one ontology and a class that has only leaf classes from another ontology are matched. The method for obtaining subclass relation can be described in the following way.

```
/* Let A_rel be a list of the matched classes calculated
by a previous relational matcher
If {C_i1, C_j2} ∈ A_rel and ∃A_l1 | A_l1 ⊆ C_i1 and ∄A_k2 | A_k2 ⊆ C_j2 then
If ∄{A_l1, A_m2} | {A_l1, A_m2} ∈ A_rel, A_l1 ∈ O_1, A_m2 ∈ O_2, A_l1 ∈ {A_11…A_n1},
{A_11…A_n1} ⊆ C_i1, n ≥1 then C_j2 ⊇ {A_12…A_n1}
/*{A_12…A_n1} from ontology O_1 are subclasses of the C_j2
                    class from O_2 ontology
```

Superclass relation is obtained in an analogous manner.

4.6. Graphical User Interface

The graphical user interface (GUI) of the developed software platform gives an overview of the ontologies' hierarchical structure, the information about the classes and the matching results. The user can choose the input ontological models and the alignment level (the alignment of secondary school and teacher education curricula models or the alignment of teacher education curricula models). The system allows the user to enter the threshold value as well.

The GUI shows the opened ontological models in a tree structure. The classes' information is shown in the panels on the left side (Figure 3). The class' information contain related comments (if they are entered), a label, an associated class (by *hasKnowledge/hasSkill* object property) and the class to which it is matched. The results' statistics is shown in a separate tab (Figure 4).

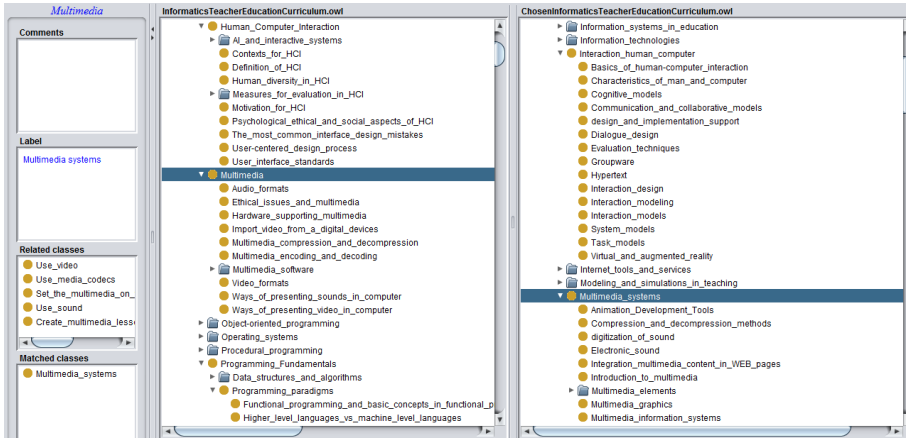


Fig. 3. A part of the classes' structure of the teacher education curricula models.

After the application of each matcher type, the tables display the matching results in the "Alignment output" tab (Figure 4). The tables contain paired classes, the relation type (superclass, subclass, equivalence) and the similarity value. When using a relational matcher, the table gets an additional column ("Bloom") that reflects the consistency of the cognitive domain of Bloom's taxonomy.

Since the local names of some classes (especially *Skills* subclasses) are the result of free text mapping (contained in learning outcomes), it was necessary to ensure that the system is semi-automatic. Hence, the GUI allows the user to improve system accuracy. Matching results can be changed in "Ontologies" tab (by using drop down menu) and "Alignment output" tab (by choosing the pairs of classes to be matched and by entering the similarity values as well as relation type). The equivalence relation is the only possible type of a relation for all matchers except for the last one. Therefore, the user can choose one of the following relations {= - equivalence, ⊆ - superclass, ⊇ - subclass} only after the application of the 1:N matcher. Manual interventions include the following possible actions:

– Adding matched classes' pair,
– Replacing the class belonging to the matched pair with another class,
– Changing the obtained similarity value for a pair of the matched classes,
– Removing the classes' pair that is not matched correctly,
– Editing the threshold value.

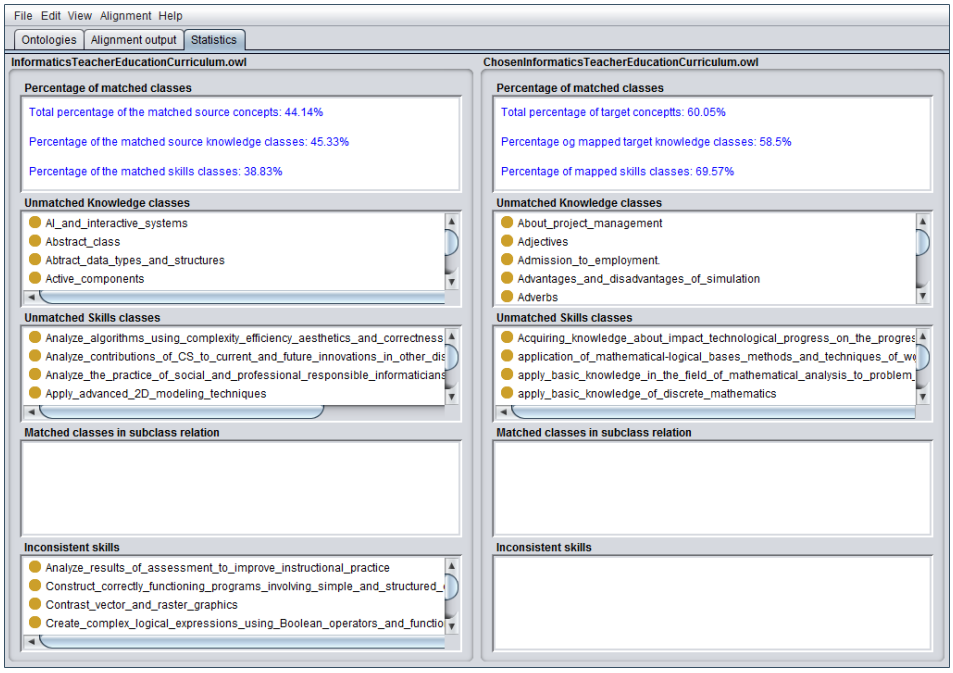


Fig. 4. Statistical presentation of results.

However, adding a matched pair of classes may include:

– Adding a new pair $\{C_{i1}, C_{j2}\}$, where both classes are unmatched.
– Adding a new classes' pair $\{C_{i1}, C_{j2}\}$, where C_{i1} or C_{j2} has already been matched, i.e.: $\exists\{C_{m1,}\, C_{j2}\} \mid$ m≠i ∨ $\exists\{C_{i1,}\, C_{k2}\} \mid$ k≠j.
– Adding a new pair $\{C_{i1}, C_{j2}\}$, where both classes are matched, i.e.: $\exists\{C_{i1,}\, C_{m2}\} \mid$ m≠j ∧ $\exists\{C_{k1,}\, C_{j2}\} \mid$ k≠i.

The type of a new relation and the type of the existing relations (for cases 2 and 3) with a class from another ontology are taken into account when creating new pairs of classes. Thus, for example, for the case 2:

– if the user defines the relation of equivalence$\{C_{i1}, C_{j2}\}$, where $\exists\{C_{k1}, C_{j2}\} \mid$ k≠i then
– the relation $\{C_{k1}, C_{j2}\}$ is deleted and a new pair $\{C_{i1}, C_{j2}\}$ is established, since one class can participate in no more than one equivalence relation with a class of other ontology.

5. Application of the Software Platform to the Ontological Models of the Teacher Education Curricula

Figures 5 -7 show a part of the results obtained after applying the developed software system on the input informatics teacher education models. The column "Source class"

contains reference teacher education model' classes, while the classes belonging to the chosen Informatics teacher education model are contained in the "Target class" column.

5.1. The Application of the Algorithm for Calculating the Taxonomic Structural Similarity

Figure 5 shows paired classes and similarity values obtained after applying all three phases of the taxonomic structural algorithm. The possible lower similarity values obtained when comparing the parent classes were taken into account in the experimental determination of the threshold value (70%). Thus, pairs of parent classes (shown in rows 11, 12, 14, 16, 17, 21, 23, 26, 31; Figure 5) have a similarity value below 85%, although they have a similar or identical local name. Such results can be considered as an expected consequence of calculating the similarity of the parent classes which includes the similarities of all subclasses and superclasses. There are also pairs of parent classes whose similarity value is higher, since they belong to very close hierarchical structures (rows 7, 8, 15, 24, 30, 33). It can be seen from Figure 5 that the classes related to the mathematical fields are matched with each other, although only an insight into the names of the classes does not indicate these results (rows 3, 28, 29). However, by looking at the hierarchical structure, it can be concluded that the matching is correct.

When comparing leaf classes, the similarity value more significantly corresponds to the probability that the classes are correctly matched. This is indicated by the pairs of leaf classes shown in rows 6, 9, 13, 19, 20, 25, 32. However, correctly obtained pairs of leaf classes with slightly lower similarity values are possible (rows 2, 10). The pairs of classes contained in rows 5, 18 and 27 can be considered as incorrect results of the application of the taxonomic structural matcher. From Figure 4 it can be seen that the pairs of classes in rows 5 and 18 belong to paired class structures {*Multimedia*, *Multimedia_systems*} and {*Human_Computer_Interaction*, *Interaction_human_computer*}, respectively. These results are a consequence of the principle according to which the system searches, lexically, the closest pairs of classes (with a threshold of 70%) among the subclasses of paired parents. Figure 5 also shows pairs of classes (rows 1 and 4) obtained by applying the third phase of the taxonomic structural matcher, i.e. by searching the classes in unpaired and non-disjoint class structures. The influence of the classes' label on the similarity value can be seen from the correctly obtained pair of classes given in row 22. After applying the taxonomic structural matcher, the percentage of the paired *Knowledge* subclasses of the reference model is 46.1%, while the percentage of pairing of the *Knowledge* subclasses of the selected model is 59.49%.

Row	Source class	Target class	Type of ...	Similarity v...
1	Aggregation	Aggregations	Equival...	100.0%
2	Aims_and_objectives_of_upbringing	Goals_of_education_and_teaching	Equival...	81.18%
3	Algebra	Mathematics_3	Equival...	84.3%
4	Animation	Animation	Equival...	100.0%
5	Audio_formats	Electronic_sound	Equival...	78.33%
6	Complex_numbers	Complex_numbers	Equival...	100.0%
7	Computer_Networks	Computer_networks	Equival...	93.48%
8	Computer_design	Computer_systems	Equival...	87.49%
9	Connection_to_Database	Connecting_to_databases	Equival...	98.0%
10	Criteria_for_quality_of_software	Quality_assessment	Equival...	80.08%
11	Data_structures_and_algorithms	Data_structures	Equival...	83.9%
12	Database	Databases	Equival...	84.92%
13	Deterministic_finite_state_machine	Finite_automata	Equival...	98.06%
14	Didactics	Didactics	Equival...	84.32%
15	E-learning_types	E-learning	Equival...	91.08%
16	Educational_psychology	Pedagogical_psychology	Equival...	75.71%
17	Educational_software	Educational_software_design	Equival...	75.28%
18	Entering_data	Basic_data_type	Equival...	81.67%
19	Entity	Entity	Equival...	100.0%
20	Ethernet	Ethernet	Equival...	100.0%
21	Evaluation_of_instruction	Evaluation_of_teaching_work	Equival...	71.03%
22	Firewalls	Attack_and_protection_of_computer_systems	Equival...	100.0%
23	General_Pedagogy	Pedagogy	Equival...	82.99%
24	Graphics_design_elements	Design_elements	Equival...	98.64%
25	Homomorphisms	Homomorphisms	Equival...	100.0%
26	Human_Computer_Interaction	Interaction_human_computer	Equival...	81.91%
27	Human_diversity_in_HCI	Characteristics_of_man_and_computer	Equival...	71.34%
28	Mathematical_Analysis	Mathematics_2	Equival...	81.11%
29	Mathematics	Mathematics_1	Equival...	89.04%
30	Modeling_and_simulation	Modeling_and_simulations_in_teaching	Equival...	89.04%
31	Models_and_phases_of_the_software_development_pr...	Software_processes_and_specifications	Equival...	74.96%
32	Multimedia_compression_and_decompression	Compression_and_decompression_methods	Equival...	100.0%
33	Programming_Fundamentals	Programming_languages	Equival...	90.56%
34	Software_engineering	Software_engineering	Equival...	85.51%
35	Structured_and_Imperative_programming	Structured_programming	Equival...	90.47%

Fig. 5. Part of the matched parent classes of the teacher education curricula.

5.2. The Application of the Algorithm for Calculating the Relational Similarity

The names of the *Skills* subclass represent the free text contained in learning outcomes. Therefore, the threshold value was experimentally set to a lower value than in the previous phase (55%). The "Bloom" column contains the T mark if the *Skills* subclass of the selected model corresponds to a higher level of the Bloom taxonomy cognitive domain than the corresponding subclass of the reference model. Conversely, the "Bloom" column is false.

R...	Source class	Target class	Type ...	Simil...	Bloom
1	Apply_educational_software	Evaluate_educational_software	Equiv...	85.0%	⊤
2	Contrast_vector_and_raster_graphics	Master_the_basic_concepts_of_computer_graphics	Equiv...	71.6%	⊥
3	Create_e_learning_content	Analysis_of_tools_for_creating_e-learning_systems	Equiv...	72.01%	⊥
4	Design_application_communicationg_and_maintaining_databases	Understand_the_components_of_database_management_software	Equiv...	72.18%	⊥
5	Design_databases	Design_a_database	Equiv...	100.0%	⊤
6	Design_interactive_user_interfaces_for_diverse_applications	Design_user_interfaces	Equiv...	74.7%	⊤
7	Design_web_pages	Creates_a_website	Equiv...	72.32%	⊤
8	Identify_the_national_high_school_CS_curriculum_intending_to_teach.	Knows_the_valid_curriculum_of_informatics_in_primary_and_seco...	Equiv...	62.77%	⊤
9	Implement_basic_algorithms	Problem_solving_through_algorithms	Equiv...	71.29%	⊤
10	Implement_of_programs_of_sufficient_complexity	Implementation_user_interfaces_of_computer_systems	Equiv...	71.52%	⊤
11	Maintain_computer_system	Know_organization_of_computer_systems	Equiv...	70.24%	⊥
12	Select_appropriate_e_learning_approach_to_teach_a_specific_content	Analyze_different_e_learning_approaches	Equiv...	63.61%	⊤
13	Set_the_multimedia_on_the_web	Create_multimedia_presentations	Equiv...	62.74%	⊤
14	Teach_CS_lessons_using_multiple_forms_of_media	Organize_teaching_material_in_the_form_of_educational_software	Equiv...	64.59%	⊤
15	Uderstand_3D_modelling	Create_a_3d_scene	Equiv...	76.89%	⊤
16	Understand_assembler_programming	Use_of_assembly_language	Equiv...	75.19%	⊤
17	Understand_computer_networks_supporting_communication_and_collab...	Configuring_computer_networks	Equiv...	59.2%	⊤
18	Understand_concepts_and_assertions_of_mathematical_analysis_and_...	Acquire_basic_concepts_of_mathematical_analysis	Equiv...	74.13%	⊤
19	Understand_data_representation_and_organization_at_the_machine_level	Understanding_the_work_of_computer_systems	Equiv...	79.3%	⊤
20	Understand_machine_level_components_and_related_issues_of_compl...	Understand_the_structural_organization_of_computers_at_multiple...	Equiv...	67.88%	⊤
21	Use_Modeling_and_simulation_to_solve_real_world_problems	Uses_modelling_and_simulations_in_teaching_informatics	Equiv...	72.28%	⊤
22	Use_UML_for_modeling_meaningful	Application_of_UML	Equiv...	63.05%	⊤
23	Use_of_Internet_in_a_safe_and_efficient_manner	Using_e-mail_services_and_www	Equiv...	57.25%	⊤
24	Use_programs_for_computer_graphics	Uses_raster_graphics_programs	Equiv...	90.63%	⊤
25	Use_programs_for_text_presentations_spreadsheets	Use_of_MS_OFFICE	Equiv...	83.33%	⊤

Fig. 6. Matched *Skills* subclasses of the curricula models.

Considering that the names of the *Skills* subclasses represent a free text and that the similarity value is performed using a terminological matcher (if some of the *Knowledge* classes' structures with which the *Skills* subclasses are associated are matched), matching accuracy is lower as expected. Thus the pairing results shown in rows 2, 3, 10 and 19 can be considered incorrect. The obtained classes' pairs, shown in rows 1, 4, 12, 13, 15, 16, 17 and 18 (Figure 6), regardless of the fact that they represent different levels of the Revised Bloom taxonomy, can be considered as the correct result of matching.

69.57% of the *Skills* subclasses of the selected model are matched after the application of the relational matcher, while the percentage of the matched *Skills* subclasses of the reference model is 38.83%.

The unmatching of the *Skills* subclass is mainly a consequence of the unmatching of the *Knowledge* subclasses with which they are associated. Thus, classes like *Implement_knowledge_representation_and_reasoning_system*, *Assess_possible_applications_and_limitations_of_the_Artificial_Intelligence* (associated with the unmatched parental class *Artificial_intelligence*), *Discuss_intellectual_property*, *Analyze_the_practice_of_social_and_professional_responsible_informaticians* (associated with the unmatched parental class Computer_ethics), etc. remain unmatched.

5.3. The Application of the Algorithm for Calculating 1:N Similarity

Figure 7 shows characteristic pairs of classes in a 1:N relation. Figure 7 shows an example of a superclass relation. The *Management_in_education* class of the reference curriculum model (described additionally by the "School management" label) has no further subclasses. It is paired with the *Organization_school_work* class of the selected curriculum model, and has become a superclass of all the *Organization_school_work* class' subclasses.

...	Source class	Target class	Type of relati...	Similarity val...
1	Management_in_education	Organization_of_school_work	Equivalence	75.31%
2	Management_in_education	Admission_to_employment	Superclass	75.31%
3	Management_in_education	Development_plan_institutions	Superclass	75.31%
4	Management_in_education	Information_system_primary_and_secon...	Superclass	75.31%
5	Management_in_education	Inspection	Superclass	75.31%
6	Management_in_education	Institution_bodies	Superclass	75.31%
7	Management_in_education	Professional_bodies	Superclass	75.31%
8	Management_in_education	Institutions_for_development_and_qualit...	Superclass	75.31%

Fig. 7. Matched class in 1:N relations.

5.4. Discussion of the Results

After the application of all phases of ontological pairing, there are *Knowledge* and *Skills* classes that remain unmatched. The reasons for their unmatching can be classified into two basic categories. One is the lack of topics (thematic areas) and/or learning outcomes in the compared curricula. The second refers to the possible shortcomings of the applied

algorithms, which results in some classes corresponding to the equivalent knowledge/outcomes not being matched. From this it can be concluded that it is either necessary to improve the curricula so that they contain all the required knowledge and outcomes or it is necessary to improve the software platform so that it finds all pairs of classes that represent the same aspects of the compared curricula.

Moreover, a part of the *Knowledge* subclasses remains unpaired as a consequence of the different levels of the description of certain thematic areas in the compared curricula. This especially refers to the pedagogical, didactic and mathematical courses of the selected teacher's curriculum from the Republic of Serbia, which contains a large number of topics (thematic areas). The unmatching of the classes representing these courses does not necessarily mean that they are not included in the compared (reference) curriculum, but may indicate that the same aspects of the curriculum are described by a different number of topics. The possible solution to this type of unmatching is twofold. One direction would be to improve the curriculum by describing the courses in more detail in accordance with the compared curriculum. Another solution is to upgrade the software platform so that algorithms that include “more to more” connections are implemented.

The evaluation of the applied algorithms in our software platform was realized by comparison with test/reference results. Precision and recall are "the most common comparison criteria" [26]. These measures are based on a comparison of the expected and obtained results of the analyzed system. In the context of ontology matching, the alignment obtained from a system that is the subject of evaluation (A) is compared with the reference alignment (R). Precision P is the ratio of the number of correctly found correspondence and the total number of obtained correspondence. Recall R is the ratio of the number of correctly found correspondence and the total number of expected correspondence. It is stated in literature [26] that it is sometimes desirable to consider only one value as a result of comparing the system. However, the systems are often not "comparable" applying only precision or only recall. For example, a system having high recall may have a low precision and vice versa. Therefore, evaluation of the system for ontology alignment [35] usually entails the use of the F-measure that combines precision and recall.

Analogous to [16][17], a team made up of educational experts evaluated the software platform. The expert team consisted of 4 university teachers (in the field of methods of teaching informatics), 2 employees in School Administration (Ministry of Education, Science and Technological Development) and 2 secondary school informatics teachers. The expert team determined the reference alignment (expected pairs of classes) for all possible curricula pairs, i.e. assessed the accuracy of the results obtained by applying the software platform. The process of defining a reference alignment consisted of two main steps. The first was a detailed analysis of the compared ontological models of the curricula done by the team of experts. After that, the expert team defined the reference alignment by finding pairs of classes that represent equivalent knowledge or skills (learning outcomes). In this case, with the exception of the superset/subset relation, one class can be in only one classes’ pair. Thus, the resulting reference alignment contains the exact set of classes‘ pairs that the software platform should ideally provide, according to the expert team. Also, the team of experts analyzed the obtained pairs of classes after applying the platform to the input ontological models. The aim was to determine the number of the correctly obtained pairs of classes as well as the total

number of obtained classes, i.e. to determine the values of the parameters needed for calculating precision, recall and f-measure.

In this section the results obtained by comparing teacher education curricula are analyzed regarding the results obtained by comparing the secondary ACM K12 curriculum model and the teacher education reference curriculum model [16], and regarding the results obtained by comparing secondary ACM K12 curriculum model and the chosen teacher education curriculum model [24] in the manner described in [17]. The values of precision, recall and F-measure for all three combinations of input ontological models are shown in Figure 8. Figure 9 shows the percentage of the matched *Knowledge* and *Skills* subclasses.

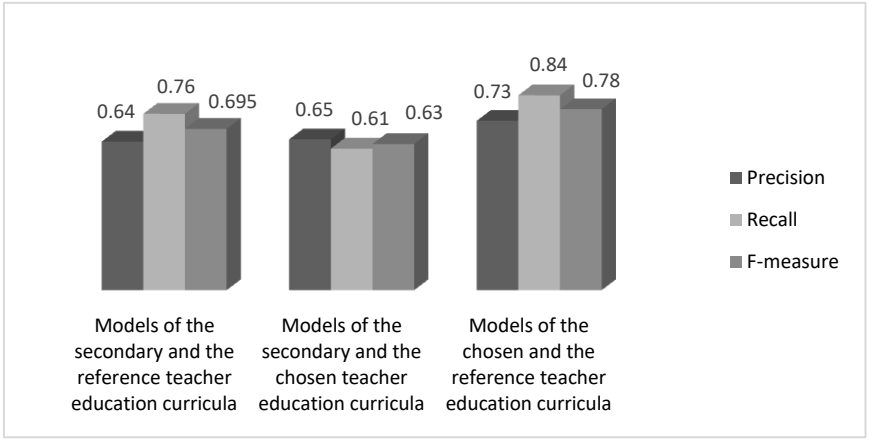


Fig. 8. System evaluation for all three combinations of the ontological models

The analysis shows that the values of precision (0.73), recall (0.84) and F-measure (0.78) are the highest when models of teacher education curricula are compared. Although this can be seen as an expected consequence of comparing the curricula of the same level of education, the results are satisfactory, especially as the largest number of classes was compared (ontological models of teacher education curricula individually contain significantly more classes than high school curriculum model).

It can be noticed that in the first and third case (Figure 8) the higher values of recall than the values of the precision were obtained (in the second case the value of recall is close to the value of precision). These results (along with satisfactorily high precision value) are in accordance to the reference [36] where "highest priority" is given to the recall when the ontological matching is a semi-automatic process [16]. In [36], (p 630) states that “since the burden of deleting false identified pairs by a platform is minimal compared to the burden of traversing two heterogeneous ontologies that might include thousands of concepts and attributes and identify similar entities, recall is a much more important measure”.

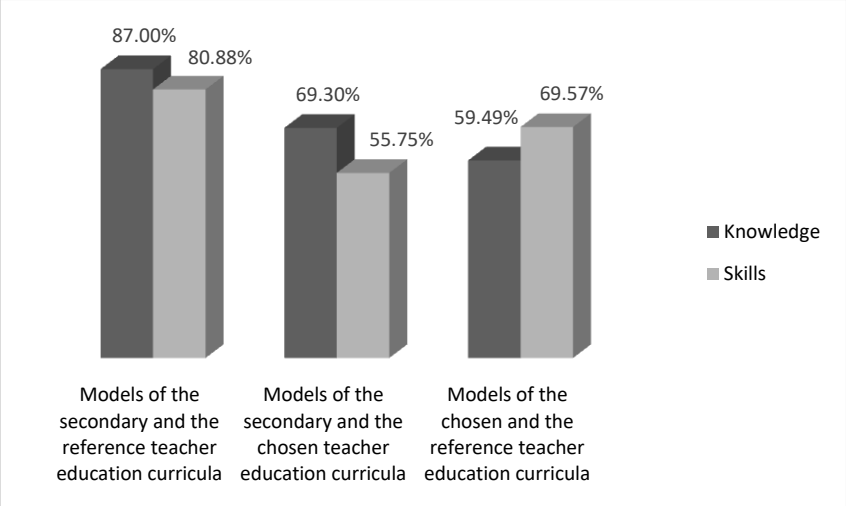


Fig. 9. The percentage of matched classes for all three combinations of the ontological models

From Figure 9 it can be observed that the percentage of the matched *Knowledge* subclasses of the secondary school model is lower when it is compared with the chosen teacher education curriculum model than when it is compared with the reference teacher education curriculum model. This could be explained by different taxonomical structure of the ontological curricula models (structurness of the chosen teacher education curriculum model is the lowest). That is especially true for "Connection between mathematics and computer science" topic [16], [17]. Still, a lower percentage of matching in the second case (Figure 9) primarily indicates that the chosen teacher education curriculum does not provide the study of thematic areas corresponding to the unmatched classes of the secondary model.

The higher level of matching between the secondary model and the reference teacher education curriculum model is expected, since the reference model of the teacher education curriculum was created according to the recommendations of international accreditation bodies and the analysis of more than 20 national and international curricula.

The percentages of the matched *Knowledge* subclasses are the lowest when teacher education curricula models are compared (third case in Figure 9). For such combination of input ontological models, a higher percentage of the matched *Skills* subclasses is obtained (compared to matched *Knowledge* subclasses). These results can be considered as a consequence of a large number of differently structured *Knowledge* subclasses when compared teacher education curricula models. Still, a large number of *Knowledge* subclasses is "correctly unmatched" (there are missing thematic areas/courses in the compared curricula). The percentages of the matched *Knowledge* and *Skills* subclasses of the reference model (the opposite case) are not shown in the table and are similar to the results of the matching of the chosen teacher education curriculum model. The results of the comparison of teacher education curricula models (section 5) are in accordance with the results of the comparison of the other combination of input ontological models. For example, when comparing the selected teacher education curriculum model and ACM K12 model, the system has shown that the concepts of

artificial intelligence are not taught in the selected teacher education curriculum in the Republic of Serbia. On the other hand, when comparing the ACM K12 model and the reference model of the teacher education curriculum [17], classes' structures that correspond to these concepts are mutually matched. From these results it can be assumed that the correct result of comparing teacher education curricula models would be an unmatched class of the reference model representing the principles of artificial intelligence, which is obtained by the application of the software platform.

6. Conclusions and Future Work

The semi-automatic software platform presented in this paper contains modified ontological matching algorithms, which enables the comparison of ontological curricula models of the same level of education i.e. informatics teacher education curricula. The main contributions of this paper are threefold: 1) model of the selected informatics teacher education curriculum has been created 2) the part of matching algorithms, adapted to teacher education curricula models, has been developed 3) comprehensive evaluation of the software platform and curricula models has been conducted. The evaluation was realized by unifying and comparing the results obtained for all three combinations of input ontological models. The results indicate the lack of specific knowledge (representing pedagogical, didactic and mathematical thematic areas) and skills in the analyzed curricula, but also the different structure of the ontological models. Therefore, it is necessary to consider the improvement of the curricula as well as the introduction of new matching algorithms that would find equivalent class belonging to related hierarchical structures. Also, the values of precision, recall and f-measure are lower when comparing curricula of different levels of education than when comparing teacher education curricula. Hence, in the case of matching secondary and teacher education models, the need for manual user interventions is greater, which can be considered as the expected result of the software platform application. Inversely, the matching of the *Knowledge* and *Skills* subclasses is greatest when comparing the secondary and teacher education curriculum (especially the reference model of teacher education curriculum).

One of the directions of further research refers to the creation of an ontological model in such a way that it also contains other important aspects of the curriculum, such as assessment methods, learning goals, anticipated literature and the like. Also, it is necessary to explore the possibility of a semi-automatic mapping of informatics curricula to ontological models. Other directions of future research are related to the limitations of the software platform and the possibilities of improving its accuracy. This primarily refers to the applied matching algorithms. Thus, when comparing a series of words, either in the name of thematic areas (*Knowledge* subclasses) or in the name of learning outcomes (*Skills* subclasses), the terminological matcher does not take into account the nature of the domain (Computer science/Informatics) or the syntax of the language. The similarity of the classes' name depends on the similarity of the separate words (tokens). For example, in the WordNet dictionary, the same word for computer science domain can have a completely different meaning. Thus, the word ontology (WordNet Search, version 3.1) is defined as "a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities

and their relations" (for Computer science domain) or "the metaphysical study of the nature of being and existence" (in general). Therefore, it is necessary to consider improvements in the terminological matcher in some of the following ways: using the semantic domain of WordNet dictionaries, using external dictionaries, using external computer ontologies or using ACM computer classification.

Since English can be seen as a de facto standard for international recommendations for Computer science curricula (ACM, CSTA) the software platform uses the English version of the WordNet dictionary and the ontological models are written in English. Therefore, another future research aim is to provide conditions for comparing ontological models in different languages (using the "Inter-Lingual Index" component of the WordNet dictionary) or comparing models whose classes' names are in the same language.

Also, future work will be focused on improving the system performance by applying the method for the initial rejection of classes that will not be considered. Also, it is necessary to investigate the accuracy of results with manual interventions of users in different stages of alignment. One more research aim is to map more curricula onto ontological models and to use the software platform to examine the accuracy of the results and the compliance of the curricula.

References

1. Seitz, P.: Curriculum Alignment Among the Intended, Enacted, and Assessed Curricula for Grade 9 Mathematics. Journal of the Canadian Association for Curriculum Studies, Vol. 15, No. 1, 72- 94. (2017).
2. Bay E.: Developing a Scale on Factors Regarding Curriculum Alignment. Journal of Education and Training Studies. Vol. 4, No. 5, 8-17. (2016)
3. Wijngaards-de Meij, L., Merx, S.: Improving curriculum alignment and achieving learning goals by making the curriculum visible. International Journal for Academic Development, Vol. 23, No. 3, 219-231. (2018)
4. Shaltry, C.: A new model for organizing curriculum alignment initiatives. Advances in physiology education, Vol. 44, No. 4, 658 – 663. (2020)
5. Ruge, G., Tokede, O., Tivendale, L.: Implementing constructive alignment in higher education – cross-institutional perspectives from Australia. Higher Education Research & Development, Vol. 38, No. 4, 833-848. (2019)
6. Zhu, Y C., Zhang, W., He, Y., Wen, J B., Li, M Y.: Design and Implementation of Curriculum Knowledge Ontology-Driven SPOC Flipped Classroom Teaching Model. Educational Sciences: Theory & Practice, Vol. 18, No. 5, 1351-1374. (2018)
7. Katis, E., Kondylakis, H., Agathangelos, G., Vassilakis, K.: Developing an Ontology for Curriculum and Syllabus. In: Gangemi A. et al. (eds) The Semantic Web: ESWC 2018 Satellite Events. ESWC 2018. Lecture Notes in Computer Science, Vol. 11155, 55-59. (2018)
8. Elsayed, E.: Interaction with Content through the Curriculum Lifecycle. Advanced Learning Technologies, ICALT 2009, 730 – 731. (2009)
9. Sarmiento, C., Duarte, O., Barrera, M., Soto, R: Semi-automated academic tutor for the selection of learning paths in a curriculum: An ontology based approach. Proc. of IEEE 8th International Conference on Engineering Education, 223-228. (2016)
10. Zhu, Y. C., Zhang, W., He, Y., Wen, J. B. , Li, M. Y.: Design and implementation of curriculum knowledge ontology-driven SPOC flipped classroom teaching model. Educational Sciences: Theory & Practice, Vol. 18, No. 5, 1351–1374. (2018)

11. Modiba, N., Ojo, S., Ncube Z.: An Ontology Based Model for Cyber Security Awareness Education, in Kennedy Njenga (editor). Proceedings of 4th International Conference on the Internet, Cyber Security and Information Systems, Vol 12, 169–179. (2019)
12. Ngo, D., Bellahsene, Z.: Overview of YAM++—(not) Yet Another Matcher for ontology alignment task. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 41, 30–49. doi:10.1016/j.websem.2016.09.002 (2016)
13. Fiallos, A., Ochoa, X.: Semi-automatic generation of intelligent curricula to facilitate learning analytics. Proceedings of the Ninth International Conference on Learning Analytics & Knowledge, Tempe, AZ, USA, 46–50. (2019).
14. Shao, C., Hu, LM., Li, JZ.et al.: RiMOM-IM: A Novel Iterative Framework for Instance Matching. Journal of Computer Science and Technology, Vol. 31**,** 185–197. (2016). https://doi.org/10.1007/s11390-016-1620-z
15. Nkisi-Orji, I., Wiratunga, N., Massie, S., Hui, KY., Heaven, R.: Ontology Alignment Based on Word Embedding and Random Forest Classification. Springer Theses, 557–572. (2019) doi:10.1007/978-3-030-10925-7_34
16. Mandić, M., Konjović, Z., Ivanović, M.: Platform for Computer-aided Harmonization of Informatics Curricula. Acta Polytechnica Hungarica, Vol. 13, No. 3. 159-179. (2016)
17. Mandić, M., Konjović, Z.: Collaborative development of informatics curricula based on semantic technologies. International Scientific Conference On Ict And E-Business Related Research – Sinteza, Ict & Business, 3-9. (2016)
18. Mandić, M., Konjović, Z., Ivanović, M.: Ontological Model of the Standardized Secondary School Curriculum in Informatics. Proceedings of the 5th International Conference on Information Society and Technology, 363-367. (2015)
19. Stephenson, C., Gal-Ezer, J., Haberman, B., Verno, A.: The new educational imperative: Improving high school computer science education. Final report of the CSTA Curriculum Improvement Task Force. New York: Computer Science Teachers Association. (2005)
20. Fuller, U., Johnson, CG., Ahoniemi, T., Cukierman, D., Hernán-Losada, I. et al.: Developing a computer science-specific learning taxonomy. ACM SIGCSE Bulletin, Vol. 39, No. 4, 152-170. (2007)
21. Gal-Ezer, J., Stephenson, C.: Computer science teacher preparation is critical. ACM Inroads. Vol. 1, No. 1, 61-66. (2010)
22. East, JP., Bentley, C., Kmoch, J., Rainwater, S., Stephenson, C.: NCATE standards for preparation of secondary computer science teachers. Proceedings of the 42nd ACM technical symposium on Computer science education, 243-244. (2011)
23. Hazzan, O., Ragonis, N., Lapidot, T.: Guide to Teaching Computer Science.Springer, Cham. (2020). doi:10.1007/978-3-030-39360-1
24. Technical faculty "Mihajlo Pupin": Informatics and techniques in education, Zrenjanin. (2013). [Online].http://www.tfzr.uns.ac.rs/Content/akreditacija/Informatika%20i%20tehnika%20u%20obrazovanju%20master.pdf
25. Ontological models of the informatics teacher education curricula. http://www.pef.uns.ac.rs/InformaticsTeacherEducationCurriculum/index.html
26. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, Berlin-Heidelberg. (2007)
27. Wu, W., Yu, C., Doan, A., Meng, W.: An Interactive Clustering-based Approach to Integrating Source Query interfaces on the Deep Web. Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, 95-106. (2004)
28. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm. In *Proceedings of the 18th International Conference on Data Engineering*. (ICDE), 117-128. (2002)
29. Huber, J., Sztyler, T., Noessner, J., Meilicke, C.: CODI: Combinatorial Optimization for Data Integration : results for OAEI 2011. In Proc. 6th ISWC workshop on Ontology Matching (OM), 134–141. (2011)

30. Ngo D, Bellahsene Z, Coletta R.: YAM++ - Results for OAEI 2011. In Proceedings of the 6th International Workshop on Ontology Matching (OM-2011), Vol. 814, 228–235 (2011)
31. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, IF., Couto, FM.: The agreementmakerlight ontology matching system. In: Meersman, R., Panetto, H., Dillon, T., Eder, J., Bellahsene, Z., Ritter, N., De Leenheer, P., Dou, D. (eds.); Heidelberg: Springer; LNCS, vol. 8185, 527–541. (2013)
32. Lin, D.: An information-theoretic definition of similarity. *Proceedings of the 15th International Conf. on Machine Learning*, 296–304. (1998)
33. Jaro, M.: UNIMATCH: A record linkage system: User's manual. Technical report. Washington DC: U.S. Bureau of the Census. (1976)
34. Jaro, M.: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association, Vol. 84, No. 406, 414–420. (1989)
35. Grau, BC., Dragisic, Z., Eckert, K., Euzenat, J, et al.: Results of the ontology alignment evaluation initiative 2013. In: *Proc. 8th ISWC workshop on ontology matching (OM)*. 61–100 (2013)
36. Stoilos, G., Stamou, G., Kollias, S.: A String Metric for Ontology Alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) *ISWC* 2005. LNCS, Vol. 3729, 623–637. (2005)

Milinko Mandić received his MSc degree in the field of electrical and computer engineering from the Faculty of Technical Sciences, University of Novi Sad, Serbia. He obtained his PhD degree in Methods of teaching informatics from the Faculty of Sciences, University of Novi Sad. He is currently an Assistant Professor at the Department of Informatics and Media, Faculty of Education, University of Novi Sad. His research interests include informatics curricula (for informatics teachers and for the primary and secondary level of education), Semantic Web and ontologies, methods of teaching informatics, e-learning standards, applying social and collaborative software in education.

Received: February 07, 2021; Accepted: July 08, 2021.

Computer Science and Information Systems 19(1):251–275 https://doi.org/10.2298/CSIS210410051S

How MCDM Method and the Number of Comparisons Influence the Priority Vector

Zorica Srđević, Bojan Srđević[1], Senka Ždero, and Milica Ilić

Faculty of Agriculture, Department of Water Management,
Trg D. Obradovića 8, 21000 Novi Sad, Serbia
{zorica.srdjevic, bojans, senka.zdero, milica.ilic}@polj.uns.ac.rs

Abstract. One of the most important issues in multi-criteria decision making is the number of requited judgments decision-maker/analyst has to perform. This paper presents a comparison of the results obtained by standard analytic hierarchy process (AHP), limited AHP, and best-worst method (BWM) if the number of criteria is 6, 7, and 8. The examples show that BWM's results are comparable with the results if standard AHP is used, while the limited version of AHP is generally inferior to the other two methods.

Keywords: analytic hierarchy process, best-worst method, criteria, comparison matrix size, number of judgments.

1. Introduction

Multi-criteria decision analysis (MCDA) is suitable for solving problems at different levels of complexity. Different sources and types of data (quantitative and qualitative; reliable and questionable; complete and incomplete, etc.) and the presence of conflicting elements (primarily criteria) influence the process of finding the final solution – decision. Multi-criteria decision-making methods (MCDMs) enable MCDA as a part of it and are broadly in use in diverse fields of science and practice. They combine knowledge and techniques of systems analysis, mathematics, computer science, social science, psychology, and many other disciplines in the complex world we live in. Most methods are considered heuristic because the rules they are based on are continuously subject to controversies among researchers. Namely, different methods may produce different results because of embedded different rules. Although the rules are usually clearly stated or intuitively respected there is not any proof that their application will lead to trustworthy output reflecting the judgment of decision-makers. MCDM methods perform differently in different environments such as individual and group context, availability or reliability of the information, quality and quantity of data, previous knowledge, expertise, and/or experience of the decision maker(s); and so forth.

As recently reported in [1] 'there are over a hundred MCDM methods, all aimed to simplify the problems and facilitate achieving optimal solutions'. The existence of a large number of methods may be seen as a strong point but also as a weakness because there is no absolute truth in any claim that a certain method is better than another for solving a given multi-criteria problem [2]. In several studies [3], [4], [5], [6], and [7]

[1] Corresponding author

there are comparisons between different methods regarding problem structures, consistency, the accuracy of final results, ability to provide accurate representations of preferences of the decision-makers and the ability to comprehend the uncertainty, effects of aggregation and normalization method, etc. The studies [8] and [9] summarized the advantages, disadvantages, and areas of applications of numerous MCDM methods. Based on a review of psycho-cognitive literature and comparative study of different MCDA methods, in [2] are presented framework guidelines for selecting the method. [10] proposed a taxonomy for making the selection based on the problem type.

One of the most used general MCDM methods is the Analytic hierarchy process (AHP) developed by [11]. An analysis of collaboration evaluation in AHP research from 1982 to 2018 is presented in [12], where also some other surveys of the method and its wide applications can be found. Following the standard model, many different versions of AHP are developed in the next decades [13] but the axioms it follows have not been changed since it has been introduced in 1980. No significant changes in the methodology of its application happened meanwhile regarding hierarchization of the decision problem, preference measuring scales, manner of performing comparisons at all levels of hierarchy, and synthesis method to derive the final solution. Interpretation of output represented as cardinal information about priorities of decision elements at the bottom of the hierarchy (alternatives) versus heading element at the top of the hierarchy (goal) has not been changed, too. The determination of elements of the hierarchy is relatively objective and may depend on the degree of decision-making intervention (local, national or regional level). In most cases, the selected elements and factors of the decision matrix do not cover all possible criteria or all possible alternatives that may exist for only one goal. As it is reported in [14] 'one of the most prominent features of AHP methodology is to evaluate quantitative as well as qualitative criteria and alternatives on the same preference scale' so it is true to say that decision-makers need to be offered relatively simple methods to express their thoughts and preferences. Subjective decision makers' opinions can be interpreted by choosing a number from the numerical scale with predetermined appropriate semantic meaning [15].

Note that there are many modifications, versions, and optional uses of the AHP, which will not be discussed here, such as (1) Fuzzy AHP; (2) multiplicative AHP; (3) interval comparisons; (4) semi-empty comparison matrices; (5) individual versus group applications; or (6) hesitant AHP (AHP-H). The latest is proposed by [16] and it offers the decision-maker to give more than one opinion while making some judgments (pairwise comparisons). Instead of direct numerical expressions of judgments, linguistic preference relations are also in use. Discussion on the advantages and drawbacks of listed approaches, as well as on consistency measures and aggregation techniques in group decision-making frameworks, are out of scope in this work. Detail description of mentioned methods and techniques within 'the AHP hemisphere' can be found in rich literature and overviews in the subject area (e.g. [1], [17], [18], [19], [20], [21] [22], [23]).

Following the discussion on possible drawbacks in the AHP method, recall that the only significant modification of the original model has been the introduction of an ideal mode of synthesis to resolve the problem of re-ordering original alternatives if one original alternative is copied and added to the original alternative set. The other modifications in the base model, which will be labeled from now on as standard AHP

(AHP-S) does not exist. Rather, there are some additions in the methodology of AHP application, two of them being of interest here:

• Limited AHP (AHP-L), proposed by [24], for improving consistency of the decision-maker while using AHP-S; and

• Best-worst method (BWM), developed by Rezaei [25], [26] for reducing the number of multiplicative preference relations.

These two modifications tackle specific parts of the standard AHP method and do not change the structure or philosophy of the base method. They offer a relaxed environment to the decision-makers while they judge decision elements. Characteristics of AHP-L and BWM will be presented in the next section after the main characteristics of AHP-S are briefly described as preliminary information about the 'comparisons framework' created to analyze the application and output of all three methods for matrices of different sizes. Note that BWM is frequently treated as a different method from AHP, but in our opinion, it is only a very good modification of the judicial process and efficient addition to the original AHP methodology.

In this study, the authors firstly extracted three criteria sets consisting of six, seven, and eight criteria from three different AHP applications in water resources, agriculture, and environmental management. Weights of criteria computed in reported studies are copied here and used for comparisons with the results of the other two aforementioned methods.

The procedure labeled as Limited AHP (AHP-L) is performed by emptying three matrices and leaving in place only comparisons in the upper triangle next to the main diagonal. By strictly following the transition rule, remaining entries of matrices are filled with generated values and thus completely consistent matrices are created. Application of the eigenvector method on these matrices produced sets of criteria weights.

The third applied method is the best-worst method (BWM) which is methodologically different from AHP in part of optimizing weights for all sets of criteria by using a lower number of judgments than AHP-S.

The principal aim of the analysis of the results of the three methods is to demonstrate the sensitivity of solutions if different information (set of judgments) is available and to enable discussion of their opportunities in real-life multi-criteria decision making and AHP implementations.

The paper is organized in the following way. Section 2 provides a brief description and preliminary knowledge about the methods and approaches used in this study. Section 3 presents the results of the study. Conclusions are given in the final Section 4 followed by selected references.

2. Methods

2.1. Standard AHP (AHP-S)

The core of the original AHP [11], usually considered as the standard version of this method, lies in presenting the problem as a hierarchy and comparing the hierarchical

elements in a pairwise manner by using Saaty's 9-point scale (Table 1) to express the importance of one element over another, in regards to the element in the higher level.

If n elements of one level of the hierarchy are compared regarding the element in the upper level, a comparison matrix (labeled also as multiplicative preference relation, MPR) has the following quadratic form:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}. \quad \textbf{(1)}$$

Each matrix element a_{ij} is a subjective judgment provided by the decision-maker of the mutual importance of the two elements, i and j. If the decision-maker is fully consistent, then the transitive rule $a_{ij}a_{jk} = a_{ik}$ should apply for all i, j, k in range 1 to n.

Table 1. Saaty's importance scale

Definition	Assigned value
Equally important	1
Weak importance	3
Strong importance	5
Demonstrated importance	7
Absolute importance	9
Intermediate values	2,4,6,8

Under perfect consistency, a_{ij} is equal to:

$$a_{ij} = w_i / w_j \quad \textbf{(2)}$$

where w_i and w_j are the local weights of elements i and j regarding the element in the upper level. So, the weight's vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$, which corresponds to the matrix (1), comprises the local weights of all the elements in the given hierarchy level regarding the element in the upper level.

However, vector $\boldsymbol{w}$ is unknown, and the problem is that there is no such unique vector because of the well-known inconsistencies of the decision-maker or the limitations imposed by Saaty's (or any other) scale. To measure the quality of the $\boldsymbol{w}$ vector, computed by any of the existing methods (e.g. [27], [1]), one can define several metrics and compare the original matrix A and corresponding matrix C:

$$C = \begin{bmatrix} w_1 / w_1 & w_1 / w_2 & \ldots & w_1 / w_n \\ w_2 / w_1 & w_2 / w_2 & \ldots & w_2 / w_n \\ \ldots & \ldots & \ldots & \ldots \\ w_n / w_1 & w_n / w_2 & \ldots & w_n / w_n \end{bmatrix}. \quad \textbf{(3)}$$

A large number of elements in the hierarchy may not affect the level of congruence between matrices *A* and *C* due to the scale imperfections, insufficient knowledge about the decision problem of the decision-maker, etc.

The differences between corresponding elements of the matrices (1) and (3) are usually treated as inconsistencies of the decision-maker. In [28] and [11] is recommended the Consistency Ratio (*CR*) as a measure of individual inconsistency and it is considered as a part of the standard version of the method. The defined threshold value of 0.1 (less than this is considered as consistent) is arbitrary.

2.2. Limited AHP (AHP-L)

The standard analytic hierarchy process (AHP-S) evaluates decision criteria and alternatives by setting all multiplicative preference relations between decision elements at all levels of the hierarchy, then calculates local weights of criteria vs. goal, alternatives vs. each criterion, and finally synthesize local weights to calculate the final weights of alternatives vs. goal. The weights are plausible if the comparison matrices are consistent or near consistent [29], [30], [24], [31]. Consistency and ways to measure it are a subject of many controversies among researchers regarding their definition, interpretation, and usage. In standard AHP, full consistency is achieved only if the elements a_{ij} of given local comparison matrix *A* satisfy transitivity and reciprocal rule given as $a_{ij}=a_{ik}\cdot a_{kj}$ and $a_{ji}=a_{ij}^{-1}$ for all *i*, *j* and *k*, all ranging from 1 to *n*, where *n* is the size of a matrix *A*. For high-order matrices, say five and higher, the first rule is very difficult to reach and the inconsistency measurement is advisable. A good review of inconsistency measures is given in [32], while many aspects of their use can be found in rich literature (e.g. [33], [27], [34], [35], [36], [37]).

In [24] it is presented an expert module, implemented in Visual Prolog to assist the user in the construction of a consistent or near consistent matrix. The module is aimed to help the decision-maker to intervene after each comparison if inconsistency appears in his/her judgment. The user is guided to improve consistency through a sequence of four steps. The leading idea in this procedure is to use only *n*-1 judgments of decision-maker elicited at matrix diagonal adjacent to the principal diagonal where values '1' are posted indicating all comparisons of decision elements with itself. Notice that the lower triangle of the matrix (as also in standard AHP) contains reciprocals of numbers in the upper triangle, symmetric to the main diagonal.

To illustrate the procedure of obtaining a fully consistent matrix of size 5, suppose that decision-maker compared by importance paired elements *C*1 with *C*2, *C*2 with *C*3, *C*3 with *C*4, and *C*4 with *C*5 (Figure 1). Values highlighted in green are from the Saaty's scale (1/9, 1/8, ..., 1/2, 1, 2, ..., 8, 9) offered to be used by the decision-maker.

	C_1	C_2	C_3	C_4	C_5
C_1	1	1/2			
C_2		1	1/4		
C_3			1	3	
C_4				1	1
C_5					1

Fig. 1. Initial matrix required for the limited pairwise comparison method

By using numerical values (green highlighted numbers) from a diagonal next to principal diagonal, calculation of elements in the upper triangle of matrix *A* can be performed by applying the transitive rule as follows:

Row #1: $a_{13}=a_{12}\cdot a_{23}=(1/2)\cdot(1/4)=1/8$; $a_{14}= a_{12}\cdot a_{23}\cdot a_{34}=(1/2)\cdot(1/4)\cdot 3=3/8$;
$a_{15}=a_{12}\cdot a_{23}\cdot a_{34}\cdot a_{45}=(1/2)\cdot(1/4)\cdot 3\cdot 1= 3/8$
Row #2: $a_{24}=a_{23}\cdot a_{34}=(1/4)\cdot 3=3/4$; $a_{25}=a_{24}\cdot a_{45}=3/4\cdot 1=3/4$
Row #3: $a_{35}=a_{34}\cdot a_{45}=3\cdot 1=3$.

	C_1	C_2	C_3	C_4	C_5
C_1	1	1/2	1/8	3/8	3/8
C_2		1	1/4	3/4	¾
C_3			1	3	3
C_4				1	1
C_5					1

Fig. 2. Generated fully consistent matrix (AHP-L)

The resulting matrix in Fig. 2 (with reciprocals in its lower triangle, not shown) is fully consistent. Once calculated weights of elements C_1-C_5 by any of known prioritization methods (e.g. EV, AN, LLS, etc.), form priority vector of the matrix which can serve as a 'reference vector' to be targeted if any additional pairwise comparison is made by the decision-maker is available. That is, besides *n*-1 comparisons, up to (*n*-1)(*n*-2)/2 more comparisons must be added until the matrix is filled up. In the given example, to four 'green judgments' (*n*-1=5-1=4), additional 6 judgments have to add to the complete matrix as in standard AHP. The problem with this matrix is that although it is consistent, it is not real because it is not obtained by the decision-maker. Instead of *n*(*n*-1)/2=10 judgments, as in AHP-S, only half of it is real (made by the decision-maker) and the other half is generated. Besides that, a partly generated matrix is artificial, generated values may not belong to the scale used by the decision-maker; these values do not correspond to the semantics defined for values in the scale (see Table 1) and therefore are not justified. Finally, generated values may in some cases be out of scale.

2.3. Best-worst method (BWM)

Different from the basic idea of the AHP method where complete multiplicative preference relation is created by the decision-maker or analyst, the BWM executes the

reduced number of comparisons of decision elements at a given level of the problem hierarchy (usually at criteria level). Multiplicative preference relation is incomplete because only requested is the preference of the best criterion over all the criteria and the preference of all criteria over the worst criterion.

Due to efficiency in reducing the number of required comparisons, the method received attention [38] and there are a quite number of reported studies on its use since the method has been originally proposed by Rezaei [25] as a non-linear model (4).

$$\begin{gathered}\min \max_{\mathrm{j}} \left\{\left|w_B/w_j - a_{Bj}\right|, \left|w_j/w_W - a_{jW}\right|\right\} \\ \text{s.t} \\ \sum_j w_j = 1, \quad \text{for all } j \\ w_j \geq 0 \quad \text{for all } j\end{gathered} \tag{4}$$

Following the ideas introduced in [33], [26] presented the linear model in which instead of minimizing the maximum value among the set of $\left\{\left|w_B/w_j - a_{Bj}\right|, \left|w_j/w_W - a_{jW}\right|\right\}$ minimization is performed over the maximums among the set of $\left\{\left|w_B - a_{Bj}\mathrm{w_B}\right|, \left|w_j - a_{jW}w_W\right|\right\}$. The problem is now:

$$\begin{gathered}\min \max_{\mathrm{j}} \left\{\left|w_B - a_{Bj}w_B\right|, \left|w_j - a_{jW}w_W\right|\right\} \\ \textbf{s.t.} \\ \sum_j w_j = 1, \quad \text{for all } j \\ w_j \geq 0 \quad \text{for all } j\end{gathered} \tag{5}$$

and corresponding linear version of the model (5) can be defined as the model (6).

$$\begin{gathered}\min \ \varepsilon \\ \textbf{s.t.} \\ \left|w_B - a_{Bj}w_{\mathrm{B}}\right| \leq \varepsilon, \ \text{for all } j \\ \left|w_j - a_{jW}w_W\right| \leq \varepsilon, \quad \text{for all } j \\ \sum_j w_j = 1, \quad \text{for all } j \\ w_j \geq 0 \quad \text{for all } j\end{gathered} \tag{6}$$

In models (5) and (6) 'for all *j*' means 'for all compared elements' in the set of decision elements, either criteria, sub-criteria, or alternatives; for instance, if there are *n* criteria, then 'for all *j*' means $j = 1, 2, ..., n$.

Differently from the non-linear model (5) which may have multiple solutions, linear model (6) produces a unique solution: the optimal set of weights w_j^* for all *j* , and ε^*. Similar to Mikhailov's model and his 'natural measure of consistency' μ^* [33] in the model (6) ε^*, with a value between 0 and 1, can be considered as an indicator of consistency demonstrated by the decision-maker. Obviously, the lower the value of this indicator, the higher the level of consistency.

So far, there is no clear suggestion of what would be tolerance limit regarding the consistency if the linear version of BWM is applied. Reported researches are focused on searching for the new consistency ratio to replace the original consistency ratio *ε.* One

of the ongoing researches [39] is aimed at how to achieve a reflection of the consistency status of the input judgment instead of output results.

The number of multiplicative preference relations in BWM is lower than in AHP-S. In the case of BWM, there are n-2 Best-to-Others comparisons, plus n-2 Others-to-Worst comparisons, and one more Best-to-Worst comparison [25]. This gives a total of $2n$-3 comparisons, which is a lower value than $n(n\text{-}1)/2$ as in AHP-S. The difference between the numbers of required comparisons in the two methods rises with the size of the matrix. For instance, if the matrix size is 7, AHP-S requires 21 comparisons, whether this number in the case of BWM is only 11.

A review of the most recent studies in the field indicates an extension of the BWM method with other techniques such as those belonging to fuzzy sets theory. For instance, BWM is used to derive the priorities from hesitant fuzzy preferences in uncertain situations such as its integration in a fuzzy environment [40], by using hesitant numbers [41], [42], or grey numbers [43], [44].

3. Numerical Examples

3.1. Brief Introduction to the Examples

Several matrices are used to illustrate differences that appear if AHP-S, AHP-L, and BWM are used to determine the weights of decision elements. Matrices of sizes 6, 7, and 8 are taken from papers of the author team as published in peer-reviewed national and international journals. These matrices are created during real-life applications of the AHP-S method and then re-considered by the methods AHP-L and BWM. The following assumptions are adopted to unify conditions for comparing the results:

1. In all cases, Saaty's 9-point ration scale is used.
2. To apply the limited version of AHP, an upper triangle of each original matrix (from standard AHP application) is emptied except entries adjacent to the main diagonal. This way, only part of original judgments is left at the original place to enable generating remaining entries which will make such a new matrix fully consistent.
3. Prioritization method in AHP-S and AHP-L is performed by the eigenvector method.
4. BWM rating of remaining decision elements versus best and worst is performed following preferences obtained in the original AHP-S application.

Note that in real-life application decision-maker would be asked to fill diagonal in an empty matrix (if AHP-L is selected), to fill the upper triangle of the matrix (AHP-S), or to select the best and worst criterion and compare other criteria vs best and worst one (BWM). So, in each of the cases, the decision maker would express her/his judgments directly.

3.2. Example #1 – Matrix size 6

The source paper [45] presents an evaluation methodology based on five different prioritization methods within the AHP-S framework. The methodology is aimed at identification of the best among four alternative dispositions of pumping stations within the canal network of the Danube-Tisza-Danube system in northern Serbia. The selection process is based on six economic and technological criteria as follows: C_1 – economic parameters, C_2 – reliability of operation, C_3 – efficiency during flood events, C_4 – conformity with water supply operational schemes and rules, C_5 – adaptability to other infrastructural staging processes and C_6 – technical controllability.

The results obtained by three methods are as follows:

Standard AHP

The original matrix created by the decision-maker and weights of criteria derived by the *EV* method is presented in Table 2. The consistency ratio *CR* for this matrix is 0.089 which is less than 0.1 considered as the maximum permitted value to declare that the derived vector of weights is consistent and can be declared as a decision.

Table 2. AHP-S comparison matrix and weights of criteria (*Example #1*)

Criteria	C_1	C_2	C_3	C_4	C_5	C_6	Weight	Rank
C_1	**1**	1/5	1/3	1/3	1/5	1/5	0.042	6
C_2		**1**	3	2	3	3	0.341	1
C_3			**1**	2	1	1/2	0.146	4
C_4				**1**	1	3	0.170	2
C_5					**1**	2	0.164	3
C_6						**1**	0.136	5

Consistency measure: CR = 0.089 (<0.100; satisfactory)

Limited AHP

If all entries of the original matrix (Table 2) are deleted and only elements in the diagonal adjacent to the main diagonal are left (see green-highlighted numbers), then by application of the transition rule all 'emptied' entries are generated and presented in Table 3.

Table 3. AHP-L comparison matrix and derived weights of criteria (*Example #1*)

Criteria	C_1	C_2	C_3	C_4	C_5	C_6	Weight	Rank
C_1	**1**	1/5	3/5	6/5	6/5	12/5	0.103	3
C_2		**1**	3	6	6	12	0.513	1
C_3			**1**	2	2	4	0.171	2
C_4				**1**	1	2	0.085	4-5
C_5					**1**	2	0.085	4-5
C_6						**1**	0.043	6

A procedure for generating missing entries in the matrix of AHP-L was as follows:

Row #1: $a_{13}= a_{12}\cdot a_{23}=1/5\cdot3=3/5$; $a_{14}= a_{13}\cdot a_{34}=3/5\cdot2=6/5$; $a_{15}= a_{14}\cdot a_{45}=6/5\cdot1=6/5$; $a_{16}=a_{15}\cdot a_{56}=6/5\cdot2=12/5$

Row #2: $a_{24}=a_{23}\cdot a_{34}=3\cdot2=6$; $a_{25}=a_{24}\cdot a_{45}=6\cdot1=6$; $a_{26}=a_{25}\cdot a_{56}=6\cdot2=12$

Row #3: $a_{35}=a_{34}\cdot a_{45}=2\cdot1=2$; $a_{36}=a_{35}\cdot a_{56}=2\cdot2=4$

Row #4: $a_{46}=a_{45}\cdot a_{56}=1\cdot2=2$.

The partly generated matrix in Table 3 is fully consistent. However, it contains entries that do not belong to the 9-point scale and therefore do not have the exact linguistic (semantic) meaning given in Table 1. For instance, the generated value of element a_{16} is 12/5=2.4 which determines preferences among criteria *C*1 and *C*6 as between scale value 2 with semantic meaning '*C*1 is slightly more important than *C*6', and scale value 3 with semantic meaning '*C*1 is more important than *C*6'. It is difficult to differentiate preference between two criteria and to give a logical explanation for 'fine tuning' of 0.4 between 2 and 3.

The other point is that generated entry value may be outside the range of scale (1/9, 1/8, ..., 1, 2, ...9). For instance, generated entry a_{26} is 12, a value higher than 9 which stands for absolute preference of one element over the other. Note also that value 12 significantly differs from value 3 (*C*2 is strongly more important than *C*6), originally inserted by the decision-maker while using AHP-S.

Needless to say, is that in this case, the consistency ratio CR is zero as a consequence of the full implementation of transition rules.

BWM

If pairwise comparison vectors for the best criterion *C*2 and worst criterion *C*6 are defined as presented in Table 4 (by using as a 'direction of behavior' original preferences from AHP-S application), then the LP problem for this setting is:

$$\begin{gathered}\min\ \varepsilon \\ \textbf{s.t.} \\ |w_2 - 7w_1| \le \varepsilon \\ |w_2 - 6w_3| \le \varepsilon \\ |w_2 - 2w_4| \le \varepsilon \\ |w_2 - 4w_5| \le \varepsilon \\ |w_2 - 8w_6| \le \varepsilon \\ |w_1 - 2w_6| \le \varepsilon \\ |w_2 - 9w_6| \le \varepsilon \\ |w_3 - 5w_6| \le \varepsilon \\ |w_4 - 6w_6| \le \varepsilon \\ |w_5 - 4w_6| \le \varepsilon \\ w_1 + w_2 + w_3 + w_4 + w_5 + w_6 = 1. \\ w_1, w_2, w_3, w_4, w_5, w_6 \ge 0\end{gathered} \tag{7}$$

The solution to model (7) is as presented in the research paper [46] and reproduced in Table 5: $w_1 = 0{,}07402330$; $w_2 = 0{,}4126114$; $w_3 = 0{,}08636052$; $w_4 = 0{,}2590816$; $w_5 = 0{,}1295408$; $w_6 = 0{,}03838245$; $\varepsilon = 0{,}1055517$.

Table 4. BWM vectors for best and worst criterion (*Example #1*)

Criteria		C1	C2	C3	C4	C5	C6
Best criterion:	**C2**	7	1	6	2	4	8
Worst criterion:	**C6**	2	9	5	6	4	1

Table 5. BWM solution (*Example #1*)

Criteria	Weight	Rank
C1	0.074	5
C2	0.413	1
C3	0.086	4
C4	0.259	2
C5	0.129	3
C6	0.038	6

Value $\varepsilon = 0.106$ can be considered as satisfactory consistency of the process. Recall that the lower value (closer to zero) ε means better consistency.

Altogether (AHP-S, AHP-L, BWM)

Table 6 summarizes the results obtained by three methods.

Table 6. Summary of AHP-S, AHP-L, and BWM solutions (Example #1)

Criteria	Standard AHP (AHP-S)		Limited AHP (AHP-L)		Best-Worst Method (BWM)	
	Weight	Rank	Weight	Rank	Weight	Rank
C1	0.044	6	0.103	3	0.074	5
C2	0.340	1	0.513	1	0.413	1
C3	0.148	4	0.171	2	0.086	4
C4	0.166	2	0.085	4-5	0.259	2
C5	0.164	3	0.085	4-5	0.129	3
C6	0.139	5	0.043	6	0.038	6

Good agreement in the final ranking of criteria is between methods AHP-S and BWM, except on the two lowest positions. Recall that in the case of AHP-S there were $n(n\text{-}1)/2 = (6 \cdot 5)/2 = 15$ multiplicative preference relations, while in the case of BWM this number was $2n\text{-}3 = 2 \cdot 6\text{-}3 = 9$. The reduction of the number of comparisons by 40% (9/15) is significant and the result is not that much different, at least regarding the ranking of criteria at the first four positions. Weights of the top-ranked criterion C1 differ by 30% and for the second-ranked criterion C4 the difference is 56%; in both cases, higher values are obtained by the BWM. Weights for the third-ranked criterion C5 differ 27% and for the fourth-ranked criterion C3 weights differ by 82%; in the case of these two criteria higher values are obtained by the AHP-S. In AHP-L ranking of criteria is very different from the other two methods, except in the case of the top-ranked criterion which is also C2. It's very high weight is obviously due to high values

of generated judgments a_{21}=5, a_{24}=6, a_{25}=6, and a_{26}=12, the last one even out of 9-point scale on its upper bound.

Remark #1.1. Applied three methods expectedly produced different weights of criteria which are following differences in methods. Although in all cases the same 9-point scale is used, the other prepositions are not followed in the same way. For instance, judgments in AHP-S and BWM are elicited from real decision-makers in two separate sessions, the second one (BWM) is undertaken several years after the first one (AHP-S) described in the source paper.

Remark #1.2. Because the evaluation of criteria is essential in any decision-making process, different weights obtained by different methods may significantly moderate the next steps of the evaluation process, such as the synthesis of local weights of alternatives by multiplication with the weights of criteria.

Remark #1.3. The number of real judgments in AHP-S is 15, in AHP-L only 5, and in BWM 9. The difference in information base used (number of real pairwise comparisons performed) for deriving weights is significant and richness of information is obviously on side of the first and third methods. In the case of AHP-L, it would be necessary to adjust the judgment of the decision-maker, for instance in an iterative way by the expert module proposed in [24]. However, this procedure we consider inappropriate, at least in the context we are analyzing here.

3.3. Example #2 - Matrix Size 7

In this source paper [47], a multi-criteria evaluation procedure is proposed for ranking five walnut cultivars selected in Serbia, Bulgaria, and France. Supported by AHP-S, the ranking process is settled with principal regard to nut characteristics of fruits. Eight criteria of different metrics are used for shaping and filtering two experts' preferences of criteria and cultivars. Here we elaborate the multiplicative preference relations contained in the matrix for criteria as obtained by one of the experts, a professor in fruit production and recognized national expert in walnut's and hazelnut's selection standards. The set of criteria was as follows: C1 – kernel's color, C2 – kernel's portion, C3 – nut's weight; C4 – the taste of the kernel, C5 – shell, C6 – storage, and C7 – trade value.

Standard AHP

Table 7. AHP-S comparison matrix and derived criteria weights (*Example #2*)

Criteria	C1	C2	C3	C4	C5	C6	C7	Weight	Rank
C1	**1**	3	1	7	5	9	7	0.321	1-2
C2		**1**	1/3	5	5	7	5	0.180	3
C3			**1**	7	5	9	7	0.321	1-2
C4				**1**	1	3	3	0.059	4
C5					**1**	3	1	0.053	5
C6						**1**	1/3	0.023	7
C7							**1**	0.042	6

Consistency measure: CR = 0.047 (<0.100; satisfactory)

The result obtained by the eigenvector prioritization method during standard AHP application is presented in Table 7. Consistency measure *CR* is at a very low value (CR=0.047), far below limit *CR*=0.01. Therefore, the vector of criteria weights can be considered as consistently derived.

Limited AHP

The results obtained by the eigenvector prioritization method during AHP-L application (with original judgments highlighted in green, and remaining judgments generated by the application of transition rule), are presented in Table 8.

Table 8. AHP-L comparison matrix and derived criteria weights (Example #2)

Criteria	C1	C2	C3	C4	C5	C6	C7	Weight	Rank
C1	**1**	3	1	7	7	21	7	0.356	1-2-3
C2		**1**	1/3	7/3	7/3	7	7/3	0.119	4
C3			**1**	7	7	21	7	0.356	1-2-3
C4				**1**	1	3	1	0.051	5-6
C5					**1**	3	1	0.051	5-6
C6						**1**	1/3	0.017	7
C7							**1**	0.356	1-2-3

Row #1: $a_{13}=a_{12}\cdot a_{23}=3\cdot 1/3=1$; $a_{14}=a_{13}\cdot a_{34}=1\cdot 7=7$; $a_{15}=a_{14}\cdot a_{45}=7\cdot 1=7$; $a_{16}=a_{15}\cdot a_{56}=7\cdot 3=21$; $a_{17}=a_{16}\cdot a_{67}=21\cdot 1/3=7$

Row #2: $a_{24}=a_{23}\cdot a_{34}=1/3\cdot 7=7/3$; $a_{25}=a_{24}\cdot a_{45}=7/3\cdot 1=7/3$; $a_{26}=a_{25}\cdot a_{56}=7/3\cdot 3=7$ $a_{27}=a_{26}\cdot a_{67}=7\cdot 1/3=7/3$

Row #3: $a_{35}=a_{34}\cdot a_{45}=7\cdot 1=7$; $a_{36}=a_{35}\cdot a_{56}=7\cdot 3=21$; $a_{37}=a_{36}\cdot a_{67}=21\cdot 1/3=7$

Row #4: $a_{46}=a_{45}\cdot a_{56}=1\cdot 3=3$; $a_{47}=a_{46}\cdot a_{67}=3\cdot 1/3=1$

Row #5: $a_{57}=a_{56}\cdot a_{67}=3\cdot 1/3=1$

Like in the other examples, consistency measure *CR*, in this case, is also zero due to the implementation of transition rules.

BWM

Pairwise comparison vectors are defined as in Table 9.

Table 9. BWM vectors for best and worst criterion (Example #2)

Criteria		C1	C2	C3	C4	C5	C6	C7
Best criterion*:	**C1**	1	3	1	7	5	9	7
Worst criterion:	**C6**	9	4	8	3	2	1	2

***Best** could also be **C3**

The corresponding optimization model is:

$$\begin{gathered}\min\ \varepsilon \\ \textbf{s.t.} \\ |w_1 - 3w_2| \le \varepsilon \\ |w_1 - 1w_3| \le \varepsilon \\ |w_1 - 7w_4| \le \varepsilon \\ |w_1 - 5w_5| \le \varepsilon \\ |w_1 - 9w_6| \le \varepsilon \\ |w_1 - 7w_7| \le \varepsilon \\ |w_2 - 4w_6| \le \varepsilon \\ |w_3 - 8w_6| \le \varepsilon \\ |w_4 - 3w_6| \le \varepsilon \\ |w_5 - 2w_6| \le \varepsilon \\ |w_7 - 2w_6| \le \varepsilon \\ w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 = 1 \\ w_1, w_2, w_3, w_{4,} w_5, w_6, w_7 \ge 0\end{gathered} \tag{8}$$

The solution to this model is (Table 10): $w_1 = 0.303761;\ w_2 = 0.123754; w_3 = 0.371263;\ w_4 = 0.053038;\ w_5 = 0.074253\ ;\ w_6 = 0.040180; w_7 = 0.033751;\ \varepsilon = 0.067502$. Value ε = 0.068 is close to zero and the solution can be considered as satisfactory like in Example #1.

Table 10. BWM solution (Example #2)

Criteria	Weight	Rank
C1	0.304	2
C2	0.124	3
C3	0.371	1
C4	0.053	5
C5	0.074	4
C6	0.040	6
C7	0.034	7

Altogether (AHP-S, AHP-L, BWM)

Table 11 summarizes the result obtained by the three methods.

All methods are in relatively good agreement regarding the final ranking of criteria, especially on the top three positions. The sum of weights of the top three criteria *C*1, *C*2, and *C*3 is 0.822 in AHP-S, 0.831 in AHP-L, and in BWM this sum is 0.799. The difference in these sums between methods of up to 4% is not significant. The sum of weights of the remaining four criteria in all cases is small and the final AHP synthesis across all criteria should result in approximately the same order of walnut cultivars if weights obtained by the AHP-L or the BW methods are used instead of weights obtained by AHP-S.

The results are interesting, especially from the standpoint of information available. In this example, AHP-S required $n(n\text{-}1)/2=(7\cdot6)/2=21$ comparisons, whether in BWM this number was $2n\text{-}3=2\cdot7\text{-}3=11$; in case of AHP-L, only 7 comparisons were required, taken from the original matrix (highlighted entries above the main diagonal in Table 8).

Reduction in the number of comparisons in AHP-L and BWM concerning AHP-S by 48% (11 vs. 21) and 67% (7 vs. 21) respectively, can be considered very significant while the result is not that much different regarding the ranking of decision elements.

Table 11. Summary of AHP-S, AHP-L, and BWM solutions (Example #2)

Criteria	**Standard AHP (AHP-S)**		**Limited AHP (AHP-L)**		**Best-Worst Method (BWM)**	
	Weight	Rank	Weight	Rank	Weight	Rank
C1	0.321	1-2	0.356	1-2	0.304	2
C2	0.180	3	0.119	3	0.124	3
C3	0.321	1-2	0.356	1-2	0.371	1
C4	0.059	4	0.051	4-5-6	0.053	5
C5	0.053	5	0.051	4-5-6	0.074	4
C6	0.023	7	0.017	7	0.040	6
C7	0.042	6	0.051	4-5-6	0.034	7

Like in the previous example, applied methods produced different weights of criteria. This is a consequence of prevailing preferences given to criteria $C1$, $C2$, and $C3$ by the decision-maker. In the case of criteria $C1$ and $C3$, AHP-L in artificially achieving full consistency of the pairwise comparison matrix (by following transition rule) gives two times value 21 which significantly surpasses the highest value from the ration scale, that is 9 – defined for absolute preference.

Remark #2.1: This example indicates possible insensitivity of ranking produced by fairly different methods. However, this is not the case if cardinal information is analyzed. The weights differ which is a clear consequence of the fact that in the case of AHP-S (real-life decision-making) the inconsistency index is CR=0.06 which is, although lower than tolerant value 0.10, still much higher than CR=0 in the case of AHP-L which generates fully consistent (but partly artificial) pairwise comparison matrix; the expression 'partly' relates to the fact that 6 comparisons above the main diagonal are real. Changes that might be produced if adjustments are implemented by the algorithm presented in [24] could make the comparison matrix more realistic, e.g. with entries corresponding to exact values from the ratio scale – to have semantic meanings. Adjustments are inherently unrealistic because there are many controversies regarding the issue of consistency, the readiness of decision-maker to accept corrections in his/her original judgments (as made in AHP-S), etc.

Remark #2.2: At this point worth commenting is the minimum error ε = 0.0676502 generated as the optimal value of the goal function by the LP program in the BWM. The structure of the LP program is such that ε is a unique tolerant difference between linear relations of all weights and their corresponding judgments in the comparison matrix. The optimal value ε in this example is not that high regarding size 7 of the matrix, and the set of weights can be considered as a sufficiently consistent solution. A conclusion might be that a low value of error ε may justify the decision to use BWM instead of AHP-S, or at least to either alternatively use the results of BWM only (once both methods are applied by the same decision-maker) or to combine the results from both

methods, for instance by their geometric aggregation. This approach might be interesting especially for matrices of higher order.

3.4. Example #3 - Matrix size 8

Source papers: (a) [48] and (b) [49].

The third example is taken from two most recently published papers related to group evaluation of the following set of eight criteria applicable in ranking by importance protected natural areas in Serbia, recognized as Ramsar areas [50]: C1 – protection of habitats, C2 – biodiversity, C3 – extreme water regime, C4 – purposes, C5 – location, C6 – tourism and education; C7 – water quality, and C8 – socio-cultural heritage.

The results obtained by the three methods are presented in Tables 12-15.

Standard AHP

Table 12. AHP-S comparison matrix and derived criteria weights (Example #3)

Criteria	C1	C2	C3	C4	C5	C6	C7	C8	Weight	Rank
C1	**1**	1/3	1/4	5	8	7	1/4	8	0.135	4
C2		**1**	1/2	5	8	7	1/2	8	0.200	3
C3			**1**	5	9	7	1	9	0.267	1-2
C4				**1**	5	3	1/5	5	0.060	5
C5					**1**	1/5	1/9	1	0.018	7-8
C6						**1**	1/7	5	0.041	6
C7							**1**	9	0.267	1-2
C8								**1**	0.018	7-8

Consistency measure: CR = 0.088 (<0.100; satisfactory)

Limited AHP

Table 13. AHP-L comparison matrix and derived criteria weights (Example #3)

Criteria	C1	C2	C3	C4	C5	C6	C7	C8	Weight	Rank
C1	**1**	1/3	1/6	5/6	25/6	25/30	25/210	225/210	0.046	6
C2		**1**	1/2	5/2	25/2	25/10	25/70	225/70	0.137	3
C3			**1**	5	25	5	5/7	45/7	0.273	2
C4				**1**	5	1	1/7	9/7	0.055	4-5
C5					**1**	1/5	1/35	9/35	0.011	8
C6						**1**	1/7	9/7	0.055	4-5
C7							**1**	9	0.382	1
C8								**1**	0.042	7

Row #1: $a_{13}=a_{12}\cdot a_{23}=1/3\cdot 1/2=1/6$; $a_{14}=a_{13}\cdot a_{34}=1/6\cdot 5=5/6$; $a_{15}=a_{14}\cdot a_{45}=5/6\cdot 5=25/6$
$a_{16}=a_{15}\cdot a_{56}=25/6\cdot 1/5=25/30$; $a_{17}=a_{16}\cdot a_{67}=25/30\cdot 1/7=25/210$;
$a_{18}=a_{17}\cdot a_{78}=25/210\cdot 9=225/210$

Row #2: $a_{24}=a_{23}\cdot a_{34}=1/2\cdot 5=5/2$; $a_{25}=a_{24}\cdot a_{45}=5/2\cdot 5=25/2$; $a_{26}=a_{25}\cdot a_{56}=25/2\cdot 1/5=25/10$; $a_{27}=a_{26}\cdot a_{67}=25/10\cdot 1/7=25/70$;
$a_{28}=a_{27}\cdot a_{78}=25/70\cdot 9=225/70$

Row #3: $a_{35}=a_{34}\cdot a_{45}=5\cdot 5=25$; $a_{36}=a_{35}\cdot a_{56}=25\cdot 1/5=5$; $a_{37}=a_{36}\cdot a_{67}=5\cdot 1/7=5/7$
$a_{38}=a_{37}\cdot a_{78}=5/7\cdot 9=45/7$;
Row #4: $a_{46}=a_{45}\cdot a_{56}=5\cdot 1/5=1$; $a_{47}=a_{46}\cdot a_{67}=1\cdot 1/7=1/7$; $a_{48}=a_{47}\cdot a_{78}=1/7\cdot 9=9/7$

Row #5: $a_{57}=a_{56}\cdot a_{67}=1/5\cdot 1/7=1/35$; $a_{58}=a_{57}\cdot a_{78}=1/35\cdot 9=9/35$
Row #6: $a_{68}=a_{67}\cdot a_{78}=1/7\cdot 9=9/7$

BWM

Pairwise comparison vectors are defined as in Table 14.

Table 14. BWM vectors for best and worst criterion (Example #3)

Criteria	C1	C2	C3	C4	C5	C6	C7	C8
Best criterion: **C7**	4	4	2	6	8	7	1	7
Worst criterion: **C5**	5	6	7	4	1	4	8	3

The LP problem for this setting is:

$$\begin{gathered}\min\ \varepsilon \\ \text{s.t.} \\ |w_7-4w_1|\le\varepsilon \\ |w_7-4w_2|\le\varepsilon \\ |w_7-2w_3|\le\varepsilon \\ |w_7-6w_4|\le\varepsilon \\ |w_7-8w_5|\le\varepsilon \\ |w_7-7w_6|\le\varepsilon \\ |w_7-7w_8|\le\varepsilon \\ |w_1-5w_5|\le\varepsilon \\ |w_2-6w_5|\le\varepsilon \\ |w_3-7w_5|\le\varepsilon \\ |w_4-4w_5|\le\varepsilon \\ |w_6-4w_5|\le\varepsilon \\ |w_8-3w_5|\le\varepsilon \\ w_1+w_2+w_3+w_4+w_5+w_6+w_7+w_8=1 \\ w_1,w_2,w_3,w_4,w_5,w_6,w_7,w_8\ge 0\end{gathered} \tag{9}$$

and the solution is: $w_1 = 0.1074169$; $w_2 = 0.1074169$; $w_3 = 0.2148338$; $w_4 = 0.07161125$; $w_5 = 0.03222506$; $w_6 = 0.06138107$; $w_7 = 0.3437340$; $w_8 = 0.06138107$; $\varepsilon = 0.0859335$.

In this example, value ε = 0.086 is sufficiently close to zero and the solution of model (9) can be considered as satisfactorily consistent.

Altogether (AHP-S, AHP-L, BWM)

Table 15 summarizes the result obtained by three methods.

Table 15. Summary of AHP-S, AHP-L, and BWM solutions (Example #3)

Criteria	Standard AHP (AHP-S)		Limited AHP (AHP-L)		Best-Worst Method (BWM)	
	Weight	Rank	Weight	Rank	Weight	Rank
C1	0.135	4	0.046	6	0.107	3-4
C2	0.188	3	0.137	3	0.107	3-4
C3	0.262	1-2	0.273	2	0.215	2
C4	0.066	5	0.055	4-5	0.072	5
C5	0.020	7-8	0.011	8	0.032	8
C6	0.048	6	0.055	4-5	0.061	6-7
C7	0.262	1-2	0.382	1	0.344	1
C8	0.020	7-8	0.042	7	0.061	6-7

The methods are in relatively good agreement regarding the final ranking of criteria, especially on the top three positions. The sum of the weights of criteria *C*7, *C*3, and *C*2 is 0.712 obtained by AHP-S. In the case of AHP-L, the sum is 0.792, and in BWM this sum is 0.666. The weights of the remaining five criteria derived by three methods are relatively small. As in the previous example, when the AHP-S synthesis of local weights of six Ramsar areas in the northern part of Serbia (known as the Vojvodina Province) across all criteria resulted in approximately the same order of Ramsar areas when the weights obtained by the AHP-L and the BW methods are used instead. In the decision processes of selecting the most important element of the hierarchy (alternative or criterion) these methodologies are suitable for use because the application of different methods shows the same best-ranked solution.

Applied methods expectedly produced different weights of criteria as a result of the difference in methodologies used while creating a comparison matrix. Note that AHP-L and BWM give a significantly higher value than AHP-S for the first ranked criterion *C*7. This difference is almost 50% higher in the case of AHP-L versus AHP-S, and 31% in the case of BWM versus AHP-S.

In this example, AHP-S required $n(n\text{-}1)/2=(8\cdot 7)/2=28$ multiplicative preference relations, BWM required $2n\text{-}3=2\cdot 8\text{-}3=13$ relations, and in the case of AHP-L, only 8 comparisons were required. Reduction in the number of comparisons used in AHP-L and BWM versus AHP-S by 54% and 71%, respectively, is furthermore significant while the result is not that much different regarding the ranking of top positioned criteria.

In presented examples for prioritizing 6, 7, and 8 criteria number of required comparisons was different regarding the prioritization method used. In all cases AHP-L required a minimum number of comparisons, more comparisons were needed if BWM is used, and finally, the number of comparisons was largest for AHP-S. Table 16 and Fig. 3 illustrate this effect of enlarging the information base with the size of the matrix. Note that if a matrix is of size 9, the number of comparisons drops by more than a half,

actually near 60% if BWM is used instead of AHP-S. The number of comparisons does not significantly affect the final result because the best (e.g. top three) and the worst (e.g. last two) criteria will be obtained by using any method.

The number of comparisons could depend on the individual preferences of decision-makers. Depending on whether the decision-maker feels, he/she can express his/her opinion (1) comparing all the criteria with each other as is the case by using AHP-S, or (2) expressing right at the beginning which is the best and which is the worst criterion creating further opinions and comparison of other criteria according to the best/worst criterion (case of BWM).

Table 16. Relationships between methods AHP-S, AHP-L, and BWM regarding the number of comparisons required for different sizes of comparison matrices/compared elements

Size of matrix/ Number of compared elements *(n)*	Number of comparisons			R BW/AHP-S = $(4n-6)/(n^2-n)$
	AHP-S $n(n-1)/2$	AHP-L $n-1$	BWM $2n-3$	
2	1	1	1	1.00
3	3	2	3	1.00
4	6	3	5	0.83
5	10	4	7	0.70
6	15	5	9	0.60
7	21	6	11	0.52
8	28	7	13	0.46
9	36	8	15	0.42

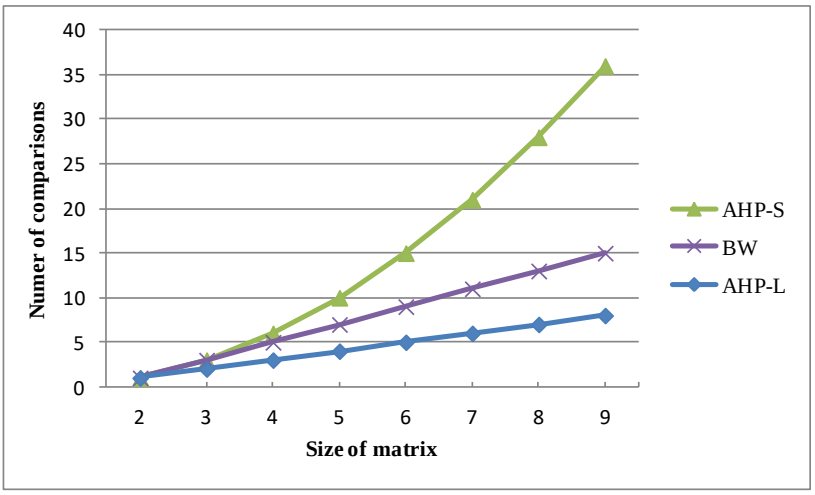


Fig. 3. The number of comparisons required by methods AHP-S. AHP-L and BWM for different sizes of comparison matrices

In the selection processes when the ranking of criteria or alternatives leads to the final choice(s) it would be desirable to choose a methodology for ranking according to the number of offered criteria and alternatives. For instance, if the size of the comparison matrix is up to 4 elements then the number of necessary comparisons is

similar regardless of which method was used (Cf. Table 16). When it comes to larger comparison matrices (the number of elements is larger than 5), it is desirable to choose a methodology that will quickly and efficiently show reliable and accurate results. Decision matrices with 8 or more criteria should be avoided in the real application of multi-criteria methods due to the proven high inconsistency of the process of comparison and evaluation of criteria. In that case, one should try to reduce the number of criteria or simplify problem hierarchies by dividing the criteria into sub-criteria (dividing one level into two or three new levels). This procedure requires a larger number of comparison matrices and thus a larger number of evaluations. In that case, the method should be chosen which requires a smaller number of comparisons.

4. Conclusions

The analytic hierarchy process (AHP) is one of the most used methods for supporting decision-making processes. It is intuitively correct because it follows the usual human intent to decompose problems into smaller parts and determine the importance of its elements by putting them into a hierarchical structure. Local evaluations are performed by pairwise comparisons of elements at a given level versus adjacent elements in the upper level. Following the prioritization process and deriving the local weights of compared elements at all levels of the hierarchy, synthesis of local weights produces the final result: priorities of decision elements (usually alternatives) at the bottom level of hierarchy regarding stated goal at the top of the hierarchy.

The most common situation is that intermediate prioritization is performed for criteria set versus goal. The final result of AHP strongly depends on priorities obtained for criteria as decision elements of prime importance in making any decision. They are usually set at the first level of the hierarchy, positioned just below the goal as a global evaluation target to be met by the alternatives at the bottom level of the hierarchy. Priorities represented by weights of criteria play the most important role in achieving the goal because they moderate the remaining synthesis process, and therefore the prioritization method to be applied at this level is essential for the complete AHP application.

While the AHP-S and AHP-L are matrix methods, the third used method in this study was BWM, a strictly linear programming (optimization) method. The methods use a different number of judgments made by the decision-maker about the mutual importance of criteria as decision elements. Because of differences in methodology and information availability (judgments used), as a consequence, all three methods produced different weights of criteria in all three studied cases with matrices of sizes 6, 7, and 8. Consistency of the prioritization process was in all analyzed cases satisfactory, of course except AHP-L where consistency is absolute due to partly artificial judgments generated by transition rules. Furthermore, the AHP-L method is very sensitive to each of the elements considered where n-1 long path presents extremal among spanning trees.

Relatively large matrices are used to better contrast differences among the models' outcomes. In the majority of cases it was evidenced that although corresponding weights of each criterion are different, the ranking of criteria is generally the same. Adopting only top-ranked criteria and 'deleting' the others may lead to more focused

manipulation of the decision-making process, especially in situations when group decisions should be made by reaching consensus or by applying some of the social choice (election) models.

In all analyzed cases applied methods used the same ratio scale, that is Saaty's scale with 17 discrete values from the set {1/9, 1/8, ... 1/2, 1, 2, ... 9}. Worth mentioning is that this scale is the only starting point in the AHP-L method because generated entries in the upper triangle of all analyzed matrices did not necessarily belong to this scale; in many cases generated entries are either within the scale range (1/9 - 9) but without clear semantic meaning, or outside the scale, again without meaning. Even in the case of this method, the results are similar to the results obtained by the other two methods, especially in ranking decision elements. Nevertheless, note that the mechanism used to create a matrix in AHP-L is dependent on the "seed a_{ij}" selected (the green highlighted cells), and it would be possible to use others but generate different results.

This study concludes that if compared with complete information used by AHP-S, BWM produces a sufficiently good vector of weights, despite the reduced information it relies on. Its result is on a 'safe side' within the decision-making framework in a sense that compensation for the slightly less trustful result than this produced by AHP-S can be found in less effort decision-maker or analyst must put while making judgments. For instance, in the case of the matrix of size 5, BWM requires 7, while AHP-S requires 10 judgments which is a reduction of 30%. In the case of the matrix of size 9, the corresponding number of judgments is 15 and 36, respectively, which is a reduction of 58%, more than a double. The AHP-L is inferior to the other two used methods because it is not semantically realistic, although it is 'heading' toward full consistency and relatively good results. Without fine-tuning to make generated entries in the comparison matrix to fit exact values from the scale (1/9 - 9), this method is not justified from a real-life usage standpoint.

Regarding recommendation which method to use, it is hard to draw a general conclusion. One of the decisive issues could be who is involved in the evaluation and calculation process. The AHP-S method, which has been widely recognized in the scientific community for years, is more flexible in a manner that today there is ready-made software that does not require background knowledge of mathematics and supplementary calculations, and it is suitable for wider use by expert groups. BWM is still a relatively new method for which there is no ready-made software and requires additional time and knowledge in the field of linear programming and the use of available mathematical software to calculate the mathematics of the method.

Acknowledgment. This work was supported by the Ministry of Education. Science and Technological Development of Serbia (Grant No. 451-03-9/2021-14/200117.)

References

1. Tomashevskii, I., Tomashevskii, D.: A non-heuristic multicriteria decision-making method with verifiable accuracy and reliability. Journal of the Operational Research Society, 1-15. (2019)
2. Guitouni, A., Martel, J.M.: Tentative guidelines to help choosing an appropriate MCDA method. European journal of operational research, Vol. 109, No. 2, 501-521. (1998)

3. Hodgett, R.E.: Comparison of multi-criteria decision-making methods for equipment selection. The International Journal of Advanced Manufacturing Technology, Vol. 85, (5-8), 1145-1157. (2016)
4. Langhans, S.D., Reichert, P., Schuwirth, N.: The method matters: a guide for indicator aggregation in ecological assessments. Ecological indicators, 45, 494-507. (2014)
5. Pollesch, N., Dale, V.H.: Applications of aggregation theory to sustainability assessment. Ecological Economics, 114: 117-127. (2015)
6. Pollesch, N., Dale, V.H.: Normalization in sustainability assessment: Methods and implications. Ecological Economics, 130: 195-208. (2016)
7. Thor, J., Ding, S.H., Kamaruddin, S.: Comparison of multi criteria decision making methods from the maintenance alternative selection perspective. The International Journal of Engineering and Science, Vol. 2, No. 6, 27-34. (2013)
8. Velasquez, M., Hester, P.T.: An analysis of multi-criteria decision making methods. International journal of operations research, 10(2), 56-66. (2013)
9. Emovon I, Oghenenyerovwho OS: Application of MCDM method in material selection for optimal design: A review. Results in Materials, 7, 100115 (2020)
10. Cinelli, M., Kadziński, M., Gonzalez, M., Słowiński, R.: How to Support the Application of Multiple Criteria Decision Analysis? Let Us Start with a Comprehensive Taxonomy. Omega, 102261. (2020)
11. Saaty, T.: The Analytic hierarchy process. McGraw-Hill, New York (1980)
12. Yu, D., Kou, G., Xu, Z., Shi, S.: Analysis of collaboration evolution in AHP research: 1982–2018. International Journal of Information Technology & Decision Making (IJITDM), Vol. 20, No. 1, 7-36. (2021)
13. Singh, A., Malik, S.K.: Major MCDM Techniques and their application-A Review. IOSR Journal of Engineering, 4(5), 15-25. (2014)
14. Franek, J., Kresta, A.: Judgment scales and consistency measure in AHP. Procedia Economics and Finance, 12: 164-173. (2014)
15. Dong, Y., Xu, Y., Li, H., Dai, M.: A comparative study of the numerical scales and the prioritization methods in AHP. European Journal of Operational Research, Vol. 186, No. 1, 229-242 (2008)
16. Zhu, B., Xu, Z.: Analytic hierarchy process-hesitant group decision making. European Journal of Operational Research, Vol. 239, No. 3, 794-801. (2014).
17. Wedley, W.C., Schoner, B., Tang, T.S.: Starting Rules for Incomplete Comparisons in the Analytic Hierarchy Process. In: Vargas. L., Zaheidi. F.M. (Eds.) Mathematical and Computer Modelling 17 no. 4–5, Analytic Hierarchy Process, Pergamon Press, Oxford, 93–100. (1993)
18. Herrera, F., Herrera-Viedma, E., Verdegay, J.L.: A rational consensus model in group decision making using linguistic assessments. Fuzzy Sets Syst, Vol. 88, 31–49. (1997)
19. Eessaar, E., Soobik, M.: A decision support method for evaluating database designs. Computer Science and Information Systems, 9(1), 81-106. (2012)
20. Dong, Y.C., Li, C.C., Xu, Y.F., Gu, X.: Consensus-based group decision making under multi-granular unbalanced 2-tuple linguistic preference relations. Group Decis. Negot., Vol. 24, 217–242. (2015)
21. Dong, Y., Chen, X., Herrera, F.: Minimizing adjusted simple terms in the consensus reaching process with hesitant linguistic assessments in group decision making. Inf. Sci., Vol. 297, 95–117. (2015)
22. Tomashevskii, I.: Eigenvector ranking method as a measuring tool: Formulas for errors. European Journal of Operational Research, Vol. 240, No. 3, 774–780. (2015)
23. Liu, L., Wang, W., Jiang, G., Zhang, J.: Identifying key node in multi-region opportunistic sensor network based on improved TOPSIS. Computer Science and Information Systems, Vol. 18, No. 3, 1041–1056. (2021)
24. Ishizaka, A., Lusti, M.: An expert module to improve the consistency of AHP matrices. International Transactions in Operational Research, Vol. 11, No. 1, 97-105. (2004)

25. Rezaei, J.: Best-worst multi-criteria decision-making method. Omega, 53, 49-57. (2015)
26. Rezaei, J.: Best-worst multi-criteria decision-making method: Some properties and a linear model. Omega, Vol. 64, 126-130. (2016)
27. Srdjevic, B.: Combining different prioritization methods in AHP synthesis. Comput. Oper. Res., Vol. 25, 1897–1919. (2005)
28. Saaty, T.: A scaling method for priorities in hierarchical structures. Journal of Mathematical Psychology, Vol. 15, 234–281. (1977)
29. Saaty, T.: Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process. The Analytic Hierarchy Process Series 6, RWS Publications, Pittsburgh (1994)
30. Triantaphyllou, E.: Multi-Criteria Decision Making Methods: A Comparative Study. Kluwer Academic Publishers, Dordrecht. (2000)
31. Dong, Y., Herrera-Viedma, E.: Consistency-driven automatic methodology to set interval numerical scales of 2-tuple linguistic term sets and its use in the linguistic GDM with preference relations. IEEE Trans. Cybern. Vol. 45, 780–792. (2015)
32. Brunelli, M., Canal, L., Fedrizzi, M.: Inconsistency indices for pairwise comparison matrices: a numerical study. Ann Oper Res, Vol. 211, 493–509. (2013)
33. Mikhailov, L.: A fuzzy programming method for deriving priorities in the analytic hierarchy process. Journal of Operational Research Society, Vol. 51, 341-349. (2000)
34. Aguaron, J, Moreno-Jimenez, J.M.: The geometric consistency index: Approximated thresholds. European Journal of Operational Research, Vol. 147, 137–145 (2003)
35. Moreno-Jimenez, J.M., Aguaron, J., Escobar, M.T.: The core of consistency in AHP-group decision making. Group Decision and Negotiations, Vol. 17, 249–265. (2008)
36. Dong, Y., Xiao, J., Zhang, H., Wang, T.: Managing consensus and weights in iterative multiple-attribute group decision making. Applied Soft Computing, Vol. 48, 80–90. (2016)
37. Zhou, J.L., Xu, Q.Q., Zhang, X.Y.: Water Resources and Sustainability Assessment Based on Group AHP-PCA Method: A Case Study in the Jinsha River Basin. Water, Vol. 10, 1880. (2018)
38. Dong, J., Wan, S., Chen, S. M.: Fuzzy best-worst method based on triangular fuzzy numbers for multi-criteria decision-making. Information Sciences, Vol. 547, 1080-1104. (2021)
39. Liang, F., Brunelli, M., Rezaei, J.: Consistency Issues in the Best Worst Method: Measurements and Thresholds. Omega, 102175. (2019)
40. Guo, S., Zhao, H.: Fuzzy best-worst multi-criteria decision-making method and its applications. Knowledge-Based Systems, Vol. 121, 23-31. (2017)
41. Ali, A., Rashid, T.: Hesitant fuzzy best-worst multi-criteria decision-making method and its applications. International Journal of Intelligent Systems, Vol. 34, No. 8, 1953-1967 (2019)
42. Zhang, R., Xu, Z., Gou, X.: An integrated method for multi-criteria decision-making based on the best-worst method and Dempster-Shafer evidence theory under double hierarchy hesitant fuzzy linguistic environment. Applied Intelligence, 1-23. (2020)
43. Bai, C., Kusi-Sarpong, S., Badri Ahmadi, H., Sarkis, J.: Social sustainable supplier evaluation and selection: a group decision-support approach. International Journal of Production Research, Vol. 57, No. 22, 7046-7067 (2019)
44. Mi, X., Liao, H.: An integrated approach to multiple criteria decision making based on the average solution and normalized weights of criteria deduced by the hesitant fuzzy best worst method. Computers and Industrial Engineering, Vol. 133, 83-94. (2019)
45. Srđević, B., Kolarov, V.: AHP evaluation of alternative dispositions of pumping stations in a river basin. Journal of Water Resources Vodoprivreda, Vol. 37, 203-214. (In Serbian) (2005)
46. Srđević, Z., Srđević, B., Bubulj, S., Ilić, M.: Usability and efficiency of best-worst method in water resources related decision-making). Journal of Water Resources Vodoprivreda, Vol. 51, 147-154. (In Serbian) (2019)
47. Srdjevic, B., Srdjevic, Z., Kolarov, V.: Group evaluation of walnut cultivars as a multi-criteria decision-making process. 2004 CIGR International Conference, 11-14 October 2004, Beijing, China (2004)

48. Bubulj, S., Srđević, Z., Ilić, M., Srđević, B.: Selection and evaluation of the criteria for assessing the vulnerability of Ramsar wetlands in Vojvodina Province to the occurrence of drought events (Izbor i vrednovanje kriterijuma za ocenu ranjivosti ramsarskih područja u Vojvodini na pojavu sušnih perioda). Annals of Agronomy, Vol. 44, No. 1, 39-46 (In Serbian) (2020)
49. Srđević, B., Ilić, M.: Group model for evaluating criteria to be used in risk assessments of Ramsar protected wet areas. Journal of Water Resources Vodoprivreda, Vol. 52, No. 306-308, 273-286 (In Serbian) (2020)
50. Ramsar Convention Secretariat Ramsar Handbooks for the Wise Use of Wetlands. 3rd ed. Ramsar Convention Secretariat. Gland, Switzerland. (2004)

Zorica Srđević is a Professor at the University of Novi Sad, Serbia, currently holding the position of Head of the Department of Water Management. She is lecturing on various undergraduate, master and doctoral courses, including water resources systems analysis, allocation of water resources, decision making in agriculture and water management, advanced methods of operational research etc. Her research interests include multicriteria analysis, decision making methodologies and supporting tools and natural resources management. She has been coordinating the team from the Faculty of Agriculture within the several international projects. She is the author of more than 200 articles and other publications, mainly in peer reviewed international and national journals.

Bojan Srdjevic is a Professor Emeritus in the Faculty of Agriculture at University of Novi Sad, Serbia. He is also a Visiting Professor at the School of Polytechnic, Federal University of Bahia (UFBA), Salvador, Brazil, and a Guest Professor at the University of Stuttgart in Germany. He received the Ph.D. degree in technical sciences from the University of Novi Sad (1987), and the MS (1984) and BS degree (1974), both in electrical engineering, from the University of Belgrade in Serbia. His research interests include multiple criteria decision analysis, decision-making under uncertainty, and decision support systems in natural resources management. He has published in Elsevier's Decision Support Systems, Computers & Operations Research and Applied Mathematics and Computation, and Springer's Water Resources Management, among others.

Senka Ždero is a junior researcher at the Faculty of Agriculture at University of Novi Sad (Serbia). At the same time, she is a PhD student od Department of Water Management where she is preparing her PhD research proposal. She has received BSc and Msc degree in Environmental Engineering at Faculty of Technical Sciences at University of Novi Sad and she was awarded with scholarship for young talents in Serbia. During PhD studies, she was a member of national and international project activities, participated in conferences and student competitions, organized and conducted lectures at the Department of Water Management, and she was engaged in scientific research on national project (Grant No. 451-03-9/2021-14/200117.). Senka has published more than 15 scientific papers and 1 technical paper in international and national peer reviewed journals; most research interests are in multi-criteria decision analysis in the field of water management and water quality.

Milica Ilić is a Ph.D. student at the University of Novi Sad, Faculty of Agriculture where she works as a Junior Researcher at the Department of Water Management since 2019. She received B.Sc (2017) and M.Sc (2018) degrees of Water management at the same faculty. Her most research interests include multi-criteria analysis, water quality, and irrigation. She is author and co-author of 15 scientific papers.

Received: December 20, 2019; Accepted: May 02, 2020

Computer Science and Information Systems 19(1):277–307 https://doi.org/10.2298/CSIS201030049S

Explainable Information Retrieval using Deep Learning for Medical images ⋆

Apoorva Singh[1], Husanbir Singh Pannu[1], and Avleen Malhi[2]

[1] Thapar Institute of Engineering and Technology
Patiala India 147004
apoorvasingh.singh1993@gmail.com, hspannu@thapar.edu
[2] Bournemouth University
Fern Barrow, Poole BH12 5BB, UK
amalhi@bournemouth.ac.uk

Abstract. Image segmentation is useful to extract valuable information for an efficient analysis on the region of interest. Mostly, the number of images generated from a real life situation such as streaming video, is large and not ideal for traditional segmentation with machine learning algorithms. This is due to the following factors (a) numerous image features (b) complex distribution of shapes, colors and textures (c) imbalance data ratio of underlying classes (d) movements of the camera, objects and (e) variations in luminance for site capture. So, we have proposed an efficient deep learning model for image classification and the proof-of-concept has been the case studied on gastrointestinal images for bleeding detection. The Explainable Artificial Intelligence (XAI) module has been utilised to reverse engineer the test results for the impact of features on a given test dataset. The architecture is generally applicable in other areas of image classification. The proposed method has been compared with state-of-the-art including Logistic Regression, Support Vector Machine, Artificial Neural Network and Random Forest. It has reported F1 score of 0.76 on the real world streaming dataset which is comparatively better than traditional methods.

Keywords: machine learning, explainable AI, image processing, medical images, capsule endoscopy.

1. Introduction

Machine learning (ML) and artificial intelligence (AI) systems imitates the humans way of learning by associating the cognitive ability and pattern recognition. AI systems are quite complex and intend to mimic human intelligence and automatic learning; ML is about automatic decision making and future predictions through given data distribution and pattern recognition and without explicit coding. Image classification using ML for commercial purposes is good but is still needs improvement in complex images such as medical imaging including cancer cells, endoscopy, x-ray and MRI images. Thus human capabilities and expertise is still superior in these fields as compared to ML.

ML models are flexible, efficient and well-generalized but they are opaque and obscure to understand about how they work. Its power of reasoning is thus limited due to

⋆ This is an extended version of a conference paper [44]: Explaining Machine Learning-Based Classifications of In-Vivo Gastral Images

inability to layout the road map of the decision making phenomenon in case of testing dataset. Thus machine learning model must be able to give justification about the model rationale which can be evaluated by experts to audit the decision making factors. There should be a quantified phenomenon to see how the machine reasons for an outcome in contrast to a human expert for potential conflicts and legal norms. Explainable artificial intelligence (XAI) provides such a formal explanation by the model agnostic interpretation against action taken or decision made by ML model, given the test data and features involved. Figure 1 shows the basic XAI framework with valid questions to be answered

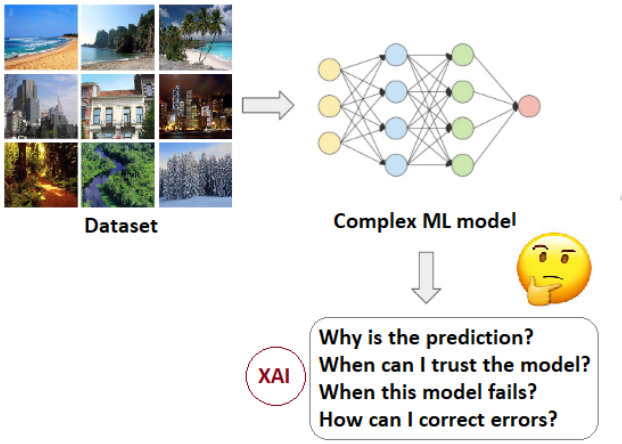


Fig. 1. Basic motivation of explainable artificial intelligence (XAI) is to answer the four questions. The inputs to XAI are the trained model and test dataset.

by the underlying ML model. They include questions like (a) What is the prediction (b) What is the credibility of model (c) Conditions of model failure (d) How to correct errors if any? Afterwards, it justifies the recommended decision through explanation interface. Now the motivation question to consider is that why to use CNN, gastral images and XAI?

1. Actually, CNN imitates human brain to learn by automatic features extraction with numerous layers and neurons as said by father of deep learning Professor Geoffrey Hinton. Professor believed that neural networks are not stuff of science fiction or toil in obscurity. Neural networks are simplified model of how the brain works [41].
2. Gastral images obtained from capsule endoscopy for example, are complex due to movement of camera, organs, noise, compression for transmission. Thus if a model can learn these obscure images then it would also work well on other real world application involving similar constraints and assumptions.
3. XAI explains the test results in regards to feature proportions involved in training just like a psychologist who explores the reasons behind humans' way of thinking. Otherwise the human mind is invincible to traverse for how it works and processes information.

1.1. Convolutional Neural Network (CNN)

Deep learning is a subdivision of machine learning involving neural networks that work similarly to the human brain and are capable of learning from unstructured data such as images [58]. The strength of deep learning is that the low-level features of an image (edges or textures) are compared and connected to the higher-level features (shapes, objects) automatically and autonomously by the model using enormous amount of training data. It involves the hierarchy of concepts for information extraction in form of image features. In traditional machine learning algorithms like SVM, Linear regression, Random Forest, the features have to be extracted from the images manually, whereas in deep learning like CNN, features are learned from the raw data automatically by the network [31].

A CNN is a feed-forward neural network that is used to identify the complicated features in the dataset. It can devise and derive features from unprocessed data automatically and can work on massive amounts of data. CNN performs remarkably good in the fields of images-analysis, pattern-detection, edge-detection, image/object recognition, powering vision in robots, for self-driving vehicles, etc. It also reduces computational burden by offering the automatic feature extraction and briefly explained in the subsection below. There are a variety to deep learning models available for CNN architecture. Similar to the CNN model proposed by Jia et al. [37], each of the model CNN involved in the ensemble has been comprised of eight-layers that involve three convolutional layers (C1-C3), two fully-connected layers (FC1, FC2) and three pooling layers (MP1-MP3). In our execution, rectified linear units (ReLUs) are used as the activation function in convolutional layers (C1-C3) and the first fully-connected layer (FC1). Max-pooling is used in the pooling layers (MP1-MP3) to detect the maximal activation over input patches. Lastly, the output of the second fully-connected layer (FC2) consists of two neurons (bleeding and normal) and can be activated by a soft-max regression function, which is defined as:

$$f_\theta(x^{(i)}) = \begin{bmatrix} Q(y=1|x^{(i)};\theta) \\ Q(y=2|x^{(i)};\theta) \\ \vdots \\ Q(y=M|x^{(i)};\theta) \end{bmatrix} = \frac{c}{b} \tag{1}$$

where

$$c = \begin{bmatrix} \exp(\theta^{(1)\top} x^{(i)}) \\ \exp(\theta^{(2)\top} x^{(i)}) \\ \vdots \\ \exp(\theta^{(M)\top} x^{(i)} \end{bmatrix} \tag{2}$$

and

$$b = \sum_{k=1}^{M} \exp(\theta^{(k)\top} x^{(i)}) \tag{3}$$

So

$$f_\theta(x^{(i)}) = \frac{c}{b} \tag{4}$$

where $x^{(i)} \in \Re^n$ are the input attributes with the corresponding labels $y(i)$. M is the number of classes. The model parameters $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)} \in \Re^n$ are trained to minimize the loss function:

$$L(\theta) = -(1)\left[\sum_{m=1}^{t}\sum_{n=1}^{K} 1\left\{y^{(m)} = n\right\} \log \frac{\exp(\theta^{(n)\top} x^{(m)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(m)})}\right] \tag{5}$$

where t denotes the size of the training set. Particularly, in the binary classification setting, we have $y^{(i)} \in \{0, 1\}$ and $K = 2$. CNN derives the features from images automatically, which results in a reduced computational burden and reducing the semantic gap between humans way of perceiving and algorithmic approach. Image features are learned during the training of the network on the image dataset. One more benefit of using CNN is that only the number of filters and the filter size is required to be defined, whereas the values of the filters are determined by CNN automatically during the training phase. Unlike most of the other ML techniques, object detection in images is carried out by CNN regardless of the location of the object to be recognized. Pooling feature of CNN also prevents overfitting of the network.

1.2. Explainable Artificial Intelligence

Explainable artificial intelligence is getting a lot of attention nowadays. Machine learning algorithms have been used for medical imaging but these models do not explain the assessment they make. Humans cannot trust these models since they do not understand the reason of their assessment. Although there is an increasing number of works on interpretable and transparent machine learning algorithms, they are mostly intended for the technical users. Explanations for the end-user have been neglected in many usable and practical applications. Many researchers have applied the explainable framework to the decisions made by model for understanding the actions performed by a machine. There are many existing surveys for providing an entry point for learning key aspects for research relating to XAI [6]. Anjomshoae et al. [10] gives the systematic literature review for literature providing explanations about inter-agent explainability. The classification of the problems relating to explanation and black box have been addressed in a survey conducted by Guidotti et al. [32] which helped the researchers to find more useful proposals. Machine learning models can be considered reliable after integration of explainability feature for the expert analysis and retraining of the model. Contextual Importance and Utility has a quite significance in explaining the machine learning models by giving the rules for explanation [26]. Framling et al. provides the black box explanations for neural networks with the help of contextual importance utility [25][27].

There are many methods used for providing the explanations for example; LIME (Local Interpretable Model-Agnostic Explanations) [3], CIU (Contextual Importance and Utility) [26], ELI5 [2], Skater [5], SHAP (SHapley Additive exPlanations) [4] etc. Most of them are the extensions of LIME which is an original framework and approach being proposed for model interpretation. These techniques provide model prediction explanations with local interpretation, model prediction values with shape values, building interpretable models with surrogate tree based models and much more. Contextual Importance (CI) and Contextual Utility (CU) explains the prediction results without transforming the model into an interpretable one. These are numerical values represented as visuals and natural language form for presenting explanations for individual instances [26]. The CIU has been used by Anjomshoae et al. [9] to explain the classification and prediction results made by machine learning models for Iris dataset and Car Pricing dataset where the authors have CIU for justifying the decisions made by the models. The prediction results are explained by this method without being transformed into interpretable model. It yields the explanations for linear as well as non linear models demonstrating the felexibility of the method.

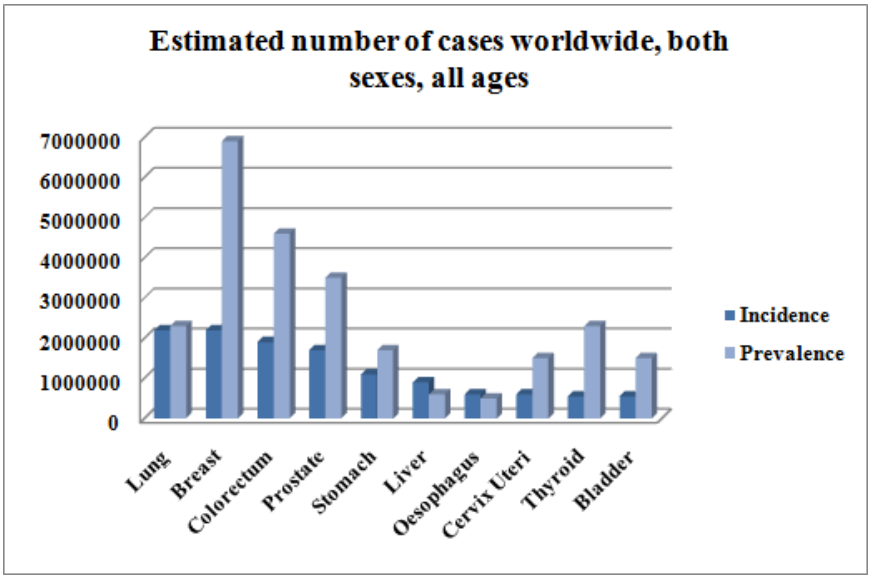


Fig. 2. The estimated incidence and prevalence of worldwide cancer cases [17], [49]. Most of them are related to gastrointestinal organs

1.3. Capsule Endoscopy

In Gastroenterology (GI), gastric cancer is the fifth most common cancer worldwide and seventh most prevalent in accordance to the GLOBOCAN 2018 as shown in Figure 2. In few states of India such as Tamil Nadu, Assam, Kerala, and Karnataka, the malignant

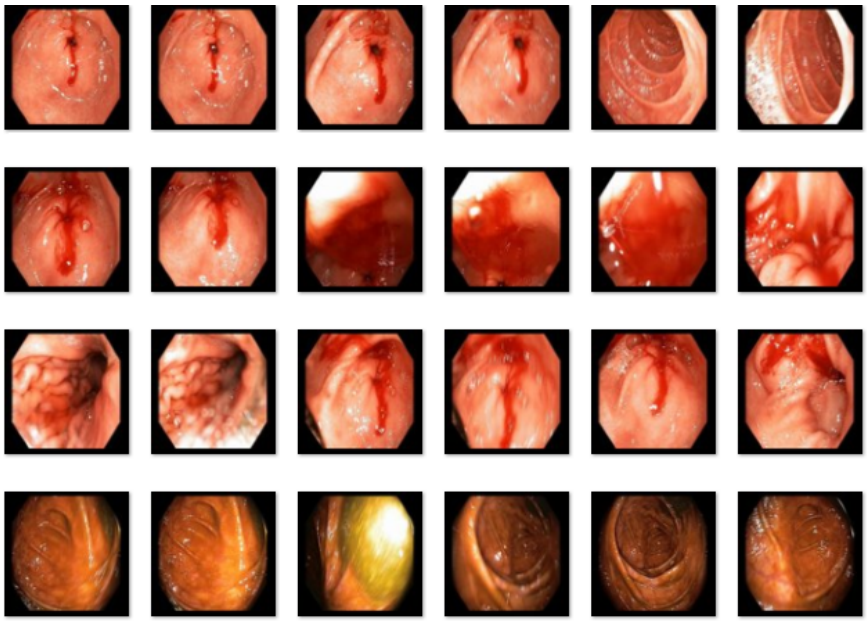


Fig. 3. Sample video shots of GIngastrointestinal tract with bleeding symptoms. Capsule endoscopy images collected from PSRI hospital, New Delhi - www.psrihospital.com

tumors or cancers are most commonly present in the squamous cells of the esophageal carcinoma section of the digestive tract [18]. For males, 10% of the total cancers is Colorectal cancer and that makes it the third most prevalent cancer in males with cases of 663,000 all over the world. Whereas, it is the other most frequent cancer in females with cases of 570,000, which is 9.4% of the total cases globally [49], [17].

GI tract comprises of several organs such as digestive canal, throat or esophagus, liver, duodenum, bile ducts, pancreas, gallbladder, small intestine, and large intestine, colon, rectum. Gastroenterology also addresses the complications that may harm these organs including polyps, ulcers, cancer, and esophageal reflux. Wireless capsule endoscopy (WCE) technique was first introduced in year 2000 [36]. The small intestine is one of the complex organs to diagnose and heal without conducting surgery. CE supports physicians to see inside the regions of our body that are not readily reached with traditional endoscopy. The video obtained by the pill-sized camera used in WCE is carefully observed by the doctor for irregularities inside the digestive system. The manual review of WCE video is not time-efficient, with an average reading time of 45–120 minutes approximately [21], [**?**]. RAPID software is available with the PillCam kit to automatically detect the bleeding frames out of all the frames from video captured by the camera. The efficiency obtained by this software is not satisfactory [36], [22], [50]. It may also skip the frames with inactive bleeding or frames with blood spots of very small sizes. As the results obtained by PillCam's RAPID software algorithm are not efficient, there is a need for a better algorithm to detect the anomalies in WCE frames [55].

Our goal is to propose an automated soft-computing technique for the detection of presumed frames that can have the appearance of bleeding. This may significantly lessen the evaluation time while the ultimate judgment is still left to the endoscopy experts. The paper has been organised as follows: next section 2 is about the contemporary literature survey for research motivation and gaps; section 3 is the proposed methodology; section 4 is performance metrics; section 5 is about results of the case study; section 6 is about conclusion and future directions.

2. Literature Review

This section studies the motivational state-of-art work done in the field of image segmentation to detect plausible abnormalities like polyp, tumor, ulcer and bleeding. Table 1 gives the comparison of the existing machine learning approaches proposed for the endoscopic bleeding detection whereas figure 4 gives the classification of the different approaches used for the anomaly detection in the digestive tract. Most of the investigated modern techniques based on this comparison have worked on the automated detection of abnormalities seen in the GI Tract Endoscopy.

Table 1. State-of-the-art comparison for the bleeding detection in endoscopic images

Type	Study	Year	Dataset	Method	Results
ML	[48] Obukhova et al.	2019	Bleeding frames in KVASIR (open), 8000 images	Block-based segmentation and color characteristics	95% accuracy
ML	[53] Tuba et al.	2019	Bleeding framesinF. Deeba, "Bleeding images and corresponding ground truth of CE	Texture and color features (HSI, CIE ULBP), GrowCut	0.85 Dice similarity coefficient, 0.092 misclassification error images, 50 images
ML	[56] Jia et al.	2018	Bleeding frames in 1000 WCE images from random patients.	Superpixel-color histogram, KNN	0.9922 accuracy
ML	[30]Ghosh et al.	2018	Bleeding zones in Kid: Koulaouzidis-iakovidis database for capsule	Semantic segmentation, SegNet, CNN	94.42% accuracy Endoscopy, 335 images
ML	[13] Bchir et al.	2018	Multiple bleeding frames in Imaging PillCam, 1275 frames	Fuzzy C-means clustering, KNN	90.92% accuracy
ML	[28] Ghosh et al.	2018	Bleeding frames in CE (Online), 2350 images	Color Histogram of Block Statistics	97.85% accuracy

ML	[52]. Sivakumar et al.	2018	Bleeding frames	Superpixel segmentation, Semi-Naïve Bayesian classifier	N/A
ML	[29] Ghosh et al.	2017	Bleeding frames in The capsule endoscopy website (public), 2350 frames from 32 WCE videos	Cluster based statistical feature extraction	97.05% precision
CNN	[33] Hajabdollahi et al.	2019	Bleeding regionsinF. Deeba, "leeding images and corresponding ground truth of CEimages	Multi-layer perceptron, CNN	AUC-ROC 0.97, DICE for CNN 0.869 for MLP = 0.831
CNN	[38] Jia et al.	2017	Bleeding frames in 1500 WCE images	Handcrafted features based CNN	F1 score 0.9285
CNN	[37] Jia et al.	2016	Bleeding frames in 10,000 WCE images	Deep CNN with SVM	Recall 99.2%, F1 Score 99.5%
Other	[34] He et al.	2018	Hookworm frames in West China Hospital, 440K images	CNNs (Edge extraction and Hookworm classification network)	88.5% accuracy
Other	[54] Vieira et al.	2019	Small bowel angioectasiasinKID (public), 27 images, and PillCam, MiroCam in Hospital of Braga (Portugal), 300 frames	Maximum a Posteriori, Expectation-Maximization	Sensitivity 96%, Specificity 94.08%, Accuracy 95.58%
Other	[7] Alaskar et al.	2019	Ulcer in Dr. Khoroo'sMedical Clinic (Online available), 1875 images	AlexNet and GoogleNet CNN	100% accuracy with learning rate 0.0001
Other	[12] Aoki et al.	2019	Erosions and Ulcer frames in The University of Tokyo Hospital, Japan, 15800 images	Deep CNN with a Single Shot Multibox Detector	91.5% accuracy
Other	[47] Nawarathna et al.	2019	Mucosal abnormality in MiroCam WCE images	Filter bank, local binary patterns, Textons histogram	Recall 92%, Specificity 91.8%

Other	[42] Leenhardt et al.	2018	GI Angiectasia during small bowel in 6360 frames from pre-med students	CNN-based semantic segmentation	Sensitivity 100%, Specificity 96%
Other	[23] Diamantis et al.	2019	GI Abnormalities in Endovis challenge, 10,000 images, and KID, 2352 images	Look-Behind Fully CNN (LB-FCN)	AUC 93.5%
Other	[14] Billah et al.	2017	Polyp frames in Endoscopic Vision Challenge, more than 14,000 images	Color wavelet, CNN and SVM	Accuracy 98.34%, Sensitivity 98.67%, Specificity 98.23%
Other	[20] Deeba et al.	2017	Bleeding frames in PillCam SB1 and PillCam SB2, 8872 images	SVM ensemble and exhaustive feature selection	Accuracy 95%, Specificity 95.3%, Sensitivity 94%

2.1. Machine Learning Techniques

Various machine learning techniques have been used for automation of the bleeding detection in WCE images. In [48], authors proposed a method for automatic feature extraction and detection of bleeding in endoscopy images. The endoscopy images are segmented using block-based segmentation. The local features are discovered using color characteristics. Different algorithms are investigated in this approach to classify the bleeding and non-bleeding images. The performance parameters are also calculated to test the effectiveness of these algorithms. The accuracy obtained by the proposed approach is 95%. In [53], this approach presented bleeding detection in WCE images based on region-based feature extraction. This method extracts features from HSI and CIE color spaces. Authors used a uniform library binary pattern to label each region. The secondary set of features can be extracted from the grayscale image. Classification of regions is done by support vector machine (SVM) into three categories, namely, non-bleeding region, bleeding region, and background. GrowCut algorithm is used for the concluding segmentation of CE images.

Xing et al. [56] introduced a three-step algorithm for automated detection of bleeding in endoscopy images. Authors have done Key-frame extraction and edge removal as the first step of preprocessing. In the second step, they separated the bleeding images from the dataset of all the frames using the KNN classifier, applying the concept of principle color spectrum by utilizing the superpixel color histogram feature. In the last step, the segmentation of bleeding regions from the various color spaces is executed by securing a 9-D color feature vector at the superpixel feature. The accuracy attained is 0.99.

The system proposed by [13], focused on identifying the multiple blood specks in the several frames captured from the WCE video. To overcome the performance degradation due to the small size of the region of interest, the authors suggested an unsupervised ML

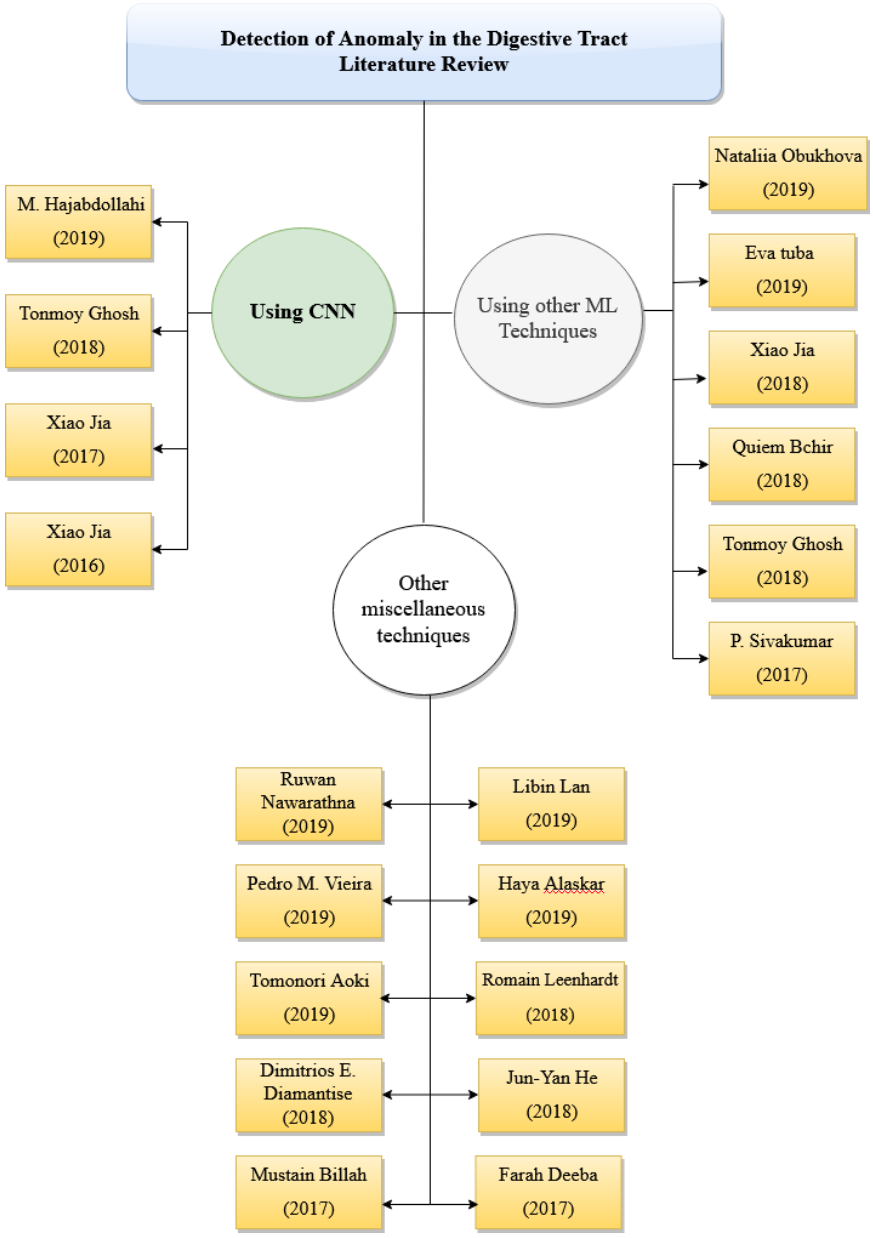


Fig. 4. Literature review organization

technique. It breaks the principal classification query into many confined classification queries. This technique cluster the training set using fuzzy C-means, then implement improvised KNN on the determined centers rather than the entire training dataset. Authors have also analyzed the performance and results of the proposed algorithm as opposed to the typical KNN and SVM.

In the paper [52], Naive Bayes classifier and superpixel segmentation are used for automated obscure bleeding disclosure in WCE video dataset. They used the color histogram for region discovery and feature extraction. In the final step, they adopted an improvised semi naive bayesian classifier. In [28], authors have classified bleeding and non-bleeding frames in wireless CE images utilizing color histogram of block statistics. Local feature extraction is done using a WCE image block, preferably of an individual pixel. For the contrasting color panels of RGB color space, index values are defined. So, the authors used index values to extract the color histogram. Color histogram is useful in securing distinct color texture characteristics. Feature reduction using color histogram and principal component analysis is adopted to decrease the dimension of these local features. Extracted local features that do not result in any computational strain provides the blocks with bleeding regions. Authors used a public dataset of 2350 images that renders 97.85% accuracy.

In [29], authors have worked on the system that distinguishes the bleeding images and regions from the WCE dataset. WCE images are preprocessed to convert into a color space defined by green to red pixel-ratio. These transformed images are used to obtain various analytical features from the overlying spatial blocks. These various blocks are then clustered into two clusters using K-means based clustering (unsupervised). These clusters are used to obtain cluster-based features. These features combined into a global feature is used along with differential cluster-based features to detect blood zone frames using an SVM (supervised learning classifier). The proposed system achieved 97% precision.

2.2. CNN based segmentation techniques

Many authors have proposed CNN based methods for WCE bleeding detection mechanisms. Authors proposed a way in [30] for detection of bleeding regions in CE images employing a semantic segmentation based on deep neural network, called SegNet. CNN is trained using the successive layers of SegNet. Bleeding regions are detected in CE images by segmenting the test images on the trained CNN. The efficiency achieved is 94.42%. In [33], authors advised a simplified neural network (NN) for the detection of bleeding frames by performing automatic bleeding regions segmentation on the CE dataset. Fitting color channels are chosen as inputs to the neural network. An multi-layer perceptrons and CNN are applied to conduct image classification individually. They decreased the number of computational operations. The performance of the recommended systems is assessed using the DICE score. The area under the receiver operating curve (AUC-ROC) is 0.97. Due to significantly fewer computations, CNN is proved to be more beneficial than multi-layer perceptron.

Authors in the paper [37] presented a high-level detection system for the bleeding frames in the WCE image dataset. This system is implemented using a deep CNN that detects both active and inactive frames. Authors have designed CNN to have 8 layers. The network composes 3 convolutional layers, 3 pooling layers, and two fully connected layers. The ReLU or the rectifier function is implemented at the convolutional layers and

the first FC layer to increase non-linearity in the network. Images are made of different objects that are not linear to each other. Without applying this activation function, the image classification is treated as a linear problem, while it is in actual a non-linear one. Pooling layers implement Max-Pooling to preserve the main features while also reducing the size of the image. This helps reduce overfitting. One of the main causes for the overfitting to occur is that too much information fed into CNN. Especially if that information is not relevant in classifying the image. SVM is more beneficial in the case where the user wants to depreciate the entropy loss for prediction. So, CNN is devised by substituting the SoftMax regression function at the secondary FC layer with an SVM classifier. But this proposed system resulted in complex computations and required a large dataset of designated images for training the CNN. So, the authors presented a way in the paper [38] that implements the deep-learning technique of CNN with handcrafted features that are obtained using a k-means clustering technique. This model is focused on detection of frames with active and inactive bleeding. This approach reduces the computational cost incurred in the training of CNN.

2.3. Other techniques

Authors in [47] have worked on a computerized way of finding abnormalities in the endoscopy images. These abnormalities can be erosion, erythema, ulcerations, polyp, bleeding, etc. This proposed method examines images for varied textures so that, it can differentiate abnormal images from the normal ones effectively. The distribution of different textures in an endoscopy image can be captured using a textons histogram. It is done by applying a FB (filter bank) and LBP (local binary patterns). This proposed approach gives 92% recall and 91.8% specificity on WCE images. In [40], authors have worked on a computer-aided system that uses various approaches to detect the abnormalities in the images obtained from CE. Authors have employed CNN, region recommendation, transfer learning. In the first step, they have used a CascadeProposal to recommend high-recall regions and abnormal frames. In the second step, the authors used a multi-regional combination technique to detect the regions of interest and have also operated a salient region segmentation approach to catch certain region spots. For object boundary filtration, a dense-region fusion algorithm is applied. And lastly, to increase the efficiency of the proposed model, transfer learning tactics are exercised in CNN.

Authors of [23] presented a computer-aided look-behind fully CNN (LB-FCN) algorithm to automatically catch the anomalies in CE images. It uses blocks of parallel convolutional layers with varied filter dimensions to derive the multi-scale features from WCE images. All the LB linked features are combined with the features deduced from prior layers. As LB-FCN has fewer free parameter as compared to conventional CNN, it makes it much easier to train the network on smaller datasets. The AUC performance of LB-FCN achieved is 93.5%. In [20], they have aimed at reducing the analysis time of the WCE video frames by presenting a computer-aided approach that automatically identifies the abnormal frames. They have worked on an ensemble of two SVMs that are based on HSV and RGB color spectrums. Feature selection and parameter tuning are done by using a nested cross-validation approach. For the betterment of performance, exhaustive analysis is carried out to decide the best feature sets. The dataset used comprises of 8872 WCE frames. This fusion system renders an accuracy of 95%, specificity of 95.3% and sensitivity of 94%. A CNN is suggested in [42] for the GI angioectasia detection during

small bowel in CE images. Local features are extracted through deep feature extraction using an approach of segmentation of images based on semantics. Authors created a semantic segmentation-based CNN for classification of GI angioectasias. The sensitivity and specificity achieved is 100% and 96% respectively.

The work done in [54] aims for an automated way for the detection of angioectasias in WCE image dataset. This approach depends on the automatic separation of a region of importance. That region is chosen by applying a module for the task of image segmentation based on the approach of Maximum a Posteriori where a new hastened variant of the Expectation-Maximization is also advised. This proposed method attained sensitivity and specificity values of 96% and 94.08% respectively with 95.58% accuracy in a database comprising 800 WCE frames designated by two gastroenterologists. In this paper [7], they have conducted ulcer and lesion detection and classification in WCE dataset employing two pre-trained CNN, GoogleNet and AlexNet. These two networks perform object classification to obtain ulcer and non-ulcer frames. Due to a huge number of layers in GoogleNet, AlexNet resulted in double the efficiency of GoogleNet for training. The efficiency of both networks is enhanced by tuning the parameters. It is also found in this study that higher the learning rate of the network, higher is the resulting accuracy. The learning rate of 0.0001 renders adequate results for both the networks.

AlexNet attained 100% accuracy with the rate of 0.001. Authors in [12] trained a deep CNN to distinguish ulcers and erosions in small bowel CE images automatically. This CNN is based on an single shot multiBox detector that holds 16 layers. The CNN is trained using SSD on the 5,360 images. For the testing phase, 10,440 WCE images are fed to CNN, out of which, 440 are of erosions or ulcers. This system renders an accuracy of 91.5%. Whereas, in the paper [14], they have focused on decreasing the misidentification rate for a polyp in CE images. This will support the professionals in finding the most significant regions to pay consideration. Features are deduced using color wavelet and CNN. These extracted features are then fed to a train an SVM. SVM will classify the CE frames into the polyp region and normal frames classes. They achieved 98.34% accuracy. The study performed by [34] has focused on detecting a hookworm abnormality in wireless capsule endoscopy images. They have adopted the deep learning algorithm to recognize the tube-like pattern of hookworm. For the better activity of the classification, two neural networks are employed, edge extraction CNN and hookworm classification CNN. Both the CNNs are seamlessly integrated into the recommended system to evade edge feature caching. The edge extraction CNN provides the tubular regions and the hookworm CNN gives the feature maps. Both the results are integrated into the pooling layers to produce an intensified feature map accentuating tubular region and achieved 88.5% accuracy.

Our research concentrates on the automatic bleeding detection in capsule endoscopy videos using a convolutional neural network. Literature review organization in terms of techniques has been shown in Figure 4 while highlighting the CNN based approaches. CNN is fast, efficient, and it needs limited preprocessing of the images [23], [7], [12]. We have used CNN in the proposed method for the detection of the bleeding frames in WCE images along with explanation of test results and the impact of involved features.

3. Methodology

A computerized system for bleeding detection in WCE images is proposed to catch the presence of a threat in the digestive tract that caused the bleeding. A dataset of WCE videos with frames holding both bleeding and non-bleeding frames is collected (from PSRI hospital, New Delhi) and used for the proposed approach.

3.1. Proposed classification approach

The basic layout of the proposed methodology is shown in Figure 5. It shows preprocessing, denoising and image learning. The obtained WCE video dataset is fed to the VLC software to extract images. Figure 3 shows the snapshot of the extracted image dataset. The number of WCE images extracted for the dataset is 2,621 with 505 bleeding and 2,116 normal frames. All the images extracted are resized consistently to prepare them for the proposed model. High-resolution images involve more computations and higher memory specifications. If the input is a scaled-down variant of the bigger images, then determining key features in the initial layers will be easier for the network. So, we have resized our WCE images to a size of 100×100 pixels for a scaled-down CNN. Stationary Discrete Wavelet Transform tool in MATLAB (SWT) is used as a de-noising algorithm to smoothen the images, remove noise/artifacts and undesired distortions present in the images. Processed images are then fed to the convolutional neural network (CNN). The complexity of the model depends upon the complexity of the data. We can start by adding only one hidden layer in the network with neurons and then check the quality of trained network using cross-validation. Subsequently, deepen the network by adding more layers and neurons until the validation becomes stable. CNN works by automatically extracting features from the image using the training set of WCE images. CNN parameters and options are tuned to obtain better performance on the empirical trail basis. The methods for tuning the options of the CNN is discussed later in this section in detail. The trained network is applied to the test set of images for classification for comparative analysis. The accuracy, sensitivity, and specificity of the network are measured from confusion matrix and precision-recall graph curve is also plotted. Performance of the proposed model is compared with traditional machine learning methods such as Linear classifier, Support Vector Machines (SVM), Artificial Neural Networks (ANN) and Random Forest (RF) algorithms.

Dataset preparation The WCE dataset used in this research is collected from Pushpawati Singhania Research Institute, (PSRI) Delhi, India of gastroenterology through a known gastroenterologist. A set of 2,621 WCE images is extracted from the video using VLC software at a rate of 2 frames per second. This dataset comprises of 505 bleeding frames and 2,116 normal frames. For video to image sampling using the VLC software, one can set the properties like image dimensions, bit depth, frame extraction rate, etc. in the preferences tab of the VLC software. The WCE images have color homogeneity problem as the color to be detected has a wide range of shades. The blood shade may fluctuate extensively from bright red to deep red, brownish, also containing redness of the normal skin tissues.

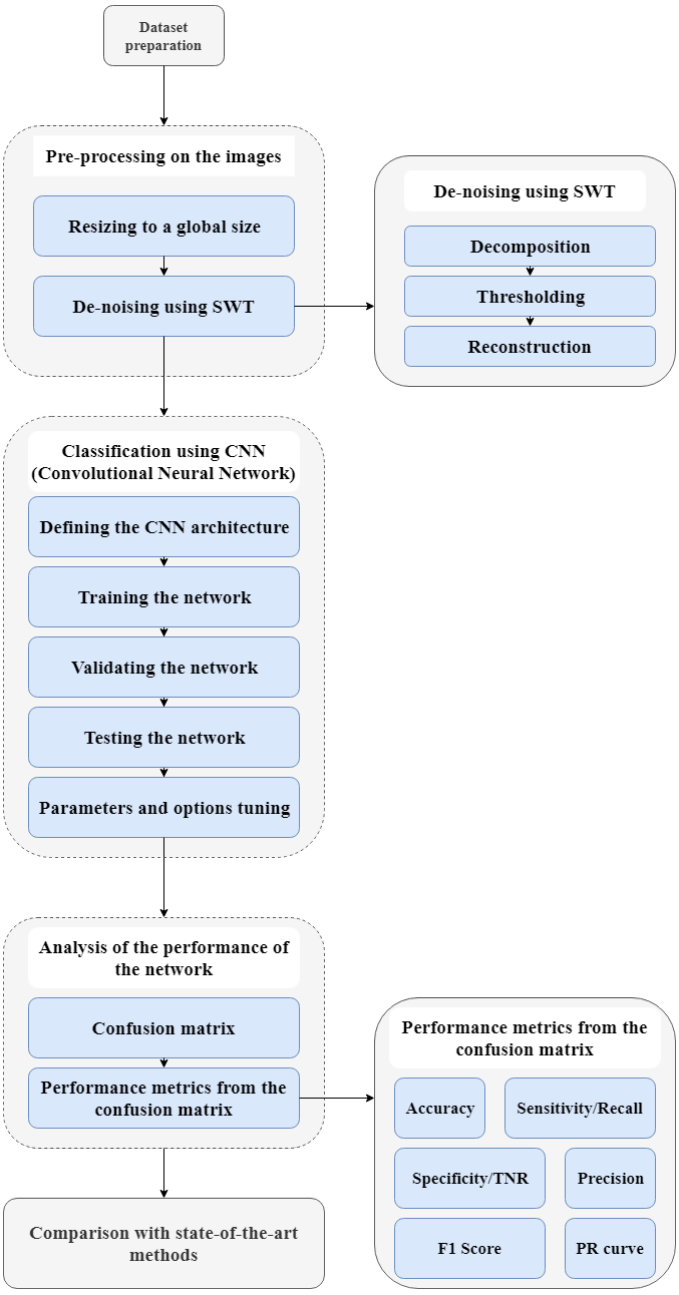


Fig. 5. Pipeline of the proposed technique

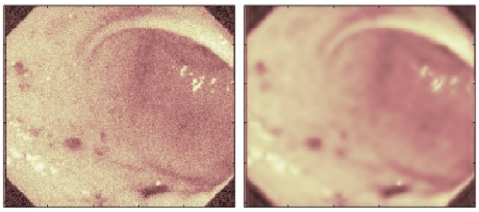


Fig. 6. (a) Input image (b) De-noised image

Pre-processing After acquiring the WCE image dataset, image preprocessing steps are done to enhance the performance of the network model to which the processed images are being fed. Pre-processing steps of image processing techniques are applied to the acquired image dataset for getting enhanced images to contrast the binary classification. The steps for preprocessing are as follows.

The first step of the pre-processing is to resize all the images to the a specific size for example 100×100. Deep learning network need consistent sizes and optimal size images for efficient and faster performance. High-resolution images involve more computations and higher memory specifications. If the input is a scaled down variant of the bigger images, then determining key features in the initial layers will be easier for the network [15].

De-noising using SWT We have implemented stationary wavelet transform (SWT) for noise removal and examine the statistically non-predictable signals, particularly at the region of discontinuities [45]. Since images have discontinuities at the edges, they can be presented spatially in a multi-resolution manner by using the SWT technique [8]. SWT is a wavelet transform that is used to transform signals or images to derive valuable information for the analysis as well as saves the computational cost and reduce required memory space. As we are working with medical images, the loss in information can adversely affect the result. SWT de-noising requires decomposition, level thresholding, and inverse transforming for reconstructing the image. For the step of decomposition, SWT algorithm decomposes an image into coefficients to get details about the image like contrast, correlation, energy, homogeneity, entropy, etc. It is done by a choosing a specific wavelet for image decomposition such as Haar, Daubechies, SymN, etc.

We have chosen Daubechies for our work by experimental analysi and also suggested by [11] for the image decomposition and de-noising. Daubechies wavelet being an orthogonal wavelet does not unnecessarily color the white noise, preserves the energy and relatively offer longer support [46]. The SWT decomposition of the image results into four sub-images to get the coefficients. These sub-images are obtained by applying vertical and horizontal filters that are low-pass filter (LPF) and high-pass filter (HPF). The resultant four sub-images of varying contrast, orientations, sharpness, and resolutions are called as approximations (average components) and detail components (horizontal, vertical and diagonal). After de-noising the images with these threshold limits, we apply an inverse transform to reconstruct the original image from the sub-images but without noise. The reconstruction is the reversed course of decomposition and known as inverse wavelet

transform. Figure 6 shows the smoothing effect by the SWT denoising which is easier for the CNN to perform binary classification.

Classification using CNN Classification of frames into bleeding and non-bleeding category is done by employing a convolutional neural network (CNN). De-noised images are fed into ConvNet (CNN) for detection of the bleeding in the processed set of images.

1. **Defining the CNN architecture:** ConvNet takes in images as three-dimensional objects. Neurons in the layers of CNN are organized in 3 dimensions: width, height, depth. Depth is same as number of color channels in the input image. Determining the optimal number of hidden layers and neurons that are incorporated into the network is a crucial part. Underfitting arises as a result of using too few neurons in the layers to sufficiently recognize the signals in a complex data set, whereas using too many neurons can give rise to time complexity and overfitting issues. Overfitting happens when the network has excessive data processing capability such that the confined amount of data comprised in the training set is not sufficient to train all of the neurons in the hidden layers.
   Another issue that can arise is the increased time taken to train the network with an overly huge number of neurons in the layers. Some adjustment needs to be made between an abundant and inadequate number of neurons in the hidden layers. We can start by adding only one hidden layer in the network with neurons and then check validation accuracy of the network through cross-validation. To optimize the network, we deepen the network gradually so that it can deal with more complex data and avoid underfitting. According to Heaton research in [1], more than 2 hidden layers in the network generally result in enabling the model to learn complex representations and extract features. We can try to optimize the performance by adding more neurons in the existing hidden layers or adding new hidden layers. Heaton [1] suggests that we can use a thumb rule instead of hit-and-trial which can be time-consuming and laborious. For defining a satisfactory number of neurons in the hidden layers, following steps should be used:
   - The number of neurons should fall within the range of the size of the input layer and the output layer.
   - It shall be 2/3 the size of the input layer, as well as the size of the output layer.
   - It needs to be less than twice the size of the input layer.

   The very first layer in the network is an image input layer that receives the denoised images scaled to $100 \times 100 \times 3$, for three color channels. We have incorporated 3 Conv layers, 2 pooling layers and, 1 fully connected layer. Due to small input image patch, 3 layers of convolution is enough, two pooling layers to be in between convolutional layers and one fully connected layer for binary classification. A batch normalization layer is always introduced after every Conv layer, followed by a ReLU layer to maintain the non-linearity of the image. Non-linear features of an image are the changes and shifts in pixels, the edges, borders, various colors, etc. Linear images do not appear normal to the human eye as they lack brightness and above-mentioned features. Conventional layers are linear in nature to learn the concept hierarchy and various non-linear activation functions have been incorporated in CNN for efficient feature extraction. A combination of linear inputs cannot generate a non-linear output, so without a non-linear activation function, the network will act like a single-layer

perceptron to accumulate all the layers of the network and follow a the standard of linear function [57]. As the second layer, a 2-D convolutional layer is defined with eight 3×3 convolutions with stride 1 and the same padding as input images. Pooling layer is applied with the arguments of 2×2 max pooling, stride of 2 and same padding. One fully connected layer for the bleeding and non-bleeding classes is defined and followed by a soft-max and classification layer to apply soft-max function and evaluate the cross-entropy loss.

2. **Training the network:** Dataset is split into training, validation and testing sets with standard proportion of 70%, 15%, and 15% respectively. So, CNN is trained on a random 70% training set split while regulating the weights on the network for less training error until the validation criteria is met.
3. **Validating the network:** Validation set is used to minimize over-fitting in the network and to ensure that any increment in accuracy over the training dataset results into an increment in accuracy over a test dataset that has not been yet exposed to the network or the network has not been trained on it such as validation dataset.
4. **Testing the network:** The trained network is then run on the test dataset of images for the classification of the bleeding and non-bleeding frames. This will yield the performance comparison and prediction robustness of the network.
5. **Parameters and options tuning:** For better performance of the network, parameter and options are tuned to optimal values for the underlying data distribution. Stochastic gradient descent with momentum (SGDM) is used as a solver with mini-batches of sizes 10. The total number of training samples in a batch is the batch size. Too small batch size results into gradient descent not being smooth, slow learning of the model and error may oscillate too much, whereas too high batch size results into the longer time required to do one training iteration with relatively small results [51]. SGDM is one of the best optimization algorithms as it helps in preventing oscillations. The number of epochs is the number of passes through the whole training set while training the network. Using a large number of epochs can result in over-fitting of the network and using a very small number of epochs result in an under-fit network [38]. Early stopping process enables us to use a large number of epochs but stops the network training as soon as a validation criterion is met. A smaller learning rate decreases the speed of the learning in the network, but it enables the network to converge smoothly. First, we chose a small learning rate varying between 1e-5 to 1 and then we check the performance of our network. To improve performance, a smaller learning rate is used [24]. The maximum number of epochs used in the proposed network is 6 with a learning rate of 1e-5 for better training after checking the performance of the network with different hit-and-trial values for these parameters.

3.2. Proposed Explanation Approach

Currently, in medical domain, XAI functionality is a necessary requirement for many machine learning-based medical research, education and clinical decision making scenarios. Systems for solving the medical domain explanation problem can be distinguished into two types; post-hoc systems and ante-hoc systems. Post-hoc systems help in providing local explanations for a particular decision made by machine learning so that it can be made interpretable on demand rather than explaining the whole systems behavior. One of

the algorithms that enables post-hoc explainability is LIME. Local interpretable model-agnostic explanations (LIME) [3] is the original Python implementation of this explanation technique. LIME takes two inputs: the neural network as generated by TensorFlow and the result of a specific frame to generate a matrix representation of the regions that triggered the corresponding classification. Ante-hoc systems are interpretable by design and referred to as *glass-box* approaches in the literature [35]; examples are decision trees, linear regression and fuzzy inference systems. In an applied science context, LIME has already been used for explaining machine learning models for the heat failure detection in air handling units [43]. Base idea of the explanation process has been published by Avleen et al. [44] and the proposed work is an extension of the previous work.

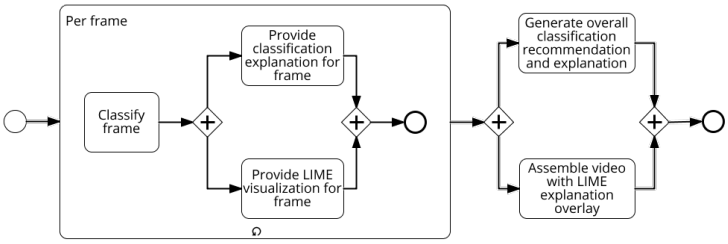


Fig. 7. Classification and Explanation Process

The classification and explanation process has been depicted in Figure 7, which describes the classification procedure by machine learning model as well as explanation and visualization by LIME for each image frame. The image data set is trained with a machine learning model (CNN in our example). The trained model is given to our proposed XAI model for providing classifications and explanations for these binary sick and healthy image classes. The overall explanations for the whole test data can be provided to the medical professionals for assisting them in decision making. The overall recommendation and explanation is provided by the health-care professional by making an aggregate ranking system for providing the severity of the intestinal bleeding in the patient case. The architecture of the proposed model is depicted in Figure 8 where the whole process can be divided into four segments: Pre-processing, applying the CNN model, explanation-generation using LIME, and decision-assistance for healthcare professionals.

4. Performance Metrics

Various performance metrics that have been used for the evaluation of the efficiency of the proposed system are as follows.

4.1. Confusion Matrix

The exactness and reliability of a system are calculated through a confusion matrix which is also termed as an error-matrix. This matrix supports in getting a clear idea of the performance of the system by analyzing the mis-classification rate and accuracy.

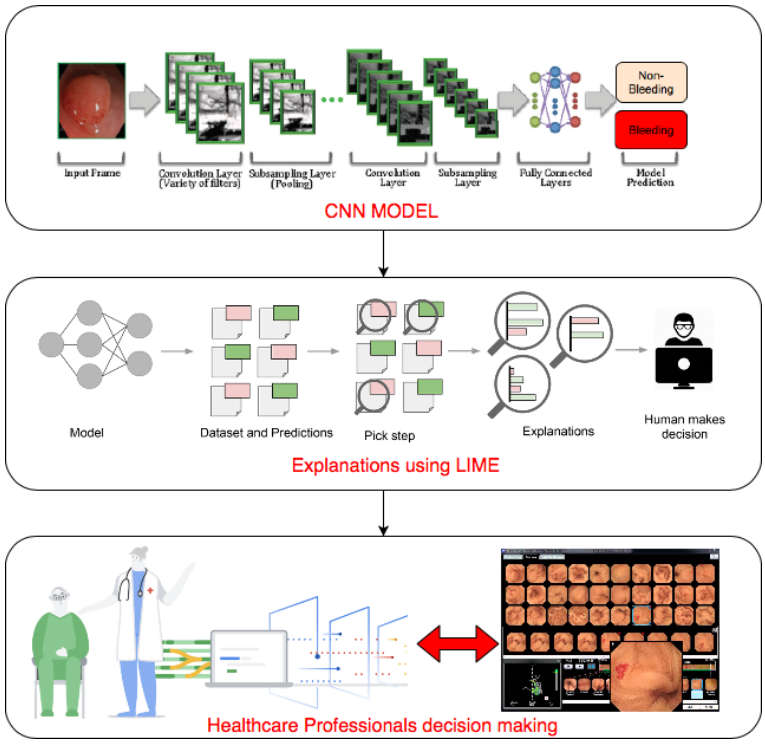


Fig. 8. The Architecture for the proposed model

1. True Positive (TP) is when the predicted and actual classes are identical tp true class. For instance, a frame that has bleeding present in it is getting predicted in the bleeding class.
2. True Negative (TN) is when the predicted and actual element is negative. For instance, a normal image without any bleeding getting predicted in normal class.
3. False Positive (FP): When the system predicts an element to be in true class but in actual it does not. For example a non-bleeding frame getting predicted in the bleeding class.
4. False Negative (FN): When the system predicts that an element does not belong to a false class but in actual it does. For example an actual bleeding frame is predicted as normal by the model.

In our research, the WCE image dataset is slightly unbalanced as it comprises of smaller ratio of bleeding frames as compared to normal frames. Accuracy alone is not considered a good evaluation metric in cases of unbalanced data classification. Both false negatives (FN) and false positives (FP) are important in medical image classification. In the case of endoscopy images, the cost of false negatives is as important as the false positives [19]. The damage of a bleeding frame to not get detected is worse than the damage of detecting a normal frame as bleeding as all the frames predicted as bleeding will be observed by the physician for the final judgment but the frames predicted as normal frames will probably

be overlooked. Nevertheless we do not want a large number of false positives in our prediction as it will reduce the efficiency of the network.

4.2. Accuracy

Accuracy refers to the total number of correct classifications done by the network out of the total number of examples. As shown in equation 6, accuracy is the rate of true predictions by all the true and false predictions combined.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{6}$$

4.3. Sensitivity or Recall

Sensitivity is the rate of accurately predicted positives to genuine positives. Recall gives us an idea about a model's performance proportionate to false negatives. As shown in equation 7, recall focuses on catching all the frames that have "bleeding" with the prediction as "bleeding", not just concerning catching frames correctly. For medical image classification problems, high sensitivity is preferred as it indicated high true positive value and low number of false negatives.

$$Specificity = \frac{TP}{(TP + FN)} \tag{7}$$

4.4. Specificity

Specificity is the rate of accurately predicted negatives to the actual negatives. Specificity is the exact reverse of sensitivity. Equation 8 shows the specificity derived from a confusion matrix.

$$Specificity = \frac{TN}{(TN + FP)} \tag{8}$$

4.5. Precision

Precision is the rate of accurately predicted positives to all the predicted positives. In equation 9, precision shows the proportion of the frames that are detected as having the presence of bleeding, actually had bleeding. Recall provides us an idea about a network's performance concerning false negatives, the frames that the network missed. Precision provides us with the idea of its performance concerning false positives for the frames that were predicted. Precision is about predicting frames correctly, whereas Recall is about prediction all the positive frames correctly. So, for minimizing false negatives, we have to focus on getting Recall as best as possible with a decent and acceptable Precision value. The values of both Precision and Recall can be monitored by a single value performance metric called as F1 score.

$$Precision = \frac{TP}{(TP + FP)} \tag{9}$$

4.6. F1 Score

To consider the role of both precision and recall, the F1 score is computed as in 10 which is simply the harmonic mean of precision and recall. In the case of unbalanced class distribution in the dataset, F1 score is a better evaluation metric than accuracy. Low value of F1 score indicates a problem when one of the Precision and Recall has a low value. In that case, F1 score is closer to the smaller value than the bigger value out of these two.

$$F1\ Score = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{10}$$

4.7. Precision-Recall curve

To demonstrate the trade-off in precision and recall, PR curve gives a more informational depiction of the performance of the network with unbalanced dataset [19], [16]. The area under the PR curve varies from 0 to 1 and also gives an idea about the network's performance. If AUC is close to 1 then the model is considered as good. The closer the curve is to the top-right edge, the more reliable the system. Henceforth, a greater area under the curve (AUC) symbolizes that the system has higher precision and higher recall.

5. Result Evaluation

In this research, we have used MATLAB 2018a for implementing the proposed model. MATLAB offers a good data visualization and it also offers a large number of toolboxes/ apps for processing and ploting image dataset with ease of usage. We have preprocessed the WCE images and trained a convolutional neural network. The performance of proposed model is compared with traditional machine learning methods including linear discriminant model, SVM, ANN, Random Forest. The performances of the models are evaluated from metrics derived from confusion matrix like accuracy, specificity, sensitivity, precision and F1 score.

5.1. System and Data Configuration

ANT PC (10 Cores 20 Threads), Intel C612 Chipset Motherboard with single Socket, 32GB ECC RAM 2400Mhz, Dual Nvidia GeForce RTX 2080TI 11GB, Intel Server Heatsink, 250GB Samsung 860 Evo SATA SSD, 2TB Western Digital HDD, 1000W 80+ Gold Power Supply, Ubuntu operating system. The WCE dataset of $2,621$ images are collected from Pushpawati Singhania Research Institute (PSRI) institute of gastroenterology in Delhi India. It comprises of 505 bleeding frames and 2,116 normal frames.

Various textures, color and contrast types in the image dataset are shown in the Figure 9. These sample images render the frames with both malignant and benign features like bleeding, polyp, wrinkles and contractions. The images with turbid present in the system and images of walls of digestive system are also depicted in the figure 9. In our research, we focused only on extracting the frames that have bleeding present in them from the entire dataset.

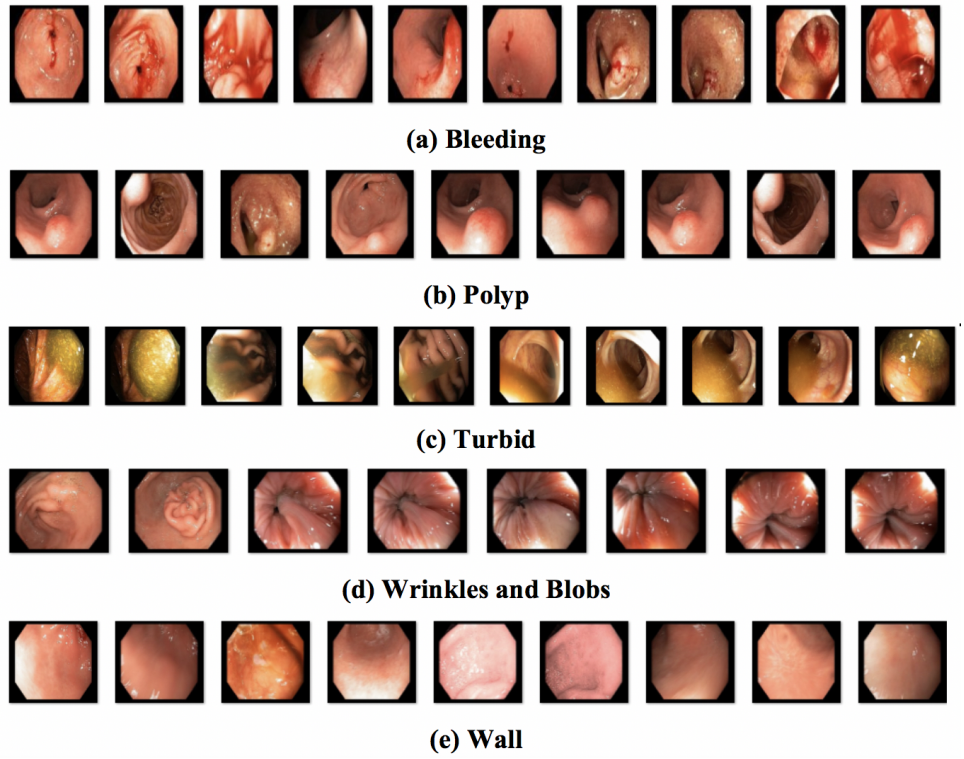


Fig. 9. Various ailments in the dataset. (a) Bleeding (b) Polyp (c) Turbid (d) Wrinkles and blobs (e) Membrane wall

5.2. Classification Results

Confusion matrix of the proposed system is plotted in Table 2 to calculate the evaluation metrics. Metrics like accuracy, sensitivity (recall), specificity, precision, F1 score are evaluated using the values of TP, FP, TN, FN using confusion matrix entries in Table 2. In Table 3, the accuracy ratios have been calculated using confusion matrix (Table 2) of the binary classification has been shown. It is observed that the false negatives are 8 and false positives are 24. The test accuracy obtained is 91.9%. The values taken from confusion plot are the TP = 52, TN = 309, FP = 8, FN = 24, sensitivity (recall or TPR) = 68.5%, specificity (TNR) = 97.5%, precision = 86.7%. Precision and recall are used to calculate F1 score. As we can see in Table 4, test accuracy is better than validation accuracy, so it can be concluded that network is not a result of over-fitting. Recall has slightly lower value and precision is higher. F1 score falls in between both the recall and precision. The recall is affected due to the high number of false negatives.

The precision-recall curve is plotted using the results from the proposed model and shown in Figure 10, the curve depicts the trade-off between precision and recall of the network. The area under the PR curve (AUC-PR) is calculated to be 0.82. The value of

Table 2. Confusion matrix of test dataset which is 15% of (2,116+500) = 393 images. Positive means bleeding and negative means normal images.

		Predicted		
		Positive	Negative	Total
Actual	Positive	52	8	60
	Negative	24	309	333
	Total	76	317	393

Table 3. Performance metric ratios such as sensitivity, specificity, precision & accuracy using confusion matrix in Table 2 for binary classification

		Predicted:Yes	*Predicted:No*	**Overall**
Bleeding	*Actual:Yes*	13.2% (52)	2% (8)	86.7% (Precision)
Normal	*Actual:No*	6.1% (24)	78.6% (309)	7.2%
Overall		68.4% (Sensitivity)	97.5% (Specificity)	91.9 % (Accuracy)

AUC-PR is affected by the low recall value in our experiments. PR Curve depicts the trade-off between precision and recall values of the model.

5.3. Comparative Analysis

In Table 5, we have compared the performance of our model with traditional machine learning models based on standard evaluation metrics. Due to the fact that WCE image dataset being imbalanced in nature, the performance evaluation of the models cannot rely on just accuracy and therefore, advanced metrics are calculated [39]. The F1 score of SVM and Random forest is close to the proposed model. But, these two models require manual feature extraction, segmentation, thresholding, etc. Whereas, the proposed model does not rely on handcrafted features. So, it reduces the human intervention of surveying the feature extraction and ML techniques followed by orchestrating various algorithms together. Hence, the proposed CNN system has outperformed to detect the bleeding frames in the WCE images. The performance of the proposed system on WCE images is slightly better (with F1 score of 0.76) than the other traditional machine learning algorithms shown in compared in Table 5. The performance of a classifier is not always evaluated just by the accuracy but other important metrics like precision, recall, F1 score and ease of implementation must also considered to analyse the efficiency and reproducibility of the model.

Table 4. Performance evaluation of the CNN mode on WCE dataset

Sr.	Metric	Value
1	Validation Accuracy	90.84%
2	Test Accuracy	91.92%
3	Sensitivity	68.42%
4	Specificity	97.48%
5	Precision	86.67%
6	F1 Score	0.7647

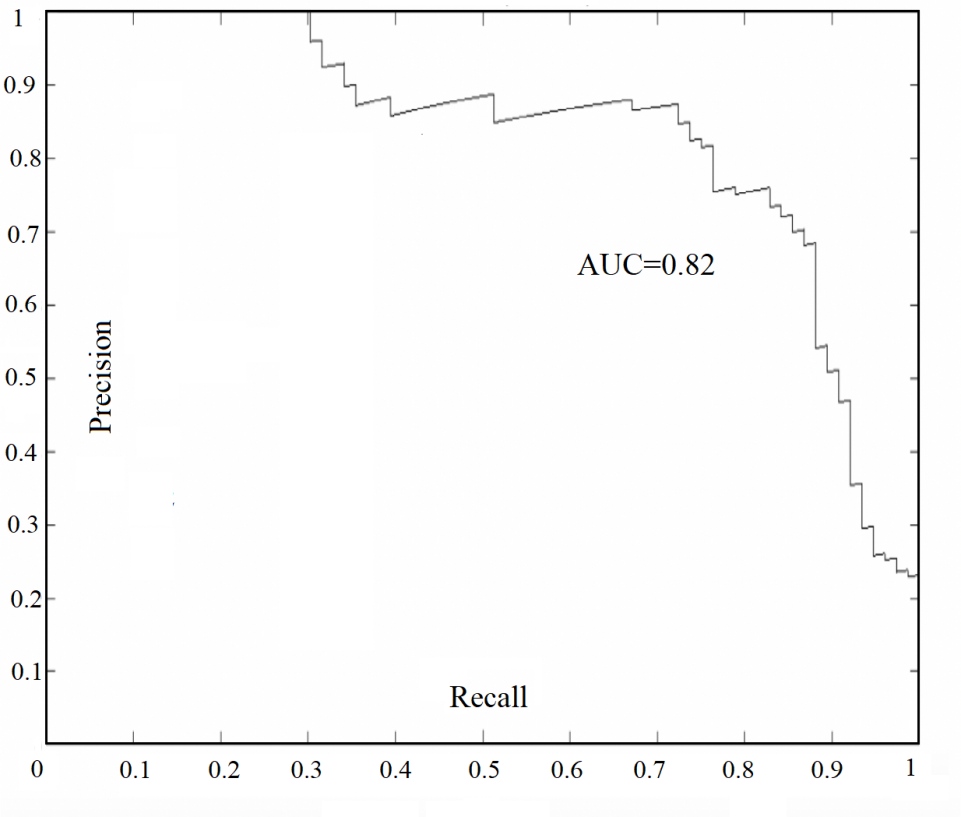


Fig. 10. Precision-Recall curve of CNN model

Table 5. Comparative analysis of the proposed model with state-of-the-art techniques

Sr.	ML Model	Accuracy	Sens.	Spec.	Precision	F1 Score
1	CNN	91.92%	68.42%	97.48%	86.67%	0.76
2	Logistic Regression Model	89.72%	82.27%	90.84%	58.92%	0.68
3	SVM	91.01%	92.19%	91.86%	63.29%	0.75
4	ANN	89.43%	56.34%	97.21%	82.79%	0.67
5	Random Forest	91.11%	87.95%	91.61%	62.30%	0.73

Moreover, the computational burden and human intervention is also decreased in the proposed model as there is no requirement for manual feature extraction as it is in other above stated models. The manual feature extraction by using color histogram and co-occurrence matrix increases the computational steps and hassle for the model application. Moreover it also requires a ground level knowledge of image processing and basic machine learning model. Whereas, using CNN network is trending nowadays for its ease of direct image input for learning and its working strategy is similar to humans way of learning.

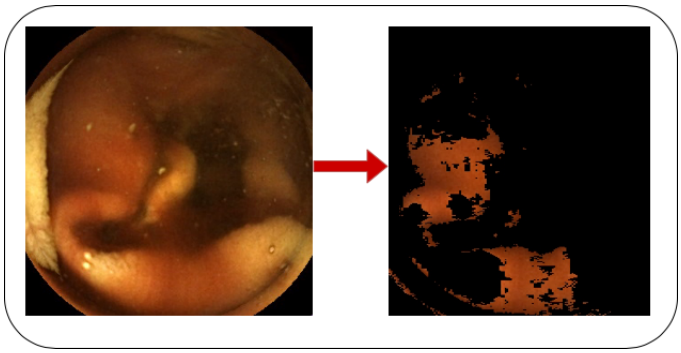


Fig. 11. Annotating the red lesions in capsule endoscopy images

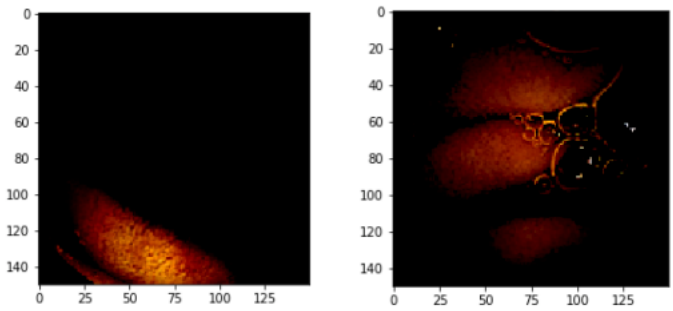


Fig. 12. The bloody regions are shown to give the glimpse of the areas due to which the image is classified as bloody image

5.4. Explanation Results with LIME

LIME provides explanations for bleeding images by drawing the boundaries over the bloody areas in the tested image known as annotations as shown in the Figures 11, 12 and 13. The decision made by black box machine learning model gets justified using LIME XAI model for further analysis by the area experts and adds on the model reliability. Bleeding region is highlighted for features or areas due to which the image is classified as bleeding case. LIME has been tested for all the bleeding images in the validation and test dataset similar to Figure 13. Thus the proposed CNN model is easy to use, efficient and transparent through model agnostic XAI technique for complex medical image applications. Nonetheless, the proposed technique is reproducible and scalable for any image classification application.

6. Conclusion

The proposed CNN model classifies and annotates the bleeding frames in wireless capsule endoscopy (WCE) video dataset. A real time video has been obtained from a known gastroenterologist. Images are sampled from the WCE video using VLC software and

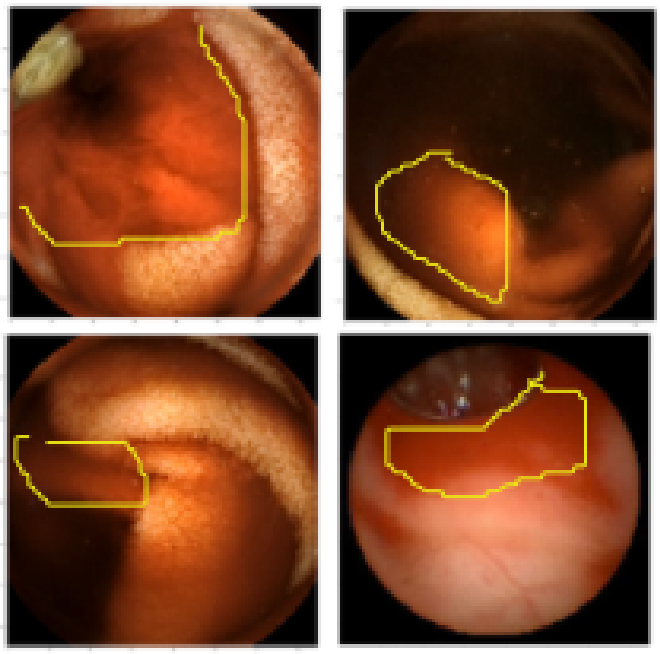


Fig. 13. LIME explanations provided in the form of boundaries for bloody regions

pre-processed to a get standard sized, de-noised images for efficient machine learning model. A convolutional neural network (CNN) is designed and proposed for the classification of WCE frames into the bleeding and non-bleeding categories. The performance of the proposed model is compared with other traditional machine learning models on the basis of evaluation parameters. We have proposed and prototyped an explainable machine learning tool that should be used by medical experts as a decision-support system to detect gastroenterological bleeding faster and in a more reliable manner. The *assumption* for using proposed model is the availability of a GPU system since CNN classification is computationally complex. Traditional machine learning models with handcrafted features could be faster if GPU system is not available.

There is still room for improvement by exploring other deep learning models and variants of explainable artificial intelligence (XAI). Image moments should also be analysed since they yield robust image features which are rotation, scaling and traslation invariants. Other modalities should also be incorporated for a multi-modal machine learning such as free expert text available with the images.

Acknowledgments. This research work has been sponsored by the Seed Money project grant at Thapar Institute of Engineering and Technology Patiala India under Grant TU/DORSP.

References

1. Deep learning project. https://jhui.github.io/2018/02/11/

How-to-start-a-deep-learning-project/ (2018), [Online; accessed 04-June-2019]
2. ELI5. https://github.com/TeamHG-Memex/eli5 (2019), [Online; accessed 04-June-2019]
3. LIME. https://towardsdatascience.com/ (2019), [Online; accessed 04-June-2019]
4. shap. https://github.com/slundberg/shap (2019), [Online; accessed 04-June-2019]
5. Skater. https://github.com/oracle/Skater (2019), [Online; accessed 04-June-2019]
6. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (xai). IEEE Access 6, 52138–52160 (2018)
7. Alaskar, H., Hussain, A., Al-Aseem, N., Liatsis, P., Al-Jumeily, D.: Application of convolutional neural networks for automated ulcer detection in wireless capsule endoscopy images. Sensors 19(6), 1265 (2019)
8. Alwan, I.M.: Color image denoising using stationary wavelet transform and adaptive wiener filter. Al-Khwarizmi Engineering Journal 8(1), 18–26 (2012)
9. Anjomshoae, S., Främling, K., Najjar, A.: Explanations of black-box model predictions by contextual importance and utility
10. Anjomshoae, S., Najjar, A., Calvaresi, D., Främling, K.: Explainable agents and robots: Results from a systematic literature review. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. pp. 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems (2019)
11. Anutam, R.: Performance analysis of image denoising with wavelet thresholding methods for different levels of decomposition. International Journal of & its Applications 6(3), 35–46 (2014)
12. Aoki, T., Yamada, A., Aoyama, K., Saito, H., Tsuboi, A., Nakada, A., Niikura, R., Fujishiro, M., Oka, S., Ishihara, S., et al.: Automatic detection of erosions and ulcerations in wireless capsule endoscopy images based on a deep convolutional neural network. Gastrointestinal endoscopy 89(2), 357–363 (2019)
13. Bchir, O., Ismail, M.M.B., AlZahrani, N.: Multiple bleeding detection in wireless capsule endoscopy. Signal, Image and Video Processing 13(1), 121–126 (2019)
14. Billah, M., Waheed, S.: Gastrointestinal polyp detection in endoscopic images using an improved feature extraction method. Biomedical engineering letters 8(1), 69–75 (2018)
15. Bitenc, M., Kieffer, D., Khoshelham, K.: Evaluation of wavelet denoising methods for small-scale joint roughness estimation using terrestrial laser scanning. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences 2 (2015)
16. Boyd, K., Eng, K.H., Page, C.D.: Area under the precision-recall curve: point estimates and confidence intervals. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 451–466. Springer (2013)
17. Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R.L., Torre, L.A., Jemal, A.: Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: a cancer journal for clinicians 68(6), 394–424 (2018)
18. Chitra, S., Ashok, L., Anand, L., Srinivasan, V., Jayanthi, V.: Risk factors for esophageal cancer in coimbatore, southern india: a hospital-based case-control study. Indian journal of gastroenterology 23(1), 19–21 (2004)
19. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on Machine learning. pp. 233–240 (2006)
20. Deeba, F., Islam, M., Bui, F.M., Wahid, K.A.: Performance assessment of a bleeding detection algorithm for endoscopic video based on classifier fusion method and exhaustive feature selection. Biomedical Signal Processing and Control 40, 415–424 (2018)

21. Delvaux, M., Gay, G.: Capsule endoscopy: technique and indications. Best Practice & Research Clinical Gastroenterology 22(5), 813–837 (2008)
22. D'Halluin, P.N., Delvaux, M., Lapalus, M.G., Sacher-Huvelin, S., Soussan, E.B., Heyries, L., Filoche, B., Saurin, J.C., Gay, G., Heresbach, D.: Does the "suspected blood indicator" improve the detection of bleeding lesions by capsule endoscopy? Gastrointestinal endoscopy 61(2), 243–249 (2005)
23. Diamantis, D.E., Iakovidis, D.K., Koulaouzidis, A.: Look-behind fully convolutional neural network for computer-aided endoscopy. Biomedical Signal Processing and Control 49, 192–201 (2019)
24. Drakos, G.: How to select the right evaluation metric for machine learning models: Part 1 regression metrics. Towards Data Science. Saatavissa: https://towardsdatascience. com/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-1-regrression-metrics-3606e25beae0. Hakupäivä 3, 2019 (2018)
25. Främling, K.: Explaining results of neural networks by contextual importance and utility. In: Proceedings of the AISB'96 conference. Citeseer (1996)
26. Främling, K.: Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère. Ph.D. thesis, INSA de Lyon (1996)
27. Främling, K., Graillot, D.: Extracting explanations from neural networks. In: Proceedings of the ICANN. vol. 95, pp. 163–168. Citeseer (1995)
28. Ghosh, T., Fattah, S.A., Wahid, K.A.: Chobs: Color histogram of block statistics for automatic bleeding detection in wireless capsule endoscopy video. IEEE journal of translational engineering in health and medicine 6, 1–12 (2018)
29. Ghosh, T., Fattah, S.A., Wahid, K.A., Zhu, W.P., Ahmad, M.O.: Cluster based statistical feature extraction method for automatic bleeding detection in wireless capsule endoscopy video. Computers in biology and medicine 94, 41–54 (2018)
30. Ghosh, T., Li, L., Chakareski, J.: Effective deep learning for semantic segmentation based bleeding zone detection in capsule endoscopy images. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 3034–3038. IEEE (2018)
31. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
32. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) 51(5), 93 (2018)
33. Hajabdollahi, M., Esfandiarpoor, R., Soroushmehr, S., Karimi, N., Samavi, S., Najarian, K.: Segmentation of bleeding regions in wireless capsule endoscopy images an approach for inside capsule video summarization. arXiv preprint arXiv:1802.07788 (2018)
34. He, J.Y., Wu, X., Jiang, Y.G., Peng, Q., Jain, R.: Hookworm detection in wireless capsule endoscopy images with deep learning. IEEE Transactions on Image Processing 27(5), 2379–2392 (2018)
35. Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B.: What do we need to build explainable ai systems for the medical domain? arXiv preprint arXiv:1712.09923 (2017)
36. Iddan, G., Meron, G., Glukhovsky, A., Swain, P.: Wireless capsule endoscopy. Nature 405(6785), 417 (2000)
37. Jia, X., Meng, M.Q.H.: A deep convolutional neural network for bleeding detection in wireless capsule endoscopy images. In: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 639–642. IEEE (2016)
38. Jia, X., Meng, M.Q.H.: Gastrointestinal bleeding detection in wireless capsule endoscopy images using handcrafted and cnn features. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 3154–3157. IEEE (2017)
39. Kaur, H., Pannu, H.S., Malhi, A.K.: A systematic review on imbalanced data challenges in machine learning: Applications and solutions. ACM Computing Surveys (CSUR) 52(4), 1–36 (2019)

40. Lan, L., Ye, C., Wang, C., Zhou, S.: Deep convolutional neural networks for wce abnormality detection: Cnn architecture, region proposal and transfer learning. IEEE Access 7, 30017–30032 (2019)
41. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature 521(7553), 436–444 (2015)
42. Leenhardt, R., Vasseur, P., Li, C., Saurin, J.C., Rahmi, G., Cholet, F., Becq, A., Marteau, P., Histace, A., Dray, X., et al.: A neural network algorithm for detection of gi angiectasia during small-bowel capsule endoscopy. Gastrointestinal endoscopy 89(1), 189–194 (2019)
43. Madhikermi, M., Malhi, A., Främling, K.: Explainable artificial intelligence based heatrecycler fault detection in air handling unit (2019)
44. Malhi, A., Kampik, T., Pannu, H., Madhikermi, M., Främling, K.: Explaining machine learning-based classifications of in-vivo gastral images. In: 2019 Digital Image Computing: Techniques and Applications (DICTA). pp. 1–7. IEEE (2019)
45. Masumdar, R., Karandikar, R.: Comparative study of different wavelet transforms in fusion of multimodal medical images. International Journal of Computer Applications 146(11) (2016)
46. Mortazavi, S., Shahrtash, S.: Comparing denoising performance of dwt, wpt, swt and dt-cwt for partial discharge signals. In: 2008 43rd International Universities Power Engineering Conference. pp. 1–6. IEEE (2008)
47. Nawarathna, R., Oh, J., Muthukudage, J., Tavanapong, W., Wong, J., De Groen, P.C., Tang, S.J.: Abnormal image detection in endoscopy videos using a filter bank and local binary patterns. Neurocomputing 144, 70–91 (2014)
48. Obukhova, N., Motyko, A., Timofeev, B., Pozdeev, A.: Method of endoscopic images analysis for automatic bleeding detection and segmentation. In: 2019 24th Conference of Open Innovations Association (FRUCT). pp. 285–290. IEEE (2019)
49. Rawla, P., Barsouk, A.: Epidemiology of gastric cancer: global trends, risk factors and prevention. Przeglad gastroenterologiczny 14(1), 26 (2019)
50. Signorelli, C., Villa, F., Rondonotti, E., Abbiati, C., Beccari, G., de Franchis, R.: Sensitivity and specificity of the suspected blood identification system in video capsule enteroscopy. Endoscopy 37(12), 1170–1173 (2005)
51. Silva, J., Histace, A., Romain, O., Dray, X., Granado, B.: Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. International Journal of Computer Assisted Radiology and Surgery 9(2), 283–293 (2014)
52. Sivakumar, P., Kumar, B.M.: A novel method to detect bleeding frame and region in wireless capsule endoscopy video. Cluster Computing pp. 1–7 (2018)
53. Tuba, E., Tomic, S., Beko, M., Zivkovic, D., Tuba, M.: Bleeding detection in wireless capsule endoscopy images using texture and color features. In: 2018 26th Telecommunications Forum (TELFOR). pp. 1–4. IEEE (2018)
54. Vieira, P.M., Silva, C.P., Costa, D., Vaz, I.F., Rolanda, C., Lima, C.S.: Automatic segmentation and detection of small bowel angioectasias in wce images. Annals of biomedical engineering 47(6), 1446–1462 (2019)
55. Westerhof, J., Koornstra, J.J., Weersma, R.K.: Can we reduce capsule endoscopy reading times? Gastrointestinal endoscopy 69(3), 497–502 (2009)
56. Xing, X., Jia, X., Meng, M.H.: Bleeding detection in wireless capsule endoscopy image video using superpixel-color histogram and a subspace knn classifier. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 1–4. IEEE (2018)
57. Yu, F.: A comprehensive guide to fine-tuning deep learning models in keras
58. Yuan, Y., Meng, M.Q.H.: Deep learning for polyp recognition in wireless capsule endoscopy images. Medical physics 44(4), 1379–1389 (2017)

Apoorva Singh got her Master's degree in Computer Science & Engineering from Thapar Institute of Engineering and Technology Patiala India. She did her Bachelor's of En-

gineering degree in Information Technology from Quantum school of technology India. Her research interests are machine learning and image processing.

Husanbir Singh Pannu is an assistant professor in Computer Science & Engineering Department at Thapar Institute India. His research areas are machine learning, text and image analysis. He got his PhD from University of North Texas USA and was postdoc fellow at Trinity College Dublin Ireland.

Avleen Malhi is a senior lecturer in data science and AI at Bournemouth University UK. She received her PhD in autonomous vehicles (2016), ME in Computer Science (2012) from Thapar Institute India and BE in Computer Science Engineering (2010). As part of her PhD research (2012-2016), she was working on an industrial project by Tata Consultancy Services. Between 2016-2018, she was working as an Assistant professor in computer science at Thapar University and from 2019-2020, she was working as postdoctoral researcher at Aalto University, Finland.

Received: October 3, 2020; Accepted: August 18, 2021.

Computer Science and Information Systems 19(1):309–326 https://doi.org/10.2298/CSIS210617055X

RICNN: A ResNet&Inception Convolutional Neural Network for Intrusion Detection of Abnormal Traffic

Benhui Xia[1], Dezhi Han[1], Ximing Yin[2], and Na Gao[1]

[1] College of Information Engineering, Shanghai Maritime University
200031 Shanghai, China
{201930310092}@stu.shmtu.edu.cn, dzhan@shmtu.edu.cn
[2] The Third Research Institute of Ministry of Public Security
201306 Shanghai, China

Abstract. To secure cloud computing and outsourced data while meeting the requirements of automation, many intrusion detection schemes based on deep learning are proposed. Though the detection rate of many network intrusion detection solutions can be quite high nowadays, their identification accuracy on imbalanced abnormal network traffic still remains low. Therefore, this paper proposes a ResNet &Inception-based convolutional neural network (RICNN) model to abnormal traffic classification. RICNN can learn more traffic features through the Inception unit, and the degradation problem of the network is eliminated through the direct mapping unit of ResNet, thus the improvement of the model's generalization ability can be achievable. In addition, to simplify the network, an improved version of RICNN, which makes it possible to reduce the number of parameters that need to be learnt without degrading identification accuracy, is also proposed in this paper. The experimental results on the dataset CICIDS2017 show that RICNN not only achieves an overall accuracy of 99.386% but also has a high detection rate across different categories, especially for small samples. The comparison experiments show that the recognition rate of RICNN outperforms a variety of CNN models and RNN models, and the best detection accuracy can be achieved.

Keywords: Intrusion Detection, ResNet, Inception, CNN, Traffic Classification, Imbalanced Samples.

1. Introduction

With the maturity of cloud computing technology, more and more outsourced data are stored in the cloud [1–4]. Worse still, the wrongdoers are tempted by the enormous value of digital asserts such as users' privacy and transaction records in this era of cloud computing [5]. They thus constantly attack the network for economic benefits [6]. To ensure the security of cloud space, intrusion detection technology is needed to secure outsourced data traffic [7]. As a defense means, network traffic detection technology can identify the abnormal data in the byte stream, so that the system managers can find the attack behaviors in time, and then take corresponding measures to resist them and therefore reduce the loss. While early intrusion detection techniques relied on manual extraction of traffic features, not only the chosen algorithm but the pre-defined set of features could make a significant influence on its recognition accuracy [8, 9]. So far, with the development of deep learning, especially the increasing maturity of convolutional neural networks (CNN)

on image recognition, many researchers have introduced neural networks into the field of network traffic detection, and have made numerous achievements. However, following problems can be generally found in existing researches. First, the datasets used by many researchers are too old [10]. In recent years, new types of attack have been emerging, but many public datasets are not adequate for the current cyberspace, either in terms of variety or quantity. Second, many datasets are processed data that has lost the full information of the original byte flow [11], resulting in solution's failure to simulate real detection environment. Third, many researchers have ignored the imbalanced distribution of different kinds of abnormal categories in the dataset [12]. As a result, models with high overall accuracy can be quite inaccurate when it comes to small samples.

Notably, data imbalance is an important factor for machine learning [13]. In the field of intrusion detection, different anomalous flows vary greatly in terms of structure, behavior, etc. For example, attacks such as port scanning [14] or DoS [15] are easy to be detected and captured, so such anomalous flows often take up a large portion of the byte stream. In contrast, some complex attack types, such as APT [16], that are difficult to be collected only account for a small proportion in the dataset. To solve the impact of imbalanced data distribution on recognition accuracy, this paper chooses to use the original byte traffic as input directly. The flow form composed of the same five-tuple features [17] (i.e., source IP address, destination IP address, source port, destination port, and protocol) maximally preserves the structure and spatial features of each abnormal flow itself. These features are then expressed by a traffic grayscale map, thus, with the help of a CNN-based neural network, different abnormal categories can be identified through the automatic extraction of them. Finally, we choose CICIDS2017 [10], which provides raw byte stream files and contains many different types of attack flow, as the dataset for our experiments. Statistically, the percentage of different malicious flow types varies greatly, also, the attack categories match the realistic cyberspace environment and thus fit the scope of our study.

Many studies at this stage often use a single-path CNN network structure [18], which often fails to extract enough features from the grayscale map. Besides, simple increasement of the network depth may lead to the downgrade of model accuracy. In order to improve the generalization ability of the model and to deal with gradient disappearance, this paper proposes a ResNet&Inception convolutional neural network (RICNN) by combining the advantages of residual networks (ResNet) [19] and Inception feature fusion [20]. RICNN adopts parallel structures for feature extraction of the input to obtain more feature maps in the form of feature fusion. And direct mapping via ResNet [19] will be used to solve the problem of gradient disappearance during learning process. The final experimental results show that RICNN achieves a recognition accuracy of over 99% on the dataset CICIDS2017 and has high recognition rate on small samples categories.

The main contributions of this paper are as follows:

(1) A new network model, RICNN, is proposed, which can effectively improve the recognition precision of small samples in multi-classification detection of abnormal traffic.

(2) An improvement model, ICNN, is proposed, which achieves the purpose of simplifying the network structure and does not affect the detection accuracy.

(3) The experimental results on the dataset CICIDS2017 show that RICNN not only has the overall accuracy of 99.386%, but also has high detection precision in different

categories, especially those with small samples. The comparison experiments show that the recognition rate of RICNN outperforms a variety of CNN models and RNN models, which can achieve the highest detection accuracy.

The rest of this paper is organized as follows. Section 2 discusses related works in this field. Section 3 describes the data pre-processing steps and the specific structure of the proposed model in this paper. Section 4 conducts an experimental comparison of our proposed model on CICIDS2017 to assess the effectiveness of our model. Finally, we draw a conclusion in Section 5.

2. Related Work

Current supervised learning models for classifying abnormal traffic are mainly divided into using CNN models to extract spatial features and using RNN models to extract timing features [18]. Most studies, though can obtain high detection accuracy, fail to take into account the impact of the imbalanced distributed dataset on the classification results.

Marín, G et al. [21] investigated the accuracy of deep learning models for recognition at the packet level and the flow level. The results showed that the recognition rate was higher for inputs in flow form. However, the authors did not consider the impact of small sample factors on the model, and also the experiments had a small variety of malicious samples. And many researchers raised the detection accuracy by improving the CNN model. Jing Ran et al. [22] first used 3-dimensional CNN networks for network traffic classification. The authors applied video analysis to traffic recognition by processing vector time series and one-hot coding to form a fixed-length 3-dimensional input, which was learned by a 3-dimensional CNN structure with features in time and space. The result showed that this model had a higher detection accuracy on multiple categories. Hyun-Kyo Lim et al. [23] introduced the residual structure to the CNN network. After experimental comparison, it was found that the ResNet model could improve the generalization ability of the network. When the input data contained more information, the ResNet model had higher accuracy. Wei Wang et al. [24] proposed to use a 1-dimensional CNN model to classify encrypted traffic, and experiments proved that 1D-CNN performed better in end-to-end encrypted traffic recognition. Yong Zhang et al. [25] proposed a parallel CNN structure, PCCN, which improved the generalization ability of the network through feature fusion without increasing the depth of the network. Peng Yujie et al. [26] proposed a 1.5D-CNN model that combined 2D-CNN and 1D-CNN to extract features of traffic in different dimensions, and its multi-classification accuracy reached 98.5%. Samson Ho et al. [27] used a modified version of LeNet which did not take into account the imbalanced dataset, so the model did not have a high detection rate for small samples at multiple classifications. In the same way, researchers have focused on the impact of hybrid models on anomaly detection. Manuel Lopez-Martin et al. [28] studied the influence of timing features in the network stream on the classification detection results. The use of CNN and LSTM for vector time series data reduced the impact of feature engineering, but experiments showed that the model's recognition accuracy was still low for certain packet types. Monika Roopak et al. [29] proposed a model that combined RNN and CNN to detect DDoS attacks on IoT networks. Although its detection accuracy reached 97.16%, the author only considered DDoS attack, and did not verify the detection accuracy of the model under multiple attacks. Jiayin Feng et al. [30] proposed a model combining cas-

cades CNN and autoencoder for mobile terminal intrusion detection to classify mobile traffic in a semi-supervised form. But the model performed poorly in the case of multiple classifications. Khan, M. A. et al. [31] proposed a hybrid model of 1D-CNN and two-layer LSTM and achieved 97.29% accuracy on the dataset ISCX2012 [32]. Pengfei Sun et al. [33] also used a hybrid model of CNN and LSTM, and eliminated the effect of sample category imbalance on the model by weight optimization. The accuracy of the model in multiple classifications reached 98.67%. Kaiyuan Jiang et al. [12] used a hybrid sampling method to solve the dataset imbalance problem by reducing the noise in large samples through one-sided selection (OSS), and increasing small samples by using the synthetic minority oversampling technique (SMOTE) to finally build a relatively balanced dataset. However, the authors' hybrid model using CNN-BiLSTM failed to achieve the required recognition accuracy on the dataset. Maonan Wang et al. [34] combined CNN and stacked autoencoder (SAE), with CNN automatically extracting high-level features from the original traffic, SAE encoding 26 statistical features, and finally merging them into new high-level features. The framework achieved 98% accuracy in the multi-classification of encrypted traffic, but it took a lot of time to perform feature statistics on the original flow, which required high labor costs. Wanqian Zhang et al. [35] explored the relationship between window size and model classification accuracy. By detecting CPU occupancy in real time and finding appropriate dynamic parameters, the recognition time of CNN model was reduced, and a new framework of online traffic detection technology was realized. Chongzhen Zhang et al. [36] proposed a general intrusion detection framework that used an unsupervised autoencoder for feature extraction, and the extracted low-dimensional recombined features were stored for testing and retraining. However, the results showed that the recognition rate of small samples was lower than other sample categories in multi-classification.

Compared with the above schemes, our model has the following advantages. Our model has higher accuracy. And under the condition of imbalance distribution of data samples, the detection effect of all malicious categories is optimal. Our model structure does not deepen the number of network layers, but uses multiple branches to process the features. In terms of data processing, we combine the pre-processing methods of [11] and [25] to segment the raw traffic into flow format and form fixed-length input samples. Two types of training samples are generated, one is to extract the header of the packet and the other is to extract the payload of the packet, and they share the same length. Instead of deepening the layers of the network to improve the detection accuracy on small sample categories, we improve the parallel units of [25] and introduce ResNet to enhance the generalization ability of the model.

3. Model and Method

In this section, we design a hybrid neural network model to implement the detection of abnormal traffic. The proposed model primarily combines two convolutional neural networks, ResNet and Inception, and uses feature fusion to achieve accurate recognition of small samples. Our model uses raw traffic as input and automatically extracts the original features from abnormal traffic to complete the multi-classification. For the model to better learn the spatial features of the original traffic, we transform the samples into two-

dimensional grayscale maps. First, we will introduce the pre-processing process for the dataset.

3.1. Data Preprocessing

In this paper, the original traffic file is processed directly to obtain sample data in the flow format with a fixed length of 256 bytes. The process is as follows.

(1) **Raw file processing.** The dataset is a large PCAP file. We use the SplitCap tool [37] to merge packets with the same five tuples into a single flow. We limit the number of packets in a flow to five, and when the number is over five, we form a new flow where vacancies are filled with 0 bytes.

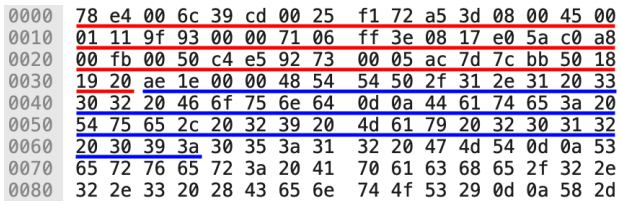


Fig. 1. Packet interception. The first 50 bytes are header features (red underline), and the 51st-100th bytes are payload features (blue underline)

(2) **Packet processing.** As shown in Fig. 1, to make the final training samples in the same length, the packets need to be intercepted and padded. The first 50 bytes of the packet are used as the header features and the 51st to 100th bytes are used as the payload features, with any shortfall in length being padded with "0". Statistical analysis shows that a length of 50 bytes can contain most of the traffic features [11]. The number "256" is used to divide each packet in a flow sample, as it equals to 0 when being transformed into a grayscale map, it will not affect the original features of the flow.

(3) **Grayscale map conversion.** Since the same type of network attack flow has a similar structure [38], and the original bytes of the packets take values within 0-255, the data can therefore be converted into a grayscale map. In this way, the classification of different malicious flows can be realized by transforming the process of extracting traffic structure features into that of extracting texture features from grayscale maps. In this paper, the flow samples with 256 bytes in length are converted to 16*16 grayscale maps.

3.2. Model Design

Because the various types of samples in the dataset are imbalanced distribution, we choose CNN to extract features from the samples to improve the recognition accuracy for small samples. The CNN-based models have good recognition capability in image classification [39]. In this paper, we propose a RICNN model combining ResNet and Inception to improve the detection rate of abnormal traffic with imbalanced data. As shown in Fig. 2,

the proposed model is mainly composed of three residual blocks, each of which is an Inception structure that learns more spatial features from the samples through feature fusion to achieve the detection of traffic categories.

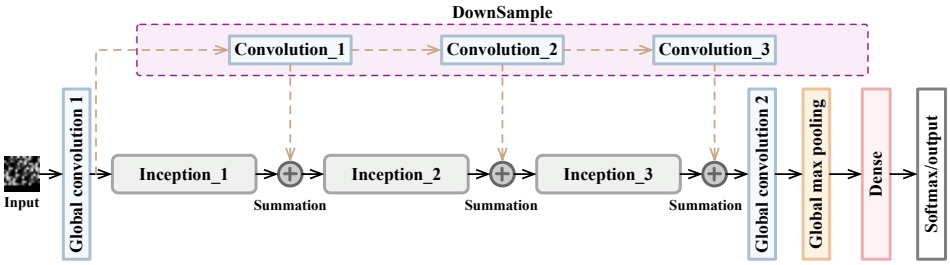


Fig. 2. RICNN network model architecture. Its input is a 16×16 grayscale matrix. The main structure of RICNN is three residual blocks, each of which consists of an Inception unit and a direct mapping. In addition, there are several global network layers to increase and reduce the dimension of the feature maps, and finally the model prediction results are output through the Softmax layer

(1) Inception Unit

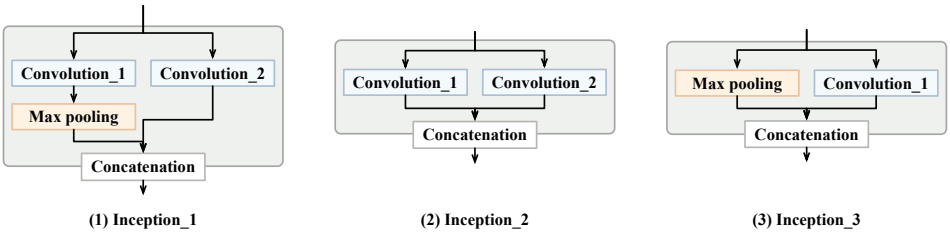


Fig. 3. Details of three Inception units

Inception is a CNN functional unit proposed by Google [20], which sets up different branching structures in the same block. Each branch extracts features from the original maps in parallel and extracting more features with different receptive field sizes without increasing the depth of the network. For a feature map X, its dimension is $H \times W \times C$. Assuming that a Inception unit has n branches, the height, width, and channel number of the output feature maps of i-th branch are h, w and c_i, then the final concatenation operation of each Inception is as follows:

$$H_{out} \times W_{out} \times C_{out} = h \times w \times \sum_{i=1}^{n} c_i \tag{1}$$

Table 1. Parameters of the Inception units

Name	Branch	Operation	Input Size	Convolution Kernel	Step	Padding	Output Size
Inception_1	Branch_1	Conv	16*16*16	3*3	2	1	8*8*32
	Branch_2	Conv	16*16*16	3*3	1	1	16*16*32
		MaxPool	16*16*32	2*2	2	0	8*8*32
Inception_2	Branch_1	Conv	8*8*64	3*3	1	1	8*8*96
	Branch_2	Conv	8*8*64	3*3	1	1	8*8*96
Inception_3	Branch_1	Conv	8*8*192	3*3	2	1	4*4*256
	Branch_2	MaxPool	8*8*192	2*2	2	0	4*4*192

Tab. 1 shows the structural parameters of the three Inception units. As shown in Fig. 3, the branches of Inception consist of convolutional layers and maximum pooling layers. The first Inception unit has one convolutional layer in each of the two branches. In the first branch, the convolutional layer is responsible for sampling and dimension reduction, but the convolutional layer in the second branch is just responsible for sampling, and the maximum pooling layer performs the dimension reduction and preserves the texture features of the grayscale maps. The second Inception unit consists of two convolutional layers, which form a parallel branch structure. The first branch in the third Inception unit is a convolutional layer, which is responsible for sampling and dimension reduction, and the second branch is a maximum pooling layer for texture reduction and preservation.

(2) ResNet Unit

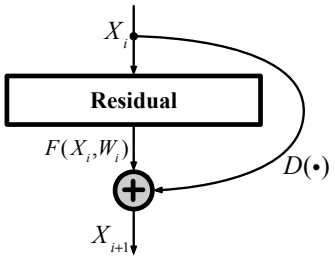


Fig. 4. Residual block structure

The residual network is designed to solve the problem of vanishing gradient and accuracy degradation in deep networks so that each layer of the network can fully learn the original traffic features and improve the classification accuracy of malicious traffic. As shown in Fig. 4, a residual block mainly consists of a residual part and a direct mapping. For an input feature map X_i, its residual part is $F(x_i, W_i)$, and W_i denotes the set of convolution operations. The direct mapping is:

$$D(x) = w' * x \tag{2}$$

w' represents a 1*1 convolution operation, which reduces the dimension of the input feature map to be consistent with the output dimension of the residual part. The final output of the residual block is:

$$X_{i+1} = \lambda_{relu}(D(X_i) + F(X_i, W_i)) \tag{3}$$

λ_{relu} represents the activation function ReLU, which can improve the nonlinear ability of the network. Through direct mapping $D(X_i)$, we can solve the problem of network learning difficulties. The residual part $F(X_i, W_i)$ of the proposed model is composed of Inception and three direct mapping branches. Each branch uses a 1*1 convolution layer to increase and reduce the dimension of the original feature maps, so that the output is consistent with the residual part.

(3) Overall Model Architecture

The proposed model consists mainly of convolutional layers and maximum pooling layers. The convolution layers are used for feature sampling on the input maps and expanding the dimension of the output maps. A convolution kernel ω with the size of $f * f$ is sampled on the input map X_i, and the output feature map is:

$$X_{i+1} = \lambda_{relu}(BN(\omega * X_i)) \tag{4}$$

For each batch, all feature maps need to be batch normalization (BN) before nonlinear activation. First, we normalize the feature data with each mini-batch as a unit:

$$\begin{aligned} X_i &= \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \\ &= \frac{x_i - \frac{1}{m}\sum_{i=1}^{m} x_i}{\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(x_i - \frac{1}{m}\sum_{i=1}^{m} x_i\right)^2 + \varepsilon}} \end{aligned} \tag{5}$$

Then move and scale the feature, that is:

$$BN_{\gamma,\beta}(X) = \gamma X + \beta \tag{6}$$

Batch normalization improves generalization ability of the network and increases the accuracy of learning, which also has the effect of preventing overfitting [40].

Maximum pooling is used for the pooling layer. The recognition of the traffic grayscale map mainly relies on the learned texture features, that the texture composed of light-colored pixel units, while weakening the dark part of the grayscale map. Therefore, when down-sampling the feature map, maximum pooling can preserve the texture features very well and reduce the loss during the learning process.

Tab. 2 shows the other parameters of the proposed model. The original traffic map is fed into the first global convolutional layer to obtain initial sampled feature maps with an expanded number of channels. The feature maps are then fed into the residual block structure, where the different features are extracted by the Inception unit in parallel, and the accuracy degradation problem is eliminated by the direct mapping. The second global convolution of the model is the expansion of the feature maps, and the global pooling layer reduces the dimension of the feature maps and the number of parameters. The final

Table 2. Parameters of the overall model

Operation	Input Size	Convolution kernel	Step	Padding	Output Size
Global Convolution 1	16*16*1	3*3	1	1	16*16*16
Global Convolution 2	4*4*448	3*3	1	1	4*4*896
Global Max Pooling	4*4*896	4*4	1	0	1*1*896
Dense	896	-	-	-	12

fully-connected and softmax layer map the final extracted high-dimensional features into specific categories, enabling multiple classification of malicious traffic.

3.3. ICNN

To simplify the network structure, we propose ICNN, an improved version based on the model in this paper. As shown in Fig. 5, in the improved version, we remove the residual block and add a 1*1 convolution layer to each Inception unit instead of direct mapping, and perform features fusion with other branches to achieve improved network generalization. Simultaneously, we abandon a global convolution network layer, which reduces the number of learning parameters.

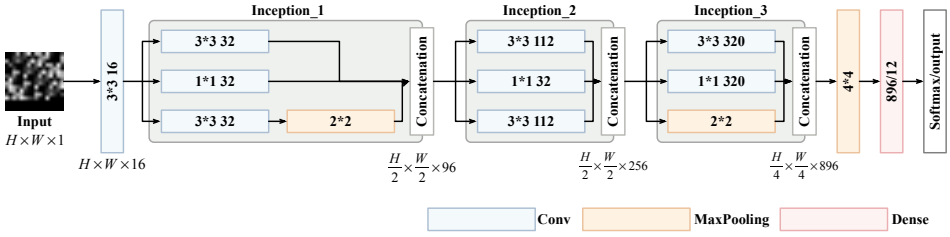


Fig. 5. Improved model based on Inception (ICNN)

4. Experimental Evaluation

In this section, we perform an experimental validation of the proposed model and test the validity of our model through contrast experiments. We choose CICIDS2017 as the dataset for our experiments. Our experimental environment is shown in Tab. 3.

4.1. Dataset

The dataset chosen for this paper needs to have the following characteristics: (I) It is the original traffic; (II) It contains various types of attack; (III) The distribution of all kinds of traffic is imbalanced; (IV) It is a relatively new dataset. Nowadays, many datasets are too

Table 3. Experimental environment parameters.

Name	Parameters
CPU	Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz
GPU	NVIDIA Quadro P4000
RAM	64GB
OS	Ubuntu 16.04

old and lack of new attack types. Similarly, some datasets only contain partial features of packets, and lack of complete traffic data. After comprehensive consideration, we choose CICIDS2017 [10] as the experimental dataset for this paper.

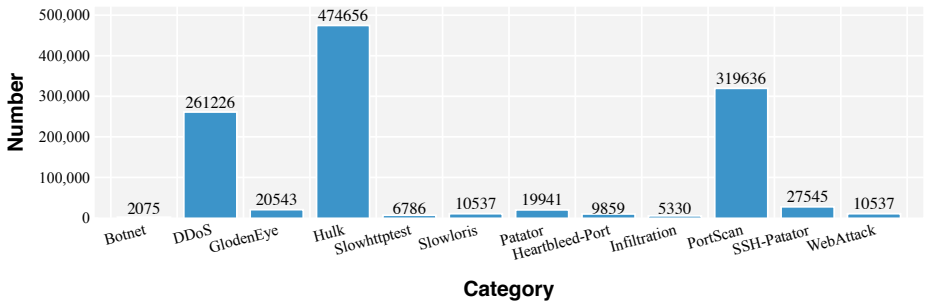


Fig. 6. Statistics on the number of various attack types in the CICIDS2017 dataset

CICIDS2017 is an open source network intrusion detection dataset, which is composed of traffic data collected by Canadian Network Security Agency over five consecutive days in 2017, and has completely marked various types of traffic. According to the statistical analysis, the dataset contains a total of 12 different types of attack traffic, and the number of different attacks is imbalanced. As shown in Fig. 6, Hulk, DDoS, and PortScan account for 90% of the overall dataset, while types such as Botnet and Infiltration make up much small percentage of the attacks. We divide the CICIDS2017 dataset, 80% of which are used as training set and 20% as testing set. To make each category equally distributed in the training set and testing set, we need to divide each type with a 4:1 ratio.

4.2. Evaluation Indicators

To analyze the detection accuracy of the model for each category of abnormal traffic, we choose Accuracy Rate (Acc) as the indicator of the overall model and Precision, Recall and F1 values as the recognition indicators for each category. For the category i of abnormal traffic, the following four indicators can be calculated:

True Positives (TP_i): The predicted category is i, and the true category is i as well.

False Positives (FP_i): The predicted category is i, but the true category is not i.

False Negatives (FN_i): The predicted category is not i, but the true one is i.

Then, for the model with n-classification, the calculation formula of Acc, which represents the overall recognition accuracy of the model, is as follows:

$$Acc = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)} \times 100\% \tag{7}$$

For the evaluation indicators Precision and Recall for category i, the formulae are:

$$P_i = \frac{TP_i}{TP_i + FP_i} \times 100\% \tag{8}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \times 100\% \tag{9}$$

The F1 indicator for category i is calculated as:

$$F1_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \times 100\% \tag{10}$$

Accuracy (Acc) is the ratio of the number of correctly identified samples to the overall samples. It measures the general classification effect of the model, but it is not specific to the recognition accuracy of a certain category. For this paper, it is important to focus not only on the general classification effect, but also on the identification of specific categories. In contrast, Precision, Recall and F1 focus more attention on evaluating the detection effectiveness of the model on different categories. The use of the above indicators provide a comprehensive and realistic assessment of our model.

4.3. Result Analysis

To investigate the detection capability of the proposed model on the imbalanced abnormal traffic dataset, we compare the performance indicators of it with other CNN models. We conduct experiments on the header and payload of samples to investigate the recognition ability of the proposed model on different segments of raw traffic features. And to investigate the effectiveness of our model in extracting features directly on the original traffic, we also compare it with LSTM and CNN+LSTM models that identify the timing properties of the samples. Furthermore, in order to simplify the proposed network, we propose an improved network ICNN that uses only the Inception structure to achieve extremely high detection rates and improve the operational efficiency of the model.

(1) Experimental content

Table 4. Variation of learning rate with epoch

Epoch	1-5	6-8	9-10
Learning Rate	0.0001	0.00001	0.000001

In the training phase, we set the epoch to 10 and the mini-batch for each round is 256. Tab. 4 shows the setting of the learning rate for different epoches. At the same time, after each epoch of training, we use the testing set to test the model and obtain the actual accuracy in the process of model learning.

(2) Analysis of experimental results

We use the proposed RICNN model in this paper as the basis for performance comparisons with other network models, and also compare the experimental performance of ICNN. For other CNN models, we choose 1d-CNN [24], 2d-CNN [27] and PCCN [25]. Meanwhile, we choose RNN models like LSTM [41,42] and CNN+LSTM [28,29,33] to compare their effectiveness in extracting timing features from the original traffic samples.

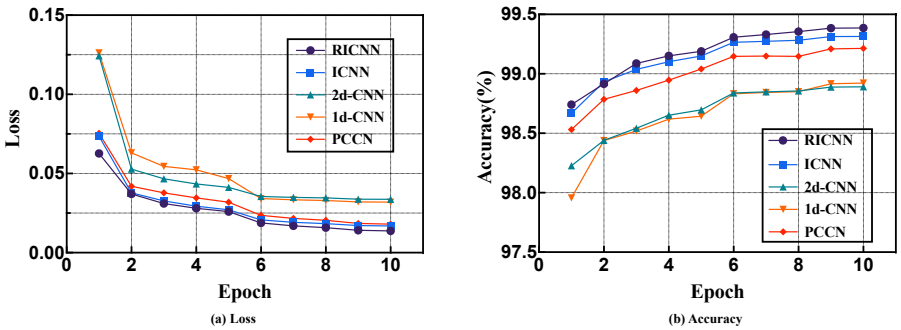


Fig. 7. Comparison of five CNN models (header). (a) indicates the variation of loss with epoch, and (b) indicates the variation of accuracy with epoch

To show the ability of RICNN in extracting the texture features from maps and obtaining the spatial features of the original traffic, three different CNN models are used to compare the detection accuracy of the 12-classification of abnormal traffic samples (header) with RICNN and ICNN. Fig. 7 (a) shows the variation of loss value with epoch in the process of model learning, and Fig. 7 (b) shows the change of accuracy. It can be seen that RICNN, ICNN, and PCCN which includes parallel feature extraction branches are more than 0.3% higher than the 1d-CNN and 2d-CNN with a single extraction path in terms of abnormal traffic detection accuracy. Due to the improved generalization of the network by direct mapping, RICNN and ICNN had a 0.17% higher detection accuracy than PCCN.

Raw traffic data holds the most complete flow features, and feature learning directly in the raw traffic can improve the accuracy of classification. LSTM can extract the timing features of the samples to distinguish between different malicious traffic. However, the comparison of the indicators in Fig. 8 shows that in BotNet, Goldeneye, Slowhttp, Infiltration and Web Attack, the number of samples is so small that the recognition accuracy in these categories is 0, that is, LSTM does not learn the correct features at all. The CNN+LSTM hybrid model has improved the recognition accuracy of the above abnormal traffic types (except BotNet) because of the spatial features extracted by CNN.

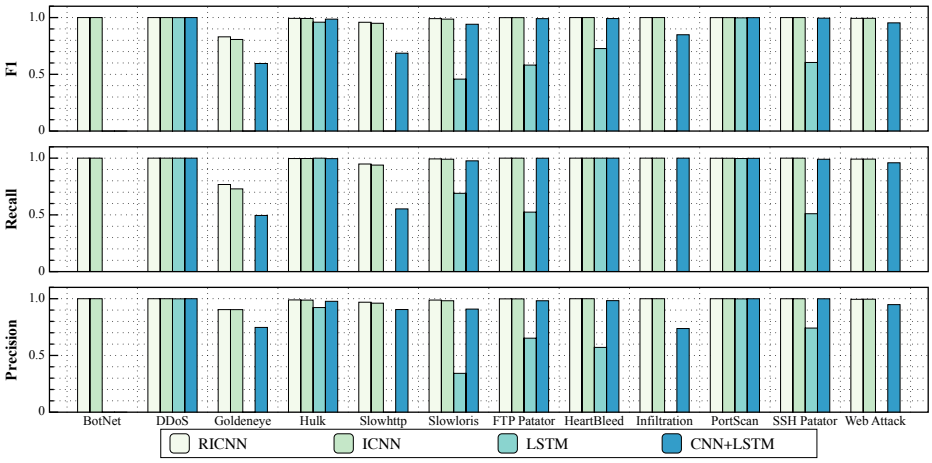


Fig. 8. Comparison of evaluation indicators between the proposed models and the RNN models for imbalanced abnormal traffic (header)

The above comparative experiments show that the original traffic needs to be specifically encoded [43, 44], and the RNN models can extract complete timing features. However, encoding also irreversibly corrupts the original traffic similar to hand-extracted features and is therefore less effective in anomaly detection for small samples.

Table 5. Precision of 12 categories of imbalanced traffic (payload). Numbers 1-12 respectively indicate: Botnet, DDoS, GlodenEye, Hulk, Slowhttptest, Slowloris, Patator, Heartbleed-Port, Infiltration, PortScan, SSH-Patator, WebAttack

Model	1	2	3	4	5	6	7	8	9	10	11	12
RICNN	0.9787	0.9963	0.9976	1.0000	0.9978	0.9915	1.0000	0.9980	0.9467	1.0000	0.9993	0.9995
ICNN	0.9921	0.9959	0.9973	0.9999	0.9985	0.9920	1.0000	0.9990	0.9605	1.0000	0.9996	0.9995
PCCN	0.9814	0.9965	0.9968	0.9999	0.9985	0.9925	1.0000	0.9995	0.9542	1.0000	0.9993	0.9995

Finally, we investigate the effect of the payload part on the detection accuracy of the model when it is used as training data. The header part of the packet mainly contains header information, including protocol, address, etc., and its structure is relatively fixed. Furthermore, the payload part contains the application layer data of the packet, which represents the real information and better expresses the feature information of the abnormal traffic. We choose PCCN, which also has parallel structures, for comparison with the two proposed models. Tab. 5-7 show the Precision, Recall and F1 score of the three models in different abnormal classes. The tables show that even the BotNet and Infiltration categories, which have the smallest number of samples, have improved recognition rates, indicating that RICNN and ICNN can have higher recognition rates on the pay-

Table 6. Recall of 12 categories of imbalanced traffic (payload). The category number is consistent with Tab 5

Model	1	2	3	4	5	6	7	8	9	10	11	12
RICNN	0.8867	0.9997	0.9998	0.9985	0.9816	0.9991	0.9992	0.9995	0.9493	0.9999	0.9989	0.9995
ICNN	0.9108	0.9998	0.9976	0.9985	0.9838	0.9995	0.9992	0.9995	0.9343	0.9999	0.9995	0.9991
PCCN	0.8916	0.9998	0.9983	0.9985	0.9816	0.9991	0.9992	0.9990	0.9568	0.9999	0.9998	0.9995

Table 7. F1-score of 12 categories of imbalanced traffic (payload). The category number is consistent with Tab 5

Model	1	2	3	4	5	6	7	8	9	10	11	12
RICNN	0.9305	0.9980	0.9987	0.9993	0.9896	0.9953	0.9996	0.9987	0.9480	1.0000	0.9991	0.9995
ICNN	0.9497	0.9979	0.9974	0.9992	0.9911	0.9957	0.9996	0.9992	0.9472	0.9999	0.9995	0.9993
PCCN	0.9343	0.9981	0.9976	0.9992	0.9900	0.9957	0.9996	0.9992	0.9555	1.0000	0.9995	0.9995

load dataset. Fig. 9 shows the variation of the three models with epoch. It can be seen that ICNN learns features quickly and achieves higher accuracy, while RICNN is less accurate than ICNN and PCCN in the first few epoches due to the more complex network structure. Finally, at the 10th epoch, all three models achieved a detection accuracy of 99.87%. With limited resources, ICNN has more advantages.

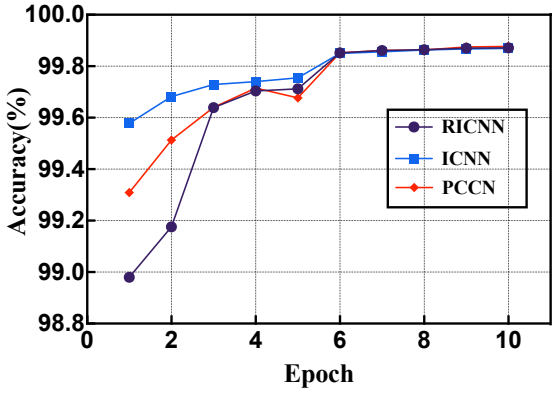


Fig. 9. Accuracy comparison of three parallel CNN multi-classification models (payload)

5. Conclusion

We want to retain the maximal amount of raw traffic features, so the raw packets are cut into header and payload parts of a specific length and then combined into separate flow

forms as input data. This paper proposes a convolutional neural network based on ResNet and Inception. Through three times of feature fusion and direct mapping, the detection accuracy of imbalanced abnormal samples is improved without increasing the depth of the network. The experimental results show that the proposed model can detect abnormal classes of small samples well on the CICIDS2017 dataset. We not only compare experimentally with other CNN models, but also compare the feature extraction of RNN models on raw traffic. The experimental results show that our models all outperform the other models. Finally, we find that the model has a higher detection accuracy on the payload feature set than the header.

In the future, we will try to use deep learning algorithms to detect unknown types of attacks. In addition, we would like to introduce recurrent neural network models and unsupervised models to mine the temporal and unknown features present in the traffic. Thus, we can improve the detection capability of real-time and persistent attack traffic to keep pace with the development of cyber environment and to improve cyber security in cloud computing scenarios.

Acknowledgments. This research is supported by the National Natura Science Foundation of China under Grant 61873160, Grant 61672338 and Natural Science Foundation of Shanghai under Grant 21ZR1426500.

References

1. Han, D., Pan, N., Li, K.C.: A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. IEEE Transactions on Dependable and Secure Computing pp. 1–1 (2020)
2. Cui, M., Han, D., Wang, J.: An efficient and safe road condition monitoring authentication scheme based on fog computing. IEEE Internet of Things Journal 6(5), 9076–9084 (2019)
3. Cui, M., Han, D., Wang, J., Li, K.C., Chang, C.C.: Arfv: An efficient shared data auditing scheme supporting revocation for fog-assisted vehicular ad-hoc networks. IEEE Transactions on Vehicular Technology 69(12), 15815–15827 (2020)
4. Xiao, T., Han, D., He, J., Li, K.C., de Mello, R.F.: Multi-keyword ranked search based on mapping set matching in cloud ciphertext storage system. Connection Science 33(1), 95–112 (2021)
5. Tian, Q., Han, D., Jiang, Y.: Hierarchical authority based weighted attribute encryption scheme. Computer Science and Information Systems 16(3), 797–813 (2019)
6. Kilincer, I.F., Ertam, F., Sengur, A.: Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks 188, 107840 (2021)
7. Liu, H., Han, D., Li, D.: Behavior analysis and blockchain based trust management in vanets. Journal of Parallel and Distributed Computing 151, 61–69 (2021)
8. Tian, Q., Han, D., Li, K., Liu, X., Duan, L., Castiglione, A.: An intrusion detection approach based on improved deep belief network. Applied Intelligence 50(10), 3162–3178 (2020)
9. Xu, J., Han, D., Li, K., Jiang, H.: A k-means algorithm based on characteristics of density applied to network intrusion detection. Computer Science and Information Systems 17(2), 665–687 (2020)
10. Sharafaldin., I., Habibi Lashkari., A., Ghorbani., A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP,. pp. 108–116. INSTICC, SciTePress (2018)

11. Zhang, Y., Chen, X., Jin, L., Wang, X., Guo, D.: Network intrusion detection: Based on deep hierarchical network and original flow data. IEEE Access 7, 37004–37016 (2019)
12. Jiang, K., Wang, W., Wang, A., Wu, H.: Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access 8, 32464–32476 (2020)
13. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent data analysis 6(5), 429–449 (2002)
14. Bailey-Lee, C., Roedel, C., Silenok, E.: Detection and characterization of port scan attacks. Univeristy of California, Department of Computer Science and Engineering pp. 1–7 (2003)
15. Bhuyan, M.H., Kashyap, H.J., Bhattacharyya, D.K., Kalita, J.K.: Detecting distributed denial of service attacks: Methods, tools and future directions. The Computer Journal 57(4), 537–556 (2014)
16. Zhao, G., Xu, K., Xu, L., Wu, B.: Detecting apt malware infections based on malicious dns and traffic analysis. IEEE Access 3, 1132–1142 (2015)
17. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking (ICOIN). pp. 712–717 (2017)
18. Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A., Foozy, C.F.M.: Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset. IEEE Access 9, 22351–22370 (2021)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
20. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
21. Marín, G., Caasas, P., Capdehourat, G.: Deepmal-deep learning models for malware traffic detection and classification. In: Data Science–Analytics and Applications, pp. 105–112. Springer (2021)
22. Ran, J., Chen, Y., Li, S.: Three-dimensional convolutional neural network based traffic classification for wireless communications. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP). pp. 624–627 (2018)
23. Lim, H.K., Kim, J.B., Heo, J.S., Kim, K., Hong, Y.G., Han, Y.H.: Packet-based network traffic classification using deep learning. In: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). pp. 046–051 (2019)
24. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 43–48 (2017)
25. Zhang, Y., Chen, X., Guo, D., Song, M., Teng, Y., Wang, X.: Pccn: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows. IEEE Access 7, 119904–119916 (2019)
26. Yujie, P., Weina, N., Xiaosong, Z., Jie, Z., Wu, H., Ruidong, C.: End-to-end android malware classification based on pure traffic images. In: 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). pp. 240–245 (2020)
27. Ho, S., Jufout, S.A., Dajani, K., Mozumdar, M.: A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network. IEEE Open Journal of the Computer Society 2, 14–25 (2021)
28. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J.: Network traffic classifier with convolutional and recurrent neural networks for internet of things. IEEE Access 5, 18042–18050 (2017)

29. Roopak, M., Yun Tian, G., Chambers, J.: Deep learning models for cyber security in iot networks. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0452–0457 (2019)
30. Feng, J., Shen, L., Chen, Z., Wang, Y., Li, H.: A two-layer deep learning method for android malware detection using network traffic. IEEE Access 8, 125786–125796 (2020)
31. Khan, M.A., Karim, M.R., Kim, Y.: A scalable and hybrid intrusion detection system based on the convolutional-lstm network. Symmetry 11(4) (2019)
32. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security 31(3), 357–374 (2012)
33. Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., Chen, J.: Dl-ids: Extracting features using cnn-lstm hybrid network for intrusion detection system. Security and Communication Networks 2020 (2020)
34. Wang, M., Zheng, K., Luo, D., Yang, Y., Wang, X.: An encrypted traffic classification framework based on convolutional neural networks and stacked autoencoders. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC). pp. 634–641 (2020)
35. Zhang, W., Wang, J., Chen, S., Qi, H., Li, K.: A framework for resource-aware online traffic classification using cnn. In: Proceedings of the 14th International Conference on Future Internet Technologies. CFI'19, Association for Computing Machinery, New York, NY, USA (2019)
36. Zhang, C., Chen, Y., Meng, Y., Ruan, F., Chen, R., Li, Y., Yang, Y.: A novel framework design of network intrusion detection based on machine learning techniques. Security and Communication Networks 2021 (2021)
37. NETRESEC: Splitcap (2010), `https://www.netresec.com/index.ashx?page=SplitCap`
38. Chen, Z., He, K., Li, J., Geng, Y.: Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 1271–1276 (2017)
39. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition 77, 354–377 (2018)
40. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille, France (07–09 Jul 2015)
41. Azzouni, A., Pujolle, G.: A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv preprint arXiv:1705.05690 (2017)
42. Yuan, X., Li, C., Li, X.: Deepdefense: Identifying ddos attack via deep learning. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP). pp. 1–8 (2017)
43. Hwang, R.H., Peng, M.C., Nguyen, V.L., Chang, Y.L.: An lstm-based deep learning approach for classifying malicious traffic at the packet level. Applied Sciences 9(16) (2019)
44. Kim, A., Park, M., Lee, D.H.: Ai-ids: Application of deep learning to real-time web intrusion detection. IEEE Access 8, 70245–70261 (2020)

Benhui Xia received the B.S. degree from China University of Mining and Technology, where he is currently pursuing the M.S. degree with Shanghai Maritime University. His main research interests include network security, cloud computing, distributed computing and blockchain.

Dezhi Han received the Ph.D. degree from the Huazhong University of Science and Technology. He is currently a Professor of computer science and engineering with Shanghai

Maritime University. His research interests include cloud computing, mobile networking, wireless communication, and cloud security.

Ximing Yin received the M.S. degree from Zhejiang University, where he is currently pursuing the Ph.D. degree with East China University of Science and Technology. His main research interests include network security and wireless network security.

Na Gao received the B.S. degree from Shanxi Agricultural University of Software,where she is currently pursuing the M.S. degree with Shanghai Maritime University. Her main research interests are port supply chain applications and blockchain technology.

Received: June 17, 2021; Accepted: September 18, 2021.

Computer Science and Information Systems 19(1):327–352 https://doi.org/10.2298/CSIS210209056V

Hyper-parameter Optimization of Convolutional Neural Networks for Classifying COVID-19 X-ray Images*

Grega Vrbančič**, Špela Pečnik, and Vili Podgorelec

University of Maribor, Faculty of Electrical Engineering and Computer Science
Koroška cesta 46, SI-2000 Maribor, Slovenia
{grega.vrbancic, spela.pecnik, vili.podgorelec}@um.si

Abstract. For more than a year the COVID-19 epidemic is threatening people all over the world. Numerous researchers are looking for all possible insights into the new corona virus SARS-CoV-2. One of the possibilities is an in-depth analysis of X-ray images from COVID-19 patients, commonly conducted by a radiologist, which are due to high demand facing with overload. With the latest achievements in the field of deep learning, the approaches using transfer learning proved to be successful when tackling such problem. However, when utilizing deep learning methods, we are commonly facing the problem of hyper-parameter settings. In this research, we adapted and generalized transfer learning based classification method for detecting COVID-19 from X-ray images and employed different optimization algorithms for solving the task of hyper-parameter settings. Utilizing different optimization algorithms our method was evaluated on a dataset of 1446 X-ray images, with the overall accuracy of 84.44%, outperforming both conventional CNN method as well as the compared baseline transfer learning method. Besides quantitative analysis, we also conducted a qualitative in-depth analysis using the local interpretable model-agnostic explanations method and gain some in-depth view of COVID-19 characteristics and the predictive model perception.

Keywords: COVID-19, classification, CNN, transfer learning, optimization.

1. Introduction

Not much more than a year since December 2019, when in Wuhan city, the capital of Hubei province in China, the cases of "unknown viral pneumonia" started to gather, the world is witnessing a huge spread of coronavirus disease 2019 (COVID-19) caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Based on the World Health Organization report published on the 2nd of Februar 2021, there were more than 102 million confirmed cases and more than 2.2 million deaths globally, spreading across 220 countries and territories [34].

Currently, one of the mostly used method globally for detecting a COVID-19 disease is using the real-time transcription-polymerase chain reaction (RT-PCR) test [35]. However, the sensitivity of such method ranges around 70%, while the alternative methods using CT or X-ray imaging can achieve significantly better performance, up to 98% [14]. While such methods can provide us with better sensitivity performance, the main bottleneck is that analysing such imaging requires an experienced radiologist, who manually,

* This is an extended version of a INISTA 2020 conference paper.

** Corresponding author

visually scans such images trying to detect some pathology. This bottleneck especially in current situation comes to the fore, when a large number of such imaging should be analyzed in very short time, and thus increasing the probability of miss-classification and putting large amount of stress on medical staff. In those terms the use of advanced machine learning approaches for classification of images radiography imaging can be justified.

With the advancements of deep learning methods and techniques in recent years, especially the ones utilizing convolutional neural networks (CNNs), various research works proved that the application of such methods against the medical domain problems is resulting in encouraging results [25]. In the last year, there were large amount of researches published, focusing on applying the machine learning algorithms to identification of COVID-19. One of the most common approaches to tackle the mentioned issue is to utilize the transfer learning approach as presented in [2, 26].

While such approaches enable us to successfully train a predictive model, we are still faced with a major problem common to all training approaches of deep neural networks – setting the values of hyper-parameters [52] also known as hyper-parameter optimization (HPO). Setting the appropriate values of hyper-parameters for the process of training has a direct impact on the final predictive performance of such models, therefore the values should be carefully chosen. While commonly this is still a manual process, a great amount of research was put into developing automatic methods [38, 43, 49], which would take care of this problem. Since many studies have addressed the problem of identifying a COVID-19 from X-ray images and since the chosen hyper-parameter values have a direct impact on the final classification performance, it is crucial to set hyper-parameter values appropriately especially when addressing such sensitive problem.

Based on our previous experience with the identification of COVID-19 [43], promising results from similar studies [3, 36] and our previous work on solving HPO problem [38,43], we set our goal to generalize our GWOTLT [43] from our previous research, in which we utilized the grey wolf optimizer (GWO) algorithm to find the most suitable values of hyper-parameters, to make it agnostic to the usage of different optimization algorithms. Such a generalized HPO method for transfer learning (HPO-TL) enables us to employ various optimization algorithms in order to find the most suitable values of hyper-parameters in order to achieve the best possible predictive model utilizing transfer learning. Beside providing predictive model using the HPO-TL method and evaluating the performance of such models from a quantitative standpoint, we also conducted an analysis of interpretable representations of our model using local interpretable model-agnostic explanations (LIME) method. To gain useful insights on how the model perceives the chest X-ray images, evaluating the model's decisions from a qualitative perspective, we took a different approach where multiple interpretable representations obtained by LIME were aggregated into one single representation, which could enable us to gain different insights into perception of predictive model.

We can sum up our main contributions presented in this research as follows:

- We generalize GWOTLT method in order to make it optimization algorithm agnostic, which enables us to use various optimization algorithms for the task of hyper-parameter optimization.

- We conducted an empirical evaluation of the generalized HPO-TL method with three optimization algorithms (GWO, DE, GA), tackling the problem of detecting COVID-19 from X-ray images.
- We conducted an extensive performance analysis and comparison against the conventional approaches of training the predictive CNN model.
- We performed a qualitative analysis of the predictive model using the LIME method.

The remaining of the paper is structured as follows. In section 2, a brief review of related work is presented. Utilized methods and generalized HPO-TL method are presented in section 3, while in section 4 the experimental framework is described. In section 5 the results of conducted experiments are presented and interpreted, while section 6 presents the conclusions.

2. Related work

So far, many analyses have been performed using convolutional neural networks over chest X-ray images of patients, which try to help better identify COVID-19 cases. For example, Apostolopoulos and Mpesiana [2] were among the first to evaluate the performance of CNNs using transfer learning over a collection of images showing COVID-19 condition, pneumonia, or a normal condition. They found that in this way we could extract significant biomarkers related to the COVID-19 disease with great accuracy (above 96%). The use of eight different pre-trained CNNs over a dataset of normal and COVID-19 cases was also used in [32], where the authors report that the best model achieved an accuracy of up to 98%. Marques et al. [28] proposed a medical decision support system based on CNN with EfficientNet architecture. The built model was used for both binary classification and multiclass classification. In the case of binary classification, X-ray images of COVID-19 positive patients and healthy patients were used. For multiclass classification, images of patients with pneumonia were added to the dataset. The results showed that better values of different metrics are achieved in binary classification. S. Govindarajan and R. Swaminathan [18] acquired critical image features using CNN with several different hyper-parameter settings and cross-validation methods. They visualized them using occlusion sensitivity maps. The resulting images showed some localized abnormal regions, which indicate COVID-19. In [21], the authors conducted a study on images obtained from portable chest X-ray (pCXR), which included two types of pneumonia, the normal condition and the COVID-19 condition. CNN with transfer learning was used over whole pCXR and over segmented lungs. Better results were obtained over segmented lungs (accuracy 88%) than over whole pCXR (accuracy 79%). Majeed et al. [27] used 12 CNN architectures with transfer learning and a shallow CNN architecture which they trained from scratch. The X-ray images were also not preprocessed before the use. The parts of the images that were supposed to influence the decision of the model were visualized by class activation maps (CAMs), which, according to their findings, are not reliable, as they indicate parts that are not characteristic for COVID-19 disease.

As we can see, the use of CNNs with transfer learning is very common in this problem area. Differences can be found in the optimization of the algorithms, the parameter settings and used datasets. In the original, our research differs from the existing ones in that we used a dataset to predict COVID-19 status, which contains X-ray images of the chest

of COVID-19 patients and images showing a normal condition or any other respiratory disease. So our main purpose was to predict whether it is a COVID-19 case or some other condition.

3. Methods

Since the first introduction of CNNs in the 1980s [16], the remarkable progress has been made in the image recognition field especially due to the availability of large annotated datasets, development of various deep CNN architectures and increased computational capabilities. The CNNs or more precisely the convolutional layers leverage three important ideas that can help improve a machine learning system: sparse interaction, parameter sharing and equivariant representations. In contrast to the traditional neural network layers which use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit, the CNNs, however, typically have sparse interactions, also known as sparse connectivity or sparse weights. The sparse interactions are achieved by making a kernel smaller than the input, which on the one side enables us to detect small, meaningful features with kernels that occupy only tens of pixels, while on the other side reduces the memory consumption of the models and improves its statistical efficiency, since we need to store fewer parameters. Additionally, the use of parameter sharing in CNNs also increases the memory and statistical efficiency in comparison to the traditional neural network, where each element of the weight matrix is used exactly once when computing the output of a layer. Furthermore, in the case of convolution, the particular form of parameter sharing causes the layer to have a property called equivariance. Basically, equivariance enables convolution to create a 2-dimensional map of where certain features appear in the input. If the object in the input is moved, its representation will also move for the same amount [17].

Those capabilities make the CNNs de facto standard for solving the image recognition tasks in various domains from medicine [45, 48], information security [19] to seismology [22] or even agriculture [20]. However, training such CNN models requires a large amount of labeled data, which can be in certain fields, especially in medicine, a challenging task. To overcome the lack of sufficient labeled dataset, one of the commonly used methods is transfer learning with fine-tuning, which enables us to adapt a pre-trained model to our domain problem, without requiring a large dataset.

3.1. Transfer Learning

The first appearances of transfer learning in publications are dating back to the 1995 [6], mostly under different names such as inductive transfer [12], incremental or cumulative learning [55], and multitask learning [51], the latter one being the most closely related to the transfer learning as we know it today. In the most broader terms, the transfer learning technique can be defined as the improvement of learning a new task through the transfer of knowledge from a related task which has been already learned. However, in the machine learning terms, the transfer learning can be defined as transferring the weights of an already trained predictive model, specialized for a specific task, to the new model addressing similar but not the same task.

There are many different techniques on how to utilize the transfer learning, one of the most commonly used being the fine-tuning. When utilizing the fine-tuning approach to transfer learning, we are transferring the weights from a pre-trained CNN to the new one [41]. Commonly, we only transfer the weights in the so-called convolutional base of CNN architecture, which is composed of a sequence of convolutional layers and pooling layers, since those layers' weights contain general feature extraction capabilities. In general, the bottom layers (more towards the input) of the CNN tend to extract more abstract, generally applicable features than the top layers (more toward the output), which tend to extract more task-specific features. Therefore, when utilizing a fine-tuning technique, most commonly we only fine-tune (train) the layers more towards the top of the CNN architecture and leave the bottom ones frozen (disabled for training) [41].

Regardless of the benefits of the transfer learning with fine-tuning, such approach still has some challenges common to the traditional approach of training CNN. One of such problem is the selection of training parameters also known as hyper-parameters. Setting appropriate value for hyper-parameters such as learning rate, batch size, optimization function, etc. directly reflects on how well the model is capable to train and consequently impacts the model classification performance.

3.2. Hyper-parameter Optimization for Transfer Learning

The problem of setting the right values for the hyper-parameters is also known as hyper-parameter optimization (HPO) task. Most commonly are such tasks addressed with the Gaussian Process approach, Tree-structured Parzen Estimator approach or Random search approach [4]. But in recent years, population-based metaheuristic algorithms are becoming more and more popular in successfully solving HPO problems [23, 46, 49].

Based on our previous success with utilization of various optimization algorithms for the purpose of optimizing hyper-parameter values [38, 47], we decided to generalize our Grey Wolf Optimizer for Transfer Learning Tuning (GWOTLT) method presented in [49] to make it work and evaluate it with other popular metaheuristics, such as genetic algorithm (GA) and differential evolution (DE). The basic concept of our generalized Hyper-parameter optimization for transfer learning (HPO-TL) method can be generally defined in the following steps:

1. Optimization algorithm produces the solution.
2. Solution is mapped to the values of sought hyper-parameters.
3. CNN with mapped hyper-parameter values is trained.
4. The solution is evaluated, calculating fitness value.
5. Fitness value is being passed back to the optimization algorithm.

Those steps are then being executed in an iterative manner, for the given number of function evaluations.

The HPO-TL is producing a solution with the same number of elements as is the number of sought hyper-parameter values. In our case the dimension of the produced solution is 4, since we are searching for the most optimal value of four different hyper-parameters, namely: learning rate, optimizer function, dropout probability of dropout layer, and the number of neurons in the last fully-connected (dense) layer. Formally, the individuals of such HPO-TL produced solutions are presented as a real-valued vectors:

$$\mathbf{x}_i^{(t)} = (x_{i,0}^{(t)}, \ldots, x_{i,n}^{(t)}), \quad \text{for } i = 0, \ldots, Np - 1, \tag{1}$$

where each element of the solution is in the interval $x_{i,1}^{(t)} \in [0, 1]$. These real-valued vectors (solutions) are then mapped to the used hyper-parameter values as defined in equations 2 to 5, where y_1 denotes the number of neurons in the last fully connected layer, y_2 denotes dropout probability, y_3 denotes optimization function and y_4 denotes learning rate. Each y_1 value is mapped to the particular member of the population $N = \{64, 128, 256, 512, 1024\}$ according to the members position in the population, which represents a group of available numbers of neurons in the last fully connected layer. All of the y_3 values are mapped to the specific member of population $O = \{adam, rmsprop, sgd\}$, which represents a group of available optimizer functions, while each y_4 value is mapped to the member of population $L = \{0.001, 0.0005, 0.0001, 0.00005, 0.00001\}$, which represents a group of learning rate choices.

$$y_1 = \begin{cases} \lfloor x[i] * 5 + 1 \rfloor; y_1 \in [1, 5] & x[i] < 1 \\ 5 & \text{otherwise,} \end{cases} \tag{2}$$

$$y_2 = x[i] * (0.9 - 0.5) + 0.5; y_2 \in [0.5, 0.9] \tag{3}$$

$$y_3 = \begin{cases} \lfloor x[i] * 3 + 1 \rfloor; y_3 \in [1, 3] & x[i] < 1 \\ 3 & \text{otherwise,} \end{cases} \tag{4}$$

$$y_4 = \begin{cases} \lfloor x[i] * 5 + 1 \rfloor; y_4 \in [1, 5] & x[i] < 1 \\ 5 & \text{otherwise,} \end{cases} \tag{5}$$

The training utilizing the fine-tuning with mapped hyper-parameter values is then being conducted in a straight-forward manner where the last block of used CNN architecture is being fine-tuned while the other (more bottom) layers remain frozen. After the training is finished, the fitness values are being calculated. We defined the fitness value as:

$$f(sol) = 1 - AUC(sol) \tag{6}$$

where sol denotes the model trained with hyper-parameters set based on the obtained HPO-TL solution, and AUC defines the area under the ROC curve metric.

The fitness value is then being passed back to the chosen optimization algorithm, based on which the new solution will be produced.

3.3. HPO-TL variations

Since our presented HPO-TL method is designed to work with various optimization algorithms, we selected three of the most popular algorithms to showcase the advantages of such an approach where the method is not conceptually bonded to a particular algorithm. For this purpose, we utilized a grey wolf optimizer algorithm, which is in recent works [15, 43, 49] showing a great performance solving various tasks, differential evolution which is still dominating in various solutions [5, 9, 50], and one of most conventional nature-inspired evolutionary algorithms – genetic algorithm.

Grey Wolf Optimizer variation (GWO-TL) is based on the GWO algorithm [31], which is one of the most popular representatives of nature-inspired population-based metaheuristic algorithms. The GWO is inspired from a strict leadership hierarchy and hunting mechanisms of grey wolves (Canis lupus). As defined by authors in [31], there are three main phases of grey wolves hunting. First one is tracking, chasing and approaching the prey, the second one is pursuing, encircling and harassing the prey until it stops moving, and final third phase is the attack toward the prey. In GWO, we consider the fittest solution as the alpha, therefore consequently, the second and third-best solutions are named beta and delta. Other candidate solutions are assumed to be omega. In general, the search process starts by creating a random population of grey wolves in the GWO algorithm. In each iteration alpha, beta, and delta candidate solutions estimate the probable position of the prey. The parameter a is decreased from 2 to 0, to emphasize the exploration and exploitation, respectively. In each iteration the candidate solutions tend to converge to the prey when vector A, which is mathematically modeling divergence, is decreasing below 1 and diverge from the prey when A is increasing above 1 [31].

Differential Evolution variation (DE-TL) is based on arguably one of the most powerful and versatile evolutionary optimizers in recent times. Standard DE algorithm consists of four basic steps: initialization, mutation, recombination or crossover, and selection. From those steps, only the last three are repeated into the subsequent DE iterations [9]. In the initialization phase, Np real-value coded vectors are randomly initialized. Each such vector is also known as genome or chromosome and forms a candidate solution. After initialization, DE creates a donor or mutant vector corresponding to each population member in the current iteration with utilization of mutation. In order to increase the diversity of the parameter vectors of DE, the crossover step is conducted, where CR parameter controls the fraction of parameters that are copied to the candidate solution. Finally, the selection step is executed in which the decision whether a produced candidate solution should become a generation member is made, using the greedy criterion [9]. Those three steps are being repeated until stopping condition is not reached.

Genetic Algorithm variation (GA-TL) is based on the one of the first population-based stochastic algorithm. Similar to the other evolutionary algorithms, the main steps of the GA are selection, crossover, and mutation [30]. In the same manner as the DE, GA starts with a random population, which represents chromosomes of individual candidate solutions. Nature is the main inspiration for the selection step in GA algorithm, which is trying to mimic the phenomena where the fittest individuals have a higher chance of getting food and mating. For this purpose GA is employing a roulette wheel to assign probabilities to individuals and select them for creating the next-generation proportional to their objective. In the crossover step the selected individuals are being combined producing new solutions in GA algorithm. In the last step, mutation is conducted, in which one or multiple genes of created new solutions are altered. This step in GA maintains the diversity of population by introducing another level of randomness [30]. The algorithm iterates those three steps in the same manner until the stopping condition is not reached.

3.4. Local Interpretable Model-agnostic Explanations

Many times, in the world of machine learning, it is not enough just to build a good decision model, its success is also influenced by how decision makers understand and trust its predictions. This is especially important in more sensitive domains, such as medicine. Decision-makers' confidence in model results usually increases when they have a clear insight into what influenced the model's decision, what is its behavior and what are the possible errors. For this purpose, various interpretive methods have been developed. Some of them are also able to give an explanation for a model built over unstructured data (in our case images). [39].

The interpretive method we used in our case is the Local Interpretable Model-Agnostic Explanations (LIME) method, which was first introduced in 2016 by Ribeiro et al. [39]. An interpretive method that also allows the interpretation of models built above images is the SHapley Additive exPlanations (SHAP) method, which is, in addition to LIME, considered as one of the most widely used methods of this kind. LIME creates an explanation for an individual input prediction by sampling its neighboring inputs and builds a sparse linear model based on the predictions of these inputs. The most relevant features for a specific prediction are then those that have the highest coefficient calculated in this linear model [54]. One of the main advantages of the algorithm behind the LIME method is that it can explain the predictions of any black box classifier with two or more output classes. The condition for its operation is that the classifier implements a function that accepts a set of classes and then returns the probabilities for each class. The main goal of the algorithm is to identify an interpretive model over an interpretative representation that is locally faithful to the classifier. In our case or in general when working with images, the interpretative representation is a binary vector that indicates the presence or absence of neighboring sets of similar pixels, while the classifier can display the image as a tensor with three color channels per pixel.

As mentioned earlier, LIME explanation is based on the sampling of neighboring inputs of the selected input x and their outputs, while returning as a result a model g from the class of potential interpretive models G according to the following formula:

$$\arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g). \tag{7}$$

If we explain the formula in more detail, then we can say that x represents the input for which we want to know on the basis of which value was determined to belong to the selected class, f denotes the built model that we want to explain and π_x denotes the probability distribution around x. With $\Omega(g)$ we mark the complexity of model interpretation, that is opposite to its interpretability. Not every model is simple enough to be interpretive. The part of the equation $\mathcal{L}(f, g, \pi_x)$ tells us how the values of g approach the values of f at the location defined by π_x. If we want to achieve high interpretability, this value must be as low as possible [39].

4. Experiments

To objectively evaluate the COVID-19 image classification results, we conducted the following experiments:

- **base**, where the CNN is trained in a conventional manner without pre-training,
- **TL**, where transfer learning methodology is utilized, and
- three **HPO-TL** methods: TL-GWO, TL-GA, and TL-DE experiments where our proposed method is used.

All conducted experiments were implemented in Python programming language with the support of following libraries: scikit-learn [37], Pandas [29], Numpy [42], NiaPy [44], Keras [7] and Tensorflow [1].

Experiments were performed using the octa-core Intel CPU, 64 GB of RAM, and two Nvidia Tesla V100 GPUs each with dedicated 32 GB of memory.

4.1. Datasets

Almost a year after we published our previous work on COVID-19 [43], the COVID-19 dataset initially prepared by Cohen et al. [8] was greatly enlarged by various researchers from all over the world. Different contributors provided additional COVID-19 and other chest x-ray images, performed double-checking for potential labeling errors, and improved the dataset both in terms of quality and quantity. Therefore, for the purpose of evaluating the proposed methods, we obtained an updated version of the COVID-19 dataset, which in current state, on January 20th 2021, consists in total of 929 chest x-ray images.

Since the chest X-ray images are collected from various sources, the image size and format are varying. In Figure 1 are presented two samples from each of the target classes.

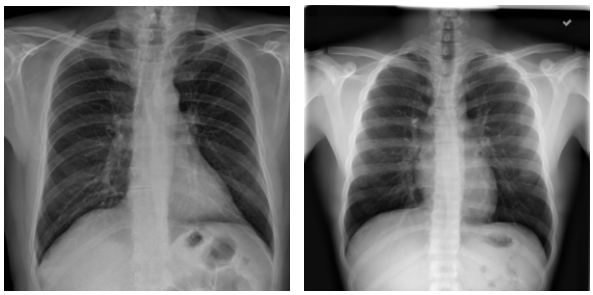


Fig. 1. Examples of X-ray images, where a) represents a COVID-19 case image, while b) represents an image with other or no pathology identified.

Inspecting the obtained dataset more in-depth, we can see that the majority of the collected chest x-ray images are labeled as an COVID-19 instances, as presented in Table 1. Comparing the number of classes in the updated version of COVID-19 dataset in comparison to the older version, we can see a significant increase. Additionally, in the updated version of the dataset we can see that one of the classes in labeled as ”todo”, which means that the instances with such label are not yet classified. Therefore, we removed instances with such label in order to avoid having some instances miss-classified and consequently

training the predictive model with wrong labeled chest x-ray images. This way we ended up with the total of 846 instances, 563 of them being labeled as ”COVID-19” and the remaining 283 labeled as ”other”.

Table 1. Target class distribution of updated COVID-19 image data collection.

Class	**COVID-19 image data collection**
COVID-19	563
Pneumonia	81
SARS	16
Pneumocystis	30
Streptococcus	22
No finding	22
Chlamydophila	3
E.Coli	4
Klebsiella	10
Legionella	10
Unknown	1
Lipoid	13
Varicella	6
Bacterial	4
Mycoplasma	11
Influenza	5
todo	83
Tuberculosis	18
H1N1	2
Aspergilliosis	2
Herpes	3
Aspiration	1
Nocardia	8
MERS-CoV	10
MRSA	1
Total	929

Similar as we did in our previous research [43], with the older version of COVID-19 dataset, we have extended the updated version. Additional 600 randomly selected ”Normal” labeled chest images from RSNA Pneumonia Detection Challenge [33] were added to the existing ”other” labeled chest x-ray images, which resulted in a final updated and extended version of COVID-19 dataset with properties presented in Table 2.

Table 2. Target class distribution of an updated and extended COVID-19 image data collection.

Class	Extended COVID-19 image data collection
COVID-19	563
Other	883
Total	1446

4.2. Data Pre-processing

As are the images in the COVID-19 image data collection in various sizes, we applied the image resizing to uniform target size of 224 x 224 pixels, which is in line with default input size of the selected VGG19 CNN architecture. Additionally, in the train time, we applied an image augmentation technique, to prevent the over-fitting which commonly occurs when dealing with pre-trained complex CNN architecture and relatively small datasets.

The image augmentation in train time is conducted in a manner where each training instance is randomly manipulated e.g. rotated, zoomed, shifted, flipped, etc. within the given value range. The complete list of utilized augmentation parameters and its values can be observed in Table 3. The value for rotation range specifies the degree range for random rotation, while the values for width shift and height shift range specifies the fraction of a total image size for corresponding dimension. Shear range value defines a shear intensity – the shear angle (in radians) in counter-clockwise direction and zoom range value specifies the randomly selected zoom between the lower and upper bounds defined as $1 - zoom\_range$ and $1 + zoom\_range$ respectively. Lastly, the horizontal flip value defines whether each image instance can be randomly flipped horizontally or not.

Table 3. Utilized image augmentation parameter settings.

Parameter	Value
Rotation range	5
Width shift range	0.1
Height shift range	0.1
Shear range	0.1
Zoom range	0.1
Horizontal flip	True

4.3. CNN Setup

For the deep CNN architecture, we adapted a well known VGG19 architecture presented by Simonyan et al. in 2014 [40]. As presented in Figure 2, we left the convolutional base (blocks from 1 to 5) of VGG19 as it was presented originally, while the classifier part of the architecture was customized. Instead of a flatten layer, we utilized a 2-dimensional

global average pooling layer, followed by a dropout layer, fully connected layer with ReLU activation function and fully connected output layer with sigmoid activation function.

The dropout probability values for the base and TL experiments were set to 0.5, while the dropout value for the experiments utilizing the HPO-TL methods is being optimized (set) by the method itself. The number of units in fully connected layer, followed by the dropout layer, was for the base and TL experiments set to 256, while the number of units for HPO-TL based experiments are also being optimized by the method itself.

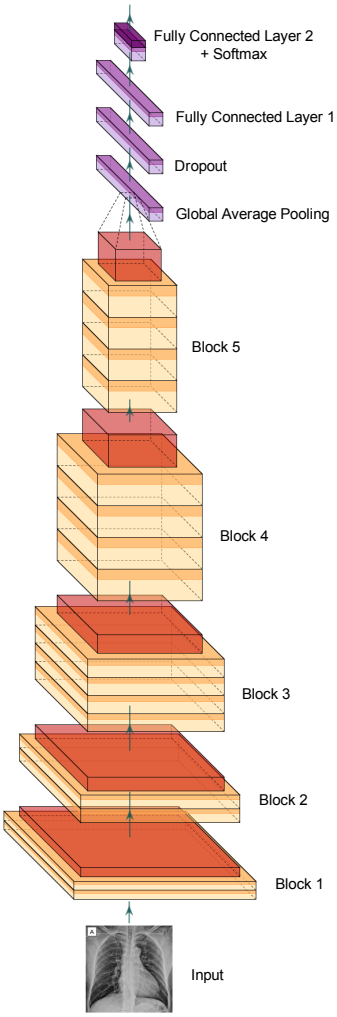


Fig. 2. The adapted VGG19 convolutional neural network architecture.

For the TL and HPO-TL based experiments, the transfer learning was utilized. The VGG19 convolutional base was pre-trained on the ImageNet dataset, while for the fine-tuning we enabled only the last convolutional block (block 5). The rest of the layers in convolutional base remained frozen (disabled for fine-tuning).

4.4. Settings of HPO-TL methods

Since the utilized HPO-TL based methods work in an iterative manner, where the next produced solution is based on fitness of the previous one, we tailored the dataset split methodology in order to retain the fairness between the compared approaches. While the base and TL experiments consume the whole training split of the dataset for the training purpose, we additionally divided the given training set in ratio 80:20, where the larger subset was used for training different solutions produced by a HPO-TL based method and evaluating them – calculating the fitness value against the remaining smaller subset of the initial training set.

For each fold, the method generates and evaluates 50 different solutions, from which the best – the one with the best (lowest) fitness value is selected and finally evaluated against the test split of the dataset. While this approach makes the such method computationally complex, we also introduced the early stopping approach to the evaluation of each solution, where the solutions which training is not improving for 5 consecutive epochs is prematurely stopped.

Table 4 presents parameter settings of three utilized optimization algorithms, namely grey wolf optimizer, differential evolution, and genetic algorithm, which were used together with the HPO-TL method. Other than population number NP parameter, all parameter values are set to default values as are defined in the NiaPy framework, from which we utilized the implementations of the selected optimization algorithms.

Table 4. Parameter settings for used optimization algorithms.

Parameter	Value		
	Grey Wolf Optimizer	**Differential Evolution**	**Genetic Algorithm**
Population NP	10	10	10
Scaling factor F	-	1	-
Crossover rate CR	-	0.8	0.25
Mutation rate MR	-	-	0.2

4.5. Training Parameter Settings

Presented in Table 5 are utilized training parameter settings for each of the conducted experiments. For each fold every method is provided with the total of 50 epochs, except the HPO-TL methods which, in worst-case scenario, consume a total of 2500 epochs (50 epochs for each solution evaluation). The batch size remains the same for all three experiments and it is set to 32. For the base and TL experiments, we set the learning rate

to $1 * 10^{-5}$ and optimizer to RMSprop, while the learning rate and optimizer for HPO-TL methods is set (optimized) by the method itself and therefore is not explicitly defined since it is not chosen deterministically.

Table 5. Training parameter settings for conducted experiments.

Parameter	Value		
	base	**TL**	**HPO-TL**
Nr. of epochs	50	50	2500
Batch size	32	32	32
Learning rate	$1 * 10^{-5}$	$1 * 10^{-5}$	-
Optimizer	RMSprop	RMSprop	-

5. Results

5.1. Classification Performance

In order to evaluate the COVID-19 X-ray image classification results, we first compared our three HPO-TL methods: TL-GWO, TL-GA, and TL-DE. For this purpose, we applied them upon the same CNN architecture using the same 10-fold cross-validation train-test folds, in order to objectively identify which of the three performed the best.

Results, obtained from the conducted experiments, are summarized in Table 6. As can be observed from the table, the difference among the three methods are quite small. Anyhow, the TL-DE method performed the best on average in most of the performance measures, while also achieving the lowest time for training. Interestingly, the results of the TL-DE method on all 10 folds were also the most stable, achieving the lowest standard deviation among the three methods regardless of the selected metric. In general, the second-best results were obtained by the TL-GWO method, while the results of the TL-GA lag a bit behind.

Table 6. Comparison of classification performance results on selected metrics over 10-fold cross-validation (averages and standard deviations are reported) for the three HPO-TL methods.

metric	TL-GWO	TL-GA	TL-DE
Accuracy	84.10 ± 3.2	82.45 ± 4.54	$\mathbf{84.44 \pm 2.91}$
AUC	83.61 ± 3.93	80.89 ± 6.26	$\mathbf{83.89 \pm 3.36}$
Precision	$\mathbf{88.52 \pm 5.59}$	85.09 ± 7.53	88.16 ± 3.84
Recall	85.82 ± 7.16	$\mathbf{87.75 \pm 6.03}$	86.38 ± 3.98
F**-1**	86.79 ± 2.93	86.01 ± 3.08	$\mathbf{87.16 \pm 2.43}$
Kappa	66.70 ± 6.92	62.27 ± 10.86	$\mathbf{67.40 \pm 6.19}$
Time	6096.40 ± 427.33	5383.10 ± 501.70	$\mathbf{5020.30 \pm 380.97}$

Fig. 3 shows a comparison of test accuracy results obtained by the three methods for all 10 folds on the Covid-19 X-ray image dataset. As we can see, in two folds the TLGA

method performed a bit worse than the other two methods, while other differences are rather insignificant. If we look in detail, there is one situation where TL-GWO performed noticeably better than TL-DE (fold-6), while TL-DE performed noticeably better than TL-GWO in two situations (in fold-0 and fold-9). Very similar to accuracy were also the results of the rest of the metrics. Fig. 4 shows the box-plot comparison of the three methods with regard to AUC. It can be seen that the TL-DE achieved the best average AUC result, while also being the most stable among the methods.

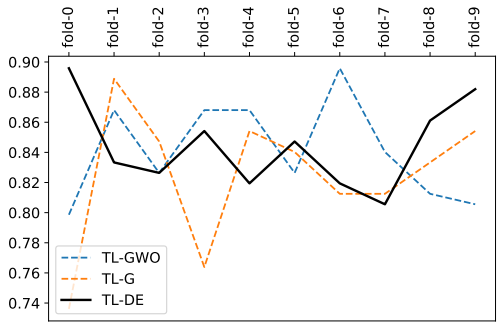


Fig. 3. Accuracy of the three HPO-TL methods on 10 folds.

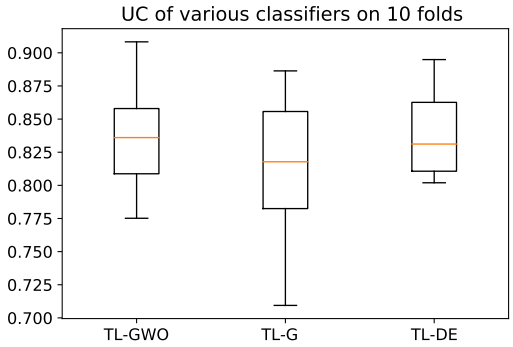


Fig. 4. Comparison of AUC for the three HPO-TL methods.

While the predictive performance results of the TL-DE and the TL-GWO methods were barely distinguishable, there was a bigger difference with regard to the time elapsed for training the CNN model (Fig. 5). As it can be seen, the TL-GWO method consumed the most time for training, while the TL-DE was the fastest of the three methods.

In general, with regard to the presented classification performance results, the TL-DE can be considered as the best method overall, although the differences between the three methods turned out to be very small.

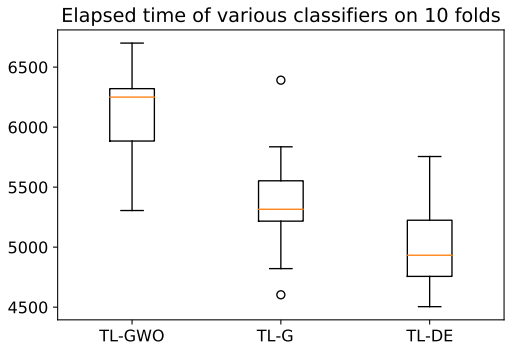


Fig. 5. Comparison of consumed time for training for the three HPO-TL methods.

5.2. Comparison with Other Methods

As the TL-DE turned out to be the best of the three methods, we wanted to compare it with the two most common existing approaches – base method for training the CNN model and TL method that performs transfer learning upon the same CNN architecture. For the base method, we utilized the VGG19 [40] CNN architecture, pre-trained on the ImageNet [11] dataset. For the TL method, we utilized the transfer learning approach and applied it on the same VGG19 CNN convolutional base. In this manner, the differences among the obtained predictive performance results can be contributed solely to the consequence of different learning method used. For the sake of comparison, we performed a series of experiments on the COVID-19 X-ray image dataset using the 10-fold cross-validation approach.

Results, obtained from the conducted experiments, are summarized in Table 7. As can be observed from the table, our proposed TL-DE method showed the best performance among the three compared methods regardless of the selected performance metric, with the exception of elapsed training time. In general, the second-best results were obtained by the TL method, while significantly the worst results were obtained by the base method.

Table 7. Comparison of classification performance results on selected metrics over 10-fold cross-validation (averages and standard deviations are reported) for the three compared methods.

metric	base	TL	TL-DE
Accuracy	54.51 ± 10.78	80.97 ± 4.37	**84.44 ± 2.91**
AUC	50.00 ± 0.00	80.85 ± 4.25	**83.89 ± 3.36**
Precision	42.85 ± 29.57	86.94 ± 4.18	**88.16 ± 3.84**
Recall	70.00 ± 48.30	81.38 ± 7.53	**86.38 ± 3.98**
F**-1**	53.16 ± 36.68	83.84 ± 4.20	**87.16 ± 2.43**
Kappa	0.00 ± 0.00	60.69 ± 8.56	**67.40 ± 6.19**
Time	377.50 ± 9.23	**340.40 ± 6.45**	5020.30 ± 380.97

Fig. 6 shows a comparison of test accuracy results obtained by the three compared methods for all 10 folds on the Covid-19 X-ray image dataset. As we can see, the TL-DE method achieved the highest accuracy in 7 out of 10 folds, followed by the TL method with 3 remaining wins, while the results of the base method lag quite distinctively behind. In all three folds, where the TL method outperformed the TL-DE, the differences were hardly noticeable, while the advantage of the TL-DE method were substantial in 6 out of 7 folds. Very similar results were obtained for all other predictive performance metrics. Fig. 7 shows the box-plot comparison of the three methods with regard to AUC, one of the most important metric when evaluating classification models in medicine, where the advantage of the TL-DE method can be easily observed. Not only that the mean AUC is the highest, the TL-DE produced results also with smaller standard deviation than the TL method.

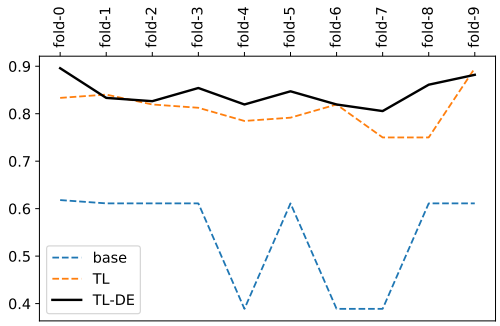


Fig. 6. Accuracy of the three compared methods on 10 folds.

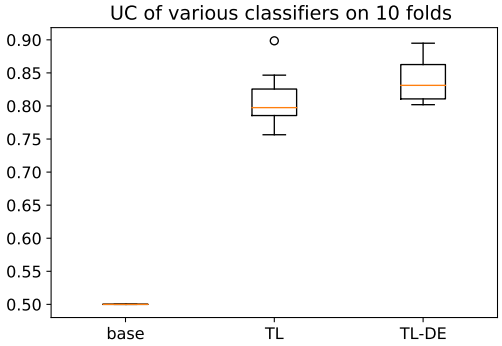


Fig. 7. Comparison of AUC for the three compared methods.

To achieve such excellent classification results, however, the TL-DE method pays its price with a much longer training time. While the base method spent on average 377.5

seconds to fully train the CNN mode, and the TL method 340.4 seconds, the HPO-based TL-DE method spent on average 5020.3 seconds, which is of course the consequence of the used optimization method.

5.3. Statistical Comparison

To evaluate the statistical significance of classification performance results of the three compared methods (base, TL, and TL-DE), we first applied the Friedman test by calculating the average Friedman ranks, Friedman asymptotic significance and p-values for all the three methods and for all 7 measures (acc, auc, prec, rec, F-1, kappa, and time), as suggested by Demšar [10]. The statistical results are summarized in Table 8. We can see that there is a significant difference among the three methods for all measures but the recall. The TL-DE is significantly better than the base method with regard to accuracy, AUC, precision, F-1, and kappa. It is also significantly better than the TL method with regard to accuracy, F-1, and kappa, while the difference is nearly significant with regard to AUC. On the other hand, the TL-DE method is significantly worse than the other two methods with regard to the required training time, as expected.

Table 8. Statistical comparison (p-values) of the Friedman test and Wilcoxon signed rank test for TL-DE vs. other two methods for all 7 metrics; significant differences are marked with *.

	Friedman test	**Wilcoxon signed rank test**	
metric	**all three**	**TL-DE vs. base**	**TL-DE vs. TL**
Accuracy	<0.001*	0.002*	0.033*
AUC	<0.001*	0.005*	0.084
Precision	<0.001*	0.002*	0.625
Recall	0.154	—	—
F-1	<0.001*	0.002*	0.014*
Kappa	<0.001*	0.002*	0.049*
Time	<0.001*	0.002*	0.002*

5.4. HPO-TL Methods Parameter Selection Analysis

Presented in Table 9 are the best performing selected values for optimized parameters for each fold. Inspecting the presented selected values, we can see that in the 4 folds, the number of selected units in the last fully-connected (dense) layer was set to 128, while also in 4 folds the number of selected units was set to 256, which is in line with the value which we handpicked. Those two values together were selected in 80% of all folds, while the remaining 2 selections were the lowest (64) and highest (1024) of possible values. The selected dropout probabilities are roughly ranging from 0.5 to 0.76, 7 of being in a range between 0.5 and 0.57 which is somewhat similar to what we manually selected for the remaining experiments (0.5). Focusing on the selected optimizer function, we can observe that the selection is almost evenly distributed between the RMSprop (4 out of 10 folds) and Adam optimizer (6 out of 10 folds), while the SGD is not a part of the best

found solution in any fold. Regarding the selection of learning rates, 4 times each were selected learning rates $5*10^{-4}$ and $5*10^{-5}$. The latter is also the same as our handpicked value for the learning rate in the TL experiment.

Table 9. Best achieved solutions for the sought parameters per fold using TL-DE.

Fold	Neurons in last dense layer	Dropout probability	Optimizer	Learning rate
0	128	0.660450	adam	0.00050
1	64	0.500000	adam	0.00050
2	128	0.754095	adam	0.00005
3	256	0.500000	rmsprop	0.00005
4	256	0.512172	rmsprop	0.00050
5	128	0.537716	rmsprop	0.00001
6	256	0.571608	adam	0.00005
7	128	0.570084	adam	0.00050
8	1024	0.763849	rmsprop	0.00010
9	256	0.536784	adam	0.00005

We have also analyzed and compared parameter selections of remaining two HPO-TL methods, TL-GWO and TL-GA. Comparing the best parameter selections from best performing variation (TL-DE) against parameters selections of TL-GWO (Table 10), we can see that in general, the values for number of last hidden layer are lower, but on the other side, the dropout probability values of the best parameter selections are more similar to the best performing TL-DE variation. Also, the selection of optimizer is somewhat similar to the TL-DE variation, with a bit more tendency to selection of adam optimizer function.

Table 10. Best achieved solutions for the sought parameters per fold using TL-GWO.

Fold	Neurons in last dense layer	Dropout probability	Optimizer	Learning rate
0	128	0.660413	adam	0.00005
1	128	0.703526	adam	0.00010
2	64	0.732706	rmsprop	0.00005
3	512	0.513470	adam	0.00050
4	64	0.532255	adam	0.00010
5	64	0.642986	adam	0.00010
6	256	0.516314	rmsprop	0.00005
7	128	0.734866	rmsprop	0.00010
8	256	0.777165	adam	0.00010
9	64	0.555470	adam	0.00010

Table 11. Best achieved solutions for the sought parameters per fold using TL-GA.

Fold	Neurons in last dense layer	Dropout probability	Optimizer	Learning rate
0	1024	0.731663	rmsprop	0.00001
1	512	0.778245	adam	0.00050
2	128	0.557576	rmsprop	0.00050
3	256	0.872333	rmsprop	0.00050
4	256	0.567304	adam	0.00010
5	512	0.512127	adam	0.00010
6	256	0.532396	adam	0.00050
7	128	0.511880	adam	0.00010
8	256	0.821332	adam	0.00010
9	256	0.613837	adam	0.00050

If we compare the TL-DE further, with the worst performing of three variations TL-GA (Table 11), we can observe that selection of values for number of hidden units are quite similar to the best performing TL-DE variation. The selection of the optimizer function is proportionally the same as in TL-GWO. Interestingly, none of the three variations chose the SGD optimizer function as the best performing optimizer in any combination of parameter selections. The biggest difference between the parameter selection values can be seen in the range of dropout probabilities, which is in the case of TL-GA varying from 0.51 to 0.87.

5.5. Interpretable Representation of Model

When employing predictive models in various mission-critical decision-making systems, one of the biggest problem is determining trust in individual prediction of such models. Especially if such systems are being used in the fields like medicine, where predictions cannot be acted upon blind faith, consequences may be catastrophic [39].

In general, it is a common practice to evaluate predictive models using different metrics against the available test dataset. However, such common metrics may not be necessarily indicative of the model's goal. Therefore, inspecting individual instances and their representations which can be interpreted is a good complementary solution, especially when dealing with so called "black-box" methods, to gain useful insights on how our model perceives it. Additionally, such evaluation can also help us increase the understanding and trust in our predictive model.

In Figure 8, we are showcasing LIME interpretable representations of our best performing predictive model, obtained by HPO-TL variation named TL-DE, which utilizes a DE algorithm for finding most suitable set of hyper-parameter values. In our previous research on COVID-10 identification [43], we have also used LIME method for evaluating the models' performance from a qualitative standpoint. The conducted analysis in the mentioned research was performed in such way, that all corresponding interpretable representations obtained from LIME were plotted on each corresponding sample (chest x-ray image), and each sample was then evaluated individually. In contrast to our previous research, here we are taking a different approach where we are obtaining the interpretable

representations in the same manner for each sample as in our previous research, but in this case we are aggregating them into one. This allows us to get an insight into predictive model behavior over all test samples in one aggregated interpretable representation, instead of analyzing each sample individually.

Labeled as *a)* in Figure 8, we are showing green groups of pixels (super-pixel) denoting sections of image which have a positive impact towards classifying our input chest x-ray image as COVID-19, while labeled as *b)*, we are showing red super-pixels denoting sections of image which have a negative impact. The scale on the right side of each analyzed image is representing the intensity of each marked region, where a darker color is denoting a higher intensity and vice versa. Inspecting the image labeled *a)*, we can see similar patterns as we already identified in our previous research [43]. In the image showing green super-pixels, those are a bit more focused on the central thorax body region in contrast to the marked red super-pixels which are spread more across the whole upper body including shoulders and neck. Also, the intensity of the red super-pixels is a bit higher in those regions.

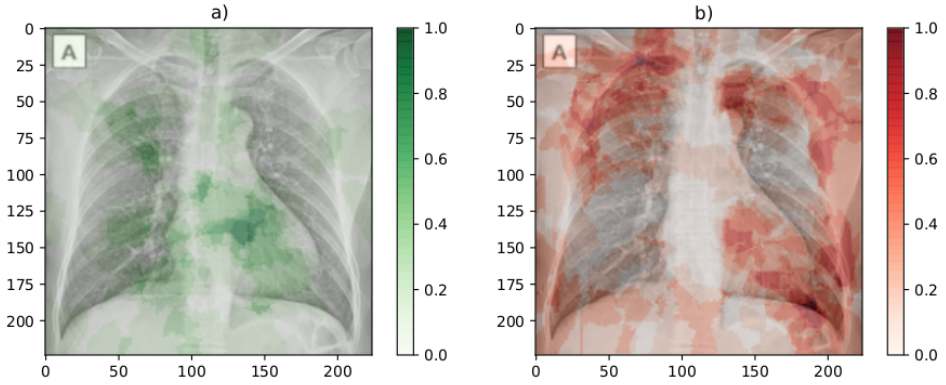


Fig. 8. Explaining predictive model decisions using LIME method. The image below the a) label represents the super-pixels which positively impact towards the COVID-19 class, while the image below the b) label is showcasing the super-pixels which negatively impact towards the COVID-19 class.

Comparing those two aggregated positive and negative interpretable representations could also be challenging, trying to compare specific regions of each sample. Therefore, we decided to aggregate those two into only one interpretable representation, which would possibly give us more clear insight into the regions which the predictive model is identifying as the ones having positive or negative impact. Such aggregated interpretable representation is presented on Figure 9 labeled as *a)*. As we can observe from the image, we can see that the most green super-pixels are still positioned in more central part of thorax body region, while the red super-pixels are also still positioned more on the outer parts of upper body. Interestingly, we can see the that green super-pixels also cover a region of the aortic arch and a part of the heart, which is similar to findings in our previous research.

Also, if we compare interpretable representations of our predictive model with similar researches [13, 53] and their interpretations, we can observe that our green super-pixels are in similar positions as in the most intense regions of the mentioned researches.

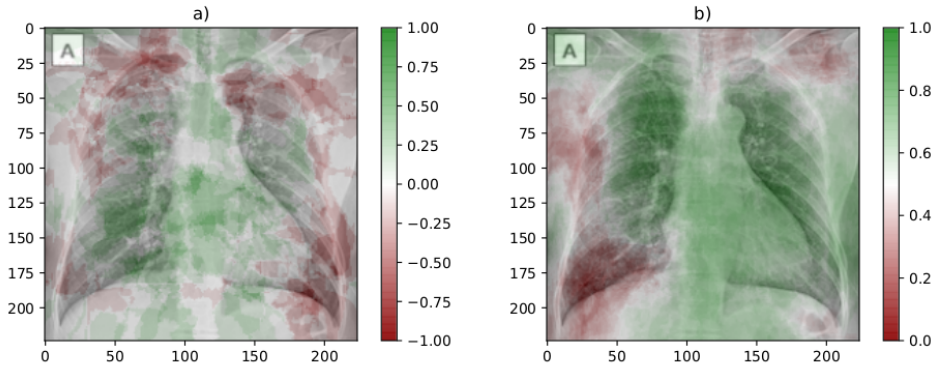


Fig. 9. Comparison of aggregated explanations between TL-DE and TL-GWO. The image below the a) label represents the TL-DE aggregated LIME explanations, while the image below the b) label is showcasing TL-GWO aggregated LIME explanations.

Interpretable representation labeled as *b)* presented on Figure 9 is also showing aggregated representation obtained from LIME but for the TL-GWO variation of the HPO-TL method. Since by the accuracy classification metric the TL-GWO lagged behind only by 0.34%, we were curious how does the representations from best performing model obtained by TL-GWO look like in comparison to the TL-DE. Comparing those two images, we can see that regardless of lagging behind for a small amount, the visual representation reveals that the difference is quite noticeable. In the case of TL-DE, the bounds of each super-pixel are more sharp and the border between them is more noticeable, while in the case of TL-GWO the borders of super-pixels are more blurred and the border between them is not so distinct. Overall, the TL-DE super-pixels are more exact in contrast to TL-GWO. Also, the green super-pixels of TL-GWO cover the majority of the upper body, which is not necessarily useful when trying to detect particular affected regions.

6. Conclusions

In this work, we proposed a generalized image classification method, based on GWOTLT [43], that trains a CNN using transfer learning with fine-tuning approach, in which hyper-parameter values are optimized with an optimization algorithm. Such generalized version, named HPO-TL, enables us to use different optimization algorithms which can be useful when dealing with various domain problems where different optimization algorithms can result in better final predictive model. The generalized method has been applied on a dataset of COVID-19 chest X-ray images using three different optimization algorithms DE, GWO, and GA. The obtained results showed that the best performing variation of

HPO-TL method is TL-DE which featured DE as an optimization algorithm. The best performing TL-DE also showed an impressive performance in all classification metrics when comparing to the conventional approaches of training a CNN.

We have also adopted a local interpretable model-agnostic explanations approach to provide insights of the COVID-19 disease, based on classification of chest X-rays. In contrast to straight-forward usage of such explanations, we have aggregated them into one, trying to get an insight on overall perception of a predictive model over all test samples instead of anayzing one by one. Thus, such approach was able to provide some interesting insights into the characteristics of COVID-19 disease and predictive model behaviour, by performing qualitative explanations upon the results of the trained model classification of a set of X-ray images.

In the future, we would like to expand our research to utilize different CNN architectures and conducted qualitative evaluations using additional methods such as SHAP [24].

Acknowledgments. The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057).

References

1. et al., M.A.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Apostolopoulos, I.D., Mpesiana, T.A.: Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine 43(2), 635–640 (jun 2020)
3. Apostolopoulos, I.D., Mpesiana, T.A.: Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine p. 1 (2020)
4. Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Advances in neural information processing systems. pp. 2546–2554 (2011)
5. Brezočnik, L., Fister, I., Vrbančič, G.: Applying differential evolution with threshold mechanism for feature selection on a phishing websites classification. In: Welzer, T., Eder, J., Podgorelec, V., Wrembel, R., Ivanović, M., Gamper, J., Morzy, M., Tzouramanis, T., Darmont, J., Kamišalić Latifić, A. (eds.) New Trends in Databases and Information Systems. pp. 11–18. Springer International Publishing, Cham (2019)
6. Ching, J.Y., Wong, A.K.C., Chan, K.C.C.: Class-dependent discretization for inductive learning from continuous and mixed-mode data. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(7), 641–651 (1995)
7. Chollet, F., et al.: Keras (2015), https://keras.io
8. Cohen, J.P., Morrison, P., Dao, L.: Covid-19 image data collection. arXiv 2003.11597 (2020), https://github.com/ieee8023/covid-chestxray-dataset
9. Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution–an updated survey. Swarm and Evolutionary Computation 27, 1–30 (2016)
10. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7 (2006)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

12. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: 7th International Conference on Information and Knowledge Management. pp. 148–152 (January 1998), https://www.microsoft.com/en-us/research/publication/inductive-learning-algorithms-and-representations-for-text-categorization/
13. Duran-Lopez, L., Dominguez-Morales, J.P., Corral-Jaime, J., Vicente-Diaz, S., Linares-Barranco, A.: Covid-xnet: a custom deep learning system to diagnose and locate covid-19 in chest x-ray images. Applied Sciences 10(16), 5683 (2020)
14. Fang, Y., Zhang, H., Xie, J., Lin, M., Ying, L., Pang, P., Ji, W.: Sensitivity of chest ct for covid-19: comparison to rt-pcr. Radiology 296(2), E115–E117 (2020)
15. Faris, H., Aljarah, I., Al-Betar, M.A., Mirjalili, S.: Grey wolf optimizer: a review of recent variants and applications. Neural computing and applications 30(2), 413–435 (2018)
16. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, BioL Cybem. 36 (1980) 193-202. S. Shiotani et al./Neurocomputing 9 (1995) Ill-130 130 (1980)
17. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
18. Govindarajan, S., Swaminathan, R.: Differentiation of COVID-19 conditions in planar chest radiographs using optimized convolutional neural networks. Applied Intelligence (2020)
19. Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). pp. 21–26. ICST (Institute for Computer Sciences, Social-Informatics and ... (2016)
20. Kamilaris, A., Prenafeta-Boldú, F.X.: Deep learning in agriculture: A survey. Computers and electronics in agriculture 147, 70–90 (2018)
21. Kikkisetti, S., Zhu, J., Shen, B., Li, H., Duong, T.Q.: Deep-learning convolutional neural networks with transfer learning accurately classify COVID-19 lung infection on portable chest radiographs. PeerJ 8 (nov 2020)
22. Kong, Q., Trugman, D.T., Ross, Z.E., Bianco, M.J., Meade, B.J., Gerstoft, P.: Machine learning in seismology: Turning data into insights. Seismological Research Letters 90(1), 3–14 (2018)
23. Lorenzo, P.R., Nalepa, J., Kawulok, M., Ramos, L.S., Pastor, J.R.: Particle swarm optimization for hyper-parameter selection in deep neural networks. In: Proceedings of the genetic and evolutionary computation conference. pp. 481–488 (2017)
24. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions. arXiv preprint arXiv:1705.07874 (2017)
25. Lundervold, A.S., Lundervold, A.: An overview of deep learning in medical imaging focusing on mri. Zeitschrift für Medizinische Physik 29(2), 102–127 (2019)
26. Majeed, T., Rashid, R., Ali, D., Asaad, A.: Covid-19 detection using cnn transfer learning from x-ray images. medRxiv (2020)
27. Majeed, T., Rashid, R., Ali, D., Asaad, A.: Issues associated with deploying CNN transfer learning to detect COVID-19 from chest X-rays. Physical and Engineering Sciences in Medicine 43(4), 1289–1303 (dec 2020)
28. Marques, G., Agarwal, D., de la Torre Díez, I.: Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. Applied Soft Computing Journal 96 (nov 2020)
29. McKinney, W.: Data structures for statistical computing in python. In: van der Walt, S., Millman, J. (eds.) Proceedings of the 9th Python in Science Conference. pp. 51 – 56 (2010)
30. Mirjalili, S.: Genetic algorithm. In: Evolutionary algorithms and neural networks, pp. 43–55. Springer (2019)
31. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in engineering software 69, 46–61 (2014)

32. Nayak, S.R., Nayak, D.R., Sinha, U., Arora, V., Pachori, R.B.: Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study. Biomedical Signal Processing and Control 64, 102365 (feb 2021), https://doi.org/10.1016/j.bspc.2020.102365
33. of North America, R.S.: RSNA Pneumonia Detection Challenge — Kaggle, https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/overview
34. Organization, W.H., et al.: Covid-19 weekly epidemiological update - 2 february 2021. In: COVID-19 Weekly Epidemiological update - 2 February 2021. World Health Organization (2021)
35. Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O., Acharya, U.R.: Automated detection of covid-19 cases using deep neural networks with x-ray images. Computers in biology and medicine 121, 103792 (2020)
36. Pathak, Y., Shukla, P.K., Tiwari, A., Stalin, S., Singh, S.: Deep transfer learning based classification model for covid-19 disease. Irbm (2020)
37. 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)
38. Podgorelec, V., Pečnik, Š., Vrbančič, G.: Classification of similar sports images using convolutional neural network with hyper-parameter optimization. Applied Sciences 10(23), 8494 (2020)
39. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
40. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
41. Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B., Liang, J.: Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE transactions on medical imaging 35(5), 1299–1312 (2016)
42. Van Der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering 13(2), 22 (2011)
43. Vrbančič, G., ˇŠ. Pečnik, Podgorelec, V.: Identification of covid-19 x-ray images using cnn with optimized tuning of transfer learning. In: 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). pp. 1–8 (2020)
44. Vrbančič, G., Brezočnik, L., Mlakar, U., Fister, D., Fister Jr., I.: NiaPy: Python microframework for building nature-inspired algorithms. Journal of Open Source Software 3 (2018), https://doi.org/10.21105/joss.00613
45. Vrbancic, G., Fister, I.J., Podgorelec, V.: Automatic Detection of Heartbeats in Heart Sound Signals Using Deep Convolutional Neural Networks. Elektronika ir Elektrotechnika 25(3), 71–76 (jun 2019), http://eejournal.ktu.lt/index.php/elt/article/view/23680
46. Vrbancic, G., Fister, I.J., Podgorelec, V.: Parameter Setting for Deep Neural Networks Using Swarm Intelligence on Phishing Websites Classification. International Journal on Artificial Intelligence Tools 28(6), 28 (oct 2019)
47. Vrbancic, G., Fister, I.J., Podgorelec, V.: Parameter Setting for Deep Neural Networks Using Swarm Intelligence on Phishing Websites Classification. International Journal on Artificial Intelligence Tools 28(6), 28 (oct 2019)
48. Vrbancic, G., Podgorelec, V.: Automatic Classification of Motor Impairment Neural Disorders from EEG Signals Using Deep Convolutional Neural Networks. Elektronika ir Elektrotechnika 24(4), 3–7 (aug 2018), http://eejournal.ktu.lt/index.php/elt/article/view/21469
49. Vrbančič, G., Zorman, M., Podgorelec, V.: Transfer learning tuning utilizing grey wolf optimizer for identification of brain hemorrhage from head ct images. In: StuCoSReC: proceedings of the 2019 6th Student Computer Science Research Conference. pp. 61–66 (2019)

50. Vrbančič, G., Podgorelec, V.: Transfer learning with adaptive fine-tuning. IEEE Access 8, 196197–196211 (2020)
51. Yang, Q., Ling, C., Chai, X., Pan, R.: Test-cost sensitive classification on data with missing values. IEEE Transactions on Knowledge & Data Engineering 18(5), 626–638 (2006)
52. Yu, T., Zhu, H.: Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689 (2020)
53. Zebin, T., Rezvy, S.: Covid-19 detection and disease progression visualization: Deep learning on chest x-rays for classification and coarse localization. Applied Intelligence pp. 1–12 (2020)
54. Zhang, Y., Song, K., Sun, Y., Tan, S., Udell, M.: "why should you trust my explanation?" understanding uncertainty in lime explanations. arXiv preprint arXiv:1904.12991 (aug 2014)
55. Zhu, X., Wu, X.: Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering. IEEE Transactions on Knowledge & Data Engineering 18(10), 1435–1440 (2006)

Grega Vrbančič received the B.Sc. and M.Sc. degrees in informatics and communication technologies from the University of Maribor in 2015 and 2017, respectively. In 2021, he received a Ph.D. degree from the University of Maribor. Currently, he is a teaching assistant and a researcher with the Faculty of Electrical Engineering and Computer Science, University of Maribor. He is the author of six peer-reviewed scientific journal articles and several conference papers. He has been involved in several industrial research and development projects. His research interests include deep learning, especially convolutional neural networks, focusing on training strategies and transfer learning.

Špela Pečnik received her bachelor's degree in Information Technology in 2017, and a master's degree in Information Technology in 2019 from the University of Maribor, Faculty of Electrical Engineering and Computer Science. Since 2019, she has been employed as a teaching assistant and researcher at the Faculty of Electrical Engineering and Computer Science, University of Maribor. Her research areas include artificial intelligence, machine learning, data mining, and information systems.

Vili Podgorelec is a professor of computer science at the University of Maribor, Slovenia, where he received the Ph.D. degree in 2001. He has been involved in AI and ML for 20 years, where he gained professional experience in implementation of many scientific and industrial R&D projects related to analysis, design, implementation, integration, and evaluation of intelligent information systems using AI and ML. He has authored more than 50 peer-reviewed scientific journal papers, more than 100 conference papers, three books and several book chapters on machine learning, computational intelligence, data science, medical informatics, and software engineering. Dr. Podgorelec has worked as a visiting professor and/or researcher at several universities around the world, including University of Osaka, Japan; Federal University of Sao Paulo, Brazil; University of Nantes, France; University of La Laguna, Spain; University of Madeira, Portugal; University of Applied Sciences Seinäjoki, Finland; University of Applied Sciences Valencia, Spain. He received several international awards and grants for his research activities.

Received: February 09, 2021; Accepted: October 08, 2021.

Computer Science and Information Systems 19(1):353–378 https://doi.org/10.2298/CSIS210222067B

A Fast Non-dominated Sorting Multi-objective Symbiotic Organism Search Algorithm for Energy Efficient Locomotion of Snake Robot*

Yesim Aysel Baysal and Ismail Hakki Altas

Department of Electrical and Electronics Engineering
Karadeniz Technical University
Trabzon, Turkey
{yabaysal, ihaltas}@ktu.edu.tr

Abstract. This paper deals with energy efficient locomotion of a wheel-less snake robot. This is very crucial for potential applications of untethered snake robots. The optimum gait parameters for the energy efficient locomotion of the snake robot are obtained with two different multi-objective algorithms based on symbiotic organism search algorithm by considering both minimizing the average power consumption and maximizing the forward velocity of the robot. This paper also investigates the energy efficient locomotion of the snake robot under different environment conditions. The obtained results demonstrate that both proposed methods achieve satisfying stable results regarding power consumption reduction with optimal forward velocity for lateral undulation motion. However, it is seen that fast non-dominated sorting multi-objective symbiotic organism search algorithm provides advantage on obtaining a uniformly distributed solution set with a good diversity only in a single run. This paper is important in terms of presenting useful results for developing efficient motion and environmental adaptability of the snake robot.

Keywords: Energy efficiency, adaptive locomotion, friction condition, optimum gait parameters, symbiotic organism search algorithm, multi-objective optimization, snake robot

1. Introduction

Developed by inspiring from the perfect motions of biological snakes in unknown, irregular environments, snake robots give an outstanding locomotion in many challenging environments in comparison with other mobile robots such as legged, tracked and wheeled robots. This unique feature makes snake robots useful for many purposes in real-world applications such as firefighting, military purposes, rescue and search operations, maintenance of nuclear plants and pipelines. Energy efficient locomotion for especially such applications is very important for the snake robot. Despite this, many of studies focus on modelling, development, and control of these

* This article is an extended version of a conference paper entitled "Optimally Efficient Locomotion of Snake Robot" that was initially published in 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) [20].

mechanism. There are very few studies addressing energy efficient locomotion for snake robot in the literature.

Powell's method, a gradient-free optimization method, is used to maximize the forward velocity of the snake robot for a given environment by optimizing the parameters of the central pattern generator (CPG) [1]. In this study, only the amplitude and phase lag parameters of the motion pattern are handled, and the effect of frequency parameter on efficient motion is not examined. The experimental studies are executed for crawling motion on a horizontal plane and for swimming motion while in the simulation studies the crawling motion on a horizontal plane and on a slope is handled. An important disadvantage of this study is that the optimization method used has the potential to get trapped into local optima. In [2], genetic algorithm (GA) is used to find forward head serpentine (FHS) gait parameters maximizing the velocity of the robot by keeping the angular changes of the head link in an acceptable range. However, the energy consumption is not considered in this study. GA is also used in [3] to optimize CPG parameters and connection weights in terms of velocity of the snake robot and furthermore a fuzzy logic tuner is designed to maintain optimal locomotion in different environmental conditions in this study. In [4], optimal CPG parameters are discussed to achieve the efficient locomotion of the snake robot under different friction or slope. A criterion for locomotion efficiency is described as the ratio between forward displacement and energy consumption. In another study, parameters of a CPG based locomotion controller are obtained using GA according to changing ground friction for adaptive locomotion of snake robot [5]. Three different environments whose tangential friction coefficients are same is used to test lateral undulation locomotion in the study. In [6], three different gaits of the snake robot including lateral undulation, sidewinding locomotion, and sinus-lifting motion are analysed at eight different environments in terms of trade-off between locomotion speed of the robot and energy efficiency. For different friction coefficients, Pareto curves between locomotion speed and efficiency are obtained for all three gaits by 1500 random combinations of amplitude and frequency parameters in a given range of values. For underwater snake robots, effect of each parameter defining the motion pattern such as the amplitude, frequency and phase offset parameters on the consumed energy and the forward velocity of the robot are separately investigated [7]. In this study, cost of transportation (COT) index is used to define energy efficient motion. In [8], an optimization framework for solving a multi-objective optimization problem in order to obtain optimal gait parameters is proposed for underwater snake robot and particle swarm optimization (PSO) is used to investigate the energy efficiency of these robots. Optimal gait parameters are obtained for two different motion pattern, the lateral and eel-like motion of the underwater snake robot. An extension of the optimization framework proposed in [8] is presented in [9] for snake robots both on land and in water. In [10], to optimize locomotion efficiency, the locomotion parameters of a snake robot controlled by CPG are investigated considering the speed and energy consumption of the robot. The locomotion parameters are optimized with the cuckoo search (CS) algorithm for environments with different space widths and different ratios between friction coefficients in the tangential and normal directions. The aim of the optimization is to maximize the locomotion efficiency of the snake robot obtained by dividing the displacement of the robot by the energy consumption. Recently, a reinforcement learning (RL) based controller for generating locomotion gaits of the snake robot using the proximal policy optimization (PPO)

algorithm is proposed in [11]. Moreover, the grid search and Bayesian optimization algorithms are used to optimize the parameter set of the motion in this study. An expanded version of this study is presented in [12] by using the adversarial inverse reinforcement learning (AIRL) algorithm.

The literature review shows that the energy efficient locomotion problem of the snake robot is generally handled considering only maximization of the velocity of the robot. Although there are some studies considering both velocity of the robot and energy consumption, they have generally combined these two objectives into a single objective function by multiplied them with a weight factor. This approach called the weighted sum method has some disadvantages. The first of these is that selecting these weights precisely and accurately is very important and a difficult task since the weights effect the priority of the objectives in the objective function. The weights are generally determined uniformly distributed. However, this is not always guaranteeing a uniformly distributed of Pareto optimal solutions. The second disadvantage is that obtaining a Pareto front curve requires that the algorithm should be run multiple times with different weights. This is a time-consuming and an exhaustive process. Another disadvantage of this approach is that in mixed optimization problems including both minimization and maximization, the objectives should be converted to same type. Due to these disadvantages, it is more suitable that energy efficient locomotion problem is addressed as multi-objective optimization problem to obtain Pareto optimal solution set between the two objectives.

Evolutionary computation (EC) techniques for solving multi-objective problems are more useful since they are able to generate a set of multiple solutions in a single optimization run. For different kinds of optimization problems, there are a lot of different EC techniques [13-16]. Among these techniques, symbiotic organism search (SOS) algorithm which simulates the symbiotic interaction strategies between organisms in an ecosystem has increasingly become popularity in recent years because it presents more robust results with a faster convergence speed for optimization problems in various domains [17]. The superiority of the SOS algorithm over other EC algorithms is due to its three-stage strategy (mutualism, commensalism, and parasitism) used for updating the solutions. The first two stages of the SOS algorithm focus on exploration, and the two-stage exploration ensures the algorithm to center upon the region where the best solution is located. Moreover, the operation of random dimension mutation in the parasitism phase results in the better solution by helping jump out of local optima [18]. Another advantage of the SOS algorithm is that it is not necessary any specific algorithm parameters unlike most other metaheuristic algorithms. Therefore, the problem of trapping to local optima resulting from improper parameter setting is removed. This property of the algorithm also provides to reduce computational time [19].

This article is an expanded version of [20] in which the weighted sum method based multi-objective symbiotic organism search (MOSOS) algorithm was used to obtain optimal locomotion of the snake robot. In this paper, a fast non-dominated sorting multi-objective symbiotic organism search (FNSMOSOS) algorithm is proposed to find the parameters of the most efficient motion pattern for snake robot. FNSMOSOS is a very powerful algorithm at finding the optimal trade-off between objectives because it combines all the advantages of SOS algorithm with two important techniques such as fast non-dominated sorting technique generating the solution as close to the Pareto optimal solution as possible and crowding distance technique ensuring diversity in

solution. The optimal trade-off between forward velocity of the snake robot and average power consumption are obtained for lateral undulation motion which is the most common snake robot locomotion. From the results, it is seen that the proposed method has ability to produce promising results on obtaining the optimal forward velocity to achieve lower power consumption. When compared FNSMOSOS with the weighted sum method based MOSOS, it is seen that solutions obtained by FNSMOSOS are more uniformly distributed along the Pareto front. Moreover, FNSMOSOS provides a larger set of different solutions to decision maker only in a single run. Thus, the results can be used as a guide for control design of the snake robot.

For energy efficient locomotion, snake robots should adapt their motions to environments with different friction conditions, just as biological snakes do. Therefore, in this paper, the effectiveness of the proposed method in finding optimal gait parameters of snake robot is also investigated when the snake robot moves in different environmental condition. In order to test the reliability and robustness of FNSMOSOS in improving adaptive locomotion of the snake robot, environments representing a fairly wide friction range from glass to wood are used. The obtained results show that FNSMOSOS contributes the snake robot to maintain optimal locomotion in different environmental condition. When considering motion efficiency of snake robots is less than other mobile robots due to high friction, these results are very important for environmental adaptability of the snake robot.

The rest of this paper is organized as follows. In Section 2, the dynamic model of the snake robot is presented. The application of the proposed algorithms for energy efficient locomotion problem are explained in detail in Section 3 and the obtained results are reported and discussed in Section 4. Finally, the main conclusions are summarized in Section 5.

2. Snake Robot Modelling

In this section, equations of motion of the snake robot are briefly presented. For more details, see [21] and [22]. In Fig.1, snake robot diagram with n rigid links and its kinematic parameters are seen.

The link is of mass m_i, length $2l_i$, and moment of inertia $J_i = (m_i l_i^2/3)$. Each link is interconnected by n-1 motorized joints. Link angle θ_i is defined as angle between link i and the global x axis while joint angle ϕ_i is the difference between the link angles of two neighboring links and defined by $\phi_i=\theta_i - \theta_{i+1}$. (x_i, y_i) describe global coordinates of the center of mass (CM) of link i, while (p_x, p_y) are global coordinates of the CM of the robot. Fig. 2 shows forces and torques acting on the link i of the snake robot. These forces are ground friction force $(f_{R,x,i}, f_{R,y,i})$ and joint constraint forces $(h_{x,i},h_{y,i})$, $(-h_{x,i-1},-h_{y,i-1})$ from link i+1 and link i–1, respectively. u_{i-1} and u_i are actuator torques exerted on link i from link i-1 and link i+1, respectively. The following given matrices and vectors are used in motion equations of the snake robot.

$$\mathbf{A}=\begin{bmatrix}1 & 1 & & \\ & \cdot & \cdot & \\ & & \cdot & \cdot \\ & & & 1 \quad 1\end{bmatrix}_{(n-1)xn},\ \mathbf{D}=\begin{bmatrix}1 & -1 & & \\ & \cdot & \cdot & \\ & & \cdot & \cdot \\ & & & 1 \quad -1\end{bmatrix}_{(n-1)xn}.$$

$$\mathbf{E}=\begin{bmatrix}\mathbf{e} & \mathbf{0}_{nx1} \\ \mathbf{0}_{nx1} & \mathbf{e}\end{bmatrix},\ \mathbf{e}=[1,...,1]^T.$$

$$\sin\boldsymbol{\theta}=[\sin\theta_1,...,\sin\theta_n]^{\mathrm{T}},\ \cos\boldsymbol{\theta}=[\cos\theta_1,...,\cos\theta_n]^{\mathrm{T}},\ \dot{\boldsymbol{\theta}}^2=[\dot{\theta}_1^2,...,\dot{\theta}_n^2]^{\mathrm{T}}.$$

$$\mathbf{S}_{\boldsymbol{\theta}}=diag(\sin\boldsymbol{\theta}),\ \mathbf{C}_{\boldsymbol{\theta}}=diag(\cos\boldsymbol{\theta}),\ \mathbf{M}=diag(m_1,...,m_n),\ \mathbf{J}=diag(J_1,...,J_n).$$

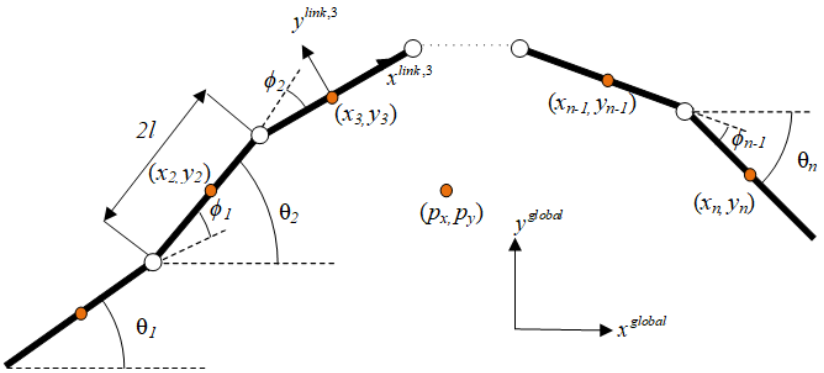


Fig. 1. *n* links snake robot diagram

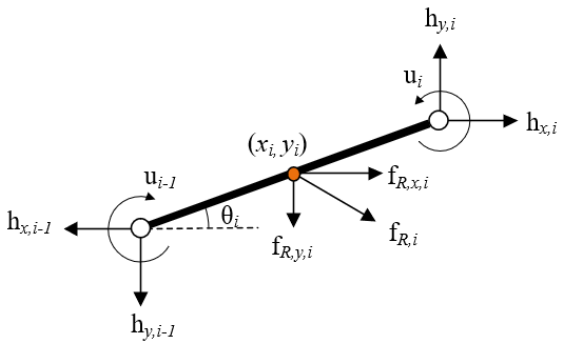


Fig. 2. The forces and torques acting on each link of the snake robot

The position of the snake robot **p** defined by its global CM is given in (1).

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{1}{nm}\sum_{i=1}^{n} m_i x_i \\ \frac{1}{nm}\sum_{i=1}^{n} m_i y_i \end{bmatrix} = \frac{1}{n}\begin{bmatrix} \mathbf{e}^T\mathbf{X} \\ \mathbf{e}^T\mathbf{Y} \end{bmatrix} . \quad \textbf{(1)}$$

where m is sum of mass of each link, $\mathbf{X} = [x_1, \ldots, x_n]^T$ and $\mathbf{Y} = [y_1, \ldots, y_n]^T$.

Frictions in normal and tangential directions between the snake robot and the ground play a crucial role especially during lateral undulation motion of the robot [23]. The friction forces on all links ($\mathbf{f}_R$) are given in (2) according to viscous friction model. A distributed contact model is adopted in the formulation of friction force.

$$\mathbf{f}_R = \begin{bmatrix} \mathbf{f}_{R,x} \\ \mathbf{f}_{R,y} \end{bmatrix} = -\begin{bmatrix} \mathbf{C}_\theta & -\mathbf{S}_\theta \\ \mathbf{S}_\theta & \mathbf{C}_\theta \end{bmatrix}\begin{bmatrix} \mathbf{C_t M} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_n M} \end{bmatrix}\begin{bmatrix} \mathbf{C}_\theta & \mathbf{S}_\theta \\ -\mathbf{S}_\theta & \mathbf{C}_\theta \end{bmatrix}\begin{bmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{Y}} \end{bmatrix} . \quad \textbf{(2)}$$

$$\mathbf{C_t} = diag(c_{t_1}, ..., c_{t_n}) , \ \mathbf{C_n} = diag(c_{n_1}, ..., c_{n_n}) .$$

where c_{ti} and c_{ni} represent friction coefficients in tangential and normal directions of the link $i \in \{1, \ldots, n\}$, respectively. $\dot{\mathbf{X}}$ and $\dot{\mathbf{Y}}$ are the linear velocities of the links.

The torque applied to CM of the link because of friction force can be defined as in (3).

$$\boldsymbol{\tau}_R = -\mathbf{C}_n \mathbf{J}\dot{\boldsymbol{\theta}} . \quad \textbf{(3)}$$

The dynamic model of the snake robot is described in matrix form as in (4).

$$\begin{bmatrix} \mathbf{M}_\theta & \mathbf{0} \\ \mathbf{0} & m\mathbf{I} \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{\theta}} \\ \ddot{\mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{W}\dot{\boldsymbol{\theta}}^2 \\ \mathbf{0} \end{bmatrix} - \mathbf{C}\begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{D}^T \\ \mathbf{0} \end{bmatrix} u . \quad \textbf{(4)}$$

$$\mathbf{M}_\theta = \mathbf{J} + \mathbf{M}\mathbf{L}^2\mathbf{S}_\theta\mathbf{V}\mathbf{S}_\theta + \mathbf{M}\mathbf{L}^2\mathbf{C}_\theta\mathbf{V}\mathbf{C}_\theta .$$

$$\mathbf{W} = \mathbf{M}\mathbf{L}^2\mathbf{S}_\theta\mathbf{V}\mathbf{C}_\theta - \mathbf{M}\mathbf{L}^2\mathbf{C}_\theta\mathbf{V}\mathbf{S}_\theta .$$

$$\mathbf{V} = \mathbf{A}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{A} , \ \mathbf{K} = \mathbf{A}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D} , \ \mathbf{N} = \mathbf{K}^T\mathbf{L} .$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_n\mathbf{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{L}^T \\ \mathbf{E}^T \end{bmatrix}\mathbf{R}\begin{bmatrix} \mathbf{C}_t\mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_n\mathbf{M} \end{bmatrix}\mathbf{R}^T\begin{bmatrix} \mathbf{L} & \mathbf{E} \end{bmatrix} .$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{S}_\theta\mathbf{N}^T & -\mathbf{C}_\theta\mathbf{N}^T \end{bmatrix}^T .$$

3. Optimization of Motion

An optimization framework for the efficient motion of the snake robot is presented in this paper. The block diagram of the proposed method is given in Fig. 3. The parts of the system are the snake robot model in contact with the environment presented in Section II, a motion pattern generator, a joint controller, and an optimization algorithm. For different gait parameters, the optimization algorithm evaluates the objective functions calculated by using the forward velocity and average power consumption obtained by simulating the dynamic model of the snake robot.

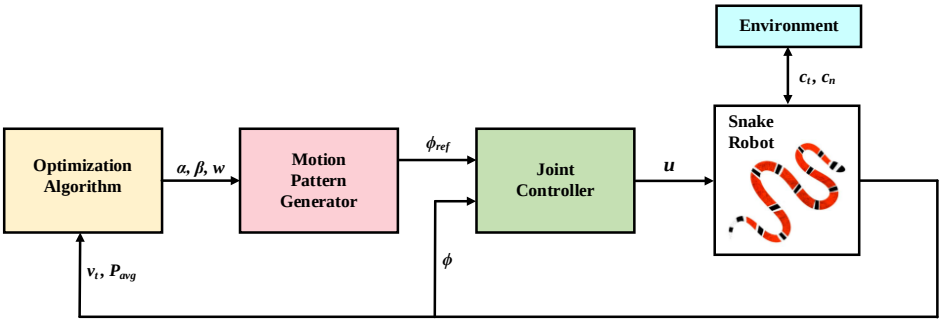


Fig. 3. Block diagram of the proposed method

3.1. Motion Pattern Generator

Lateral undulation is the most common seen gait in almost all snake species. To achieve this motion pattern for snake robot, a sinusoidal reference signal given in (5) is applied to each joint of the snake robot [24].

$$\phi_{i,ref} = \alpha \sin(\omega t + (i-1)\beta) + \gamma \,. \tag{5}$$

where α, ω and β are parameters used to determine amplitude, angular frequency, and phase shift between the neighbourhood joints of the gait pattern. The joint offset used to control the direction of the locomotion is chosen as $\gamma = 0$ in this study.

3.2. Joint Controller

A PD controller is used to let the snake robot track the reference joint angle ϕ_{ref} and the control input for joint i is given in (6). The derivative of ϕ_{ref} is obtained by passing the reference signal through a second order low pass filter [22]. The filter parameters are set to ω_n=25 and ξ=1 in this study.

$$u_i = k_p(\phi_{ref} - \phi) + k_D(\dot{\phi}_{ref} - \dot{\phi}) \,. \tag{6}$$

where $k_p > 0$ and $k_D > 0$ are controller gains.

3.3. Optimization Algorithm

In this paper, two different multi-objective SOS based algorithms are presented for energy efficient locomotion of the snake robot. These algorithms are the weighted sum method based multi-objective SOS algorithm and a fast non-dominated sorting multi-objective SOS algorithm. Details of these algorithms are given below.

An overview of the symbiotic organism search (SOS) algorithm

Developed by inspiring from the symbiotic interaction strategies between organisms in the ecosystem, SOS algorithm is a robust and effective metaheuristic method [25]. Like other heuristic optimization techniques, SOS starts with a randomly produced population of organisms named as the ecosystem and each organism representing a candidate solution is iteratively used to find optimal global solution. The algorithm consists of three phases, namely mutualism, commensalism and parasitism which are biological interactions seen mostly in the real world. Three phases of SOS algorithm are carried out for each organism selected in order by beginning from the first organism X_i in the ecosystem. The organism X_i interacts with a different organism X_j randomly selected from the ecosystem in all three phases. According to the type of the interaction, the new candidate solutions are generated in the three phases and in case these new solutions are better than previous ones, the old solutions are updated. The process is repeated until the stopping criterion, which is the maximum number of iterations, is achieved.

Mutualism phase. In this phase, because both organisms provide advantage from the association, the new candidate solutions for both X_i and X_j organisms are generated as formulated in (7) and (8). Both organisms do not benefit equally from the interaction. BF_1 and BF_2 parameters are used in the equations to reflect this situation and selected randomly 1 or 2.

$$X_{inew} = X_i + rand(0,1)*(X_{best} - MV*BF_1). \quad \textbf{(7)}$$

$$X_{jnew} = X_j + rand(0,1)*(X_{best} - MV*BF_2). \quad \textbf{(8)}$$

where the X_{best} is the organism which has best fitness value in the ecosystem. MV is the mutual vector and defined as in (9).

$$MV = \frac{X_i + X_j}{2}. \quad \textbf{(9)}$$

Commensalism phase. In commensalism phase, only one organism X_i benefits and the other X_j is not affected. Hence, the new candidate solution only for X_i is produced as in (10).

$$X_{inew} = X_i + rand(-1,1)*(X_{best} - X_j). \quad \textbf{(10)}$$

Parasitism phase. In parasitism phase, one of the two different species benefits from the other by damaging it. A parasite vector is created by duplicating X_i and then modifying it randomly in the search space. X_j is selected randomly from the ecosystem and used to serve as a host to the parasite vector.

The weighted sum method based multi-objective symbiotic organism search algorithm

This method considers a multi-objective optimization problem as a single-objective optimization problem. This single objective function is created by summing each objective function multiplied with a weight factor. The sum of the weights of all objectives should be 1.

The energy efficient locomotion problem of the snake robot is a mixed optimization problem including both maximization of the forward velocity of the snake robot and minimization of the average power consumption. For this reason, these objectives should be converted into one type while combining them into a single objective function. The single objective function used in this study is given in (11).

$$J = (1-w)\,(P_{avg})_{sc} - w(v_f)_{sc}\,. \tag{11}$$

where P_{avg} is the average power consumption and v_f is the forward velocity of the snake robot. These indices are calculated as in (12) and (13), respectively. w is the weight factor and determines priority of the P_{avg} and v_f in the objective function. The subscript sc represents the scaled values of power consumption and forward velocity.

$$P_{avg} = \frac{1}{T}\int_0^T \left(\sum_{i=1}^{n-1} \left| u_i(\mathrm{t})\,\dot{\phi}_i(\mathrm{t}) \right| \right) dt\,. \tag{12}$$

where T is the simulation time. The actuation torque u_i are calculated given as in (6) while the angular velocity for joint i is found by using derivative of the expression $\phi_i = \theta_i - \theta_{i+1}$.

$$v_f = \frac{\sqrt{(p_x(\mathrm{T}) - p_x(0))^2 + (p_y(\mathrm{T}) - p_y(0))^2}}{T}\,. \tag{13}$$

where $(p_x(0), p_y(0))$ and $(p_x(\mathrm{T}), p_y(\mathrm{T}))$ denote initial and final positions of CM of the snake robot.

P_{avg} and v_f should be firstly scaled by dividing by their maximum values as defined in (14). The maximum values are determined by performing the optimization at w=1.

$$(P_{avg})_{sc} = \frac{P_{avg}}{(P_{avg})_{\max}}, \quad (v_f)_{sc} = \frac{v_f}{(v_f)_{\max}}\,. \tag{14}$$

The ecosystem matrix is obtained as in (15) by producing n_o random organisms with dimension n_d within the lower and upper bounds. Each element of the organism vector indicates parameters of the gait to be applied to the snake robot. In this problem, the dimension of the organisms is determined as 3 because the gait parameters to be

optimized are α, β and ω. The fitness value of each organism in the ecosystem is calculated according to the objective function in (11).

$$ecosystem = \begin{bmatrix} organism_1 \\ \vdots \\ organism_{n_o} \end{bmatrix} = \begin{bmatrix} \alpha_{1,1} & \beta_{1,2} & \omega_{1,3} \\ \vdots & \ddots & \vdots \\ \alpha_{n_o,1} & \beta_{n_o,2} & \omega_{n_o,3} \end{bmatrix}_{n_o x 3} . \quad \textbf{(15)}$$

In conclusion, this optimization problem can be expressed as that the objective function in (11) is minimized subject to bound constraints given in (16) representing limitations of the servo motor and the parameters of the sinusoidal motion pattern. If these values go over the limits for any organisms, the corresponding organism is removed from the ecosystem by determining its fitness value as a high value.

$$\begin{aligned} &\min_{\alpha,\beta,\omega} \; J = \left[P_{avg}, -v_f \right] \\ &\text{s.t}: \; \left|\phi_{i,ref}\right| \le \phi_i^{max}, \left|\dot{\phi}_{i,ref}\right| \le \dot{\phi}_i^{max}, \left|u_i\right| \le u_i^{max} \\ &0 \le \alpha \le \alpha_{max}, 0 \le \beta \le \beta_{max}, 0 \le \omega \le \omega_{max} \end{aligned} \quad \textbf{(16)}$$

Fast non-dominated sorting multi-objective symbiotic organisms search algorithm (FNSMOSOS)

FNSMOSOS presents a set of Pareto optimal solutions which is non-dominated with respect to each other instead of a unique optimal solution like in the weighted sum method based MOSOS. It uses fast non-dominated sorting (FNS) technique and crowding distance (CD) technique to preserve elitism and maintain diversity. The flowchart of proposed FNSMOSOS algorithm to solve the optimal locomotion problem of the snake robot is given in Fig. 4. This algorithm initializes with an ecosystem which is generated in the same way as the weighted sum method based MOSOS. For each organism, two separate objective functions defined in (12) and (13) used to minimize the average power consumption and maximize the forward velocity of the snake robot are evaluated and the fitness values of the any organisms which do not satisfy the constraints described in (16) are set to a high value. The new candidate organisms are generated in mutualism, commensalism, and parasitism phases. These phases are carried out as in the weighted sum method based MOSOS. The only difference is to be also move the dominated organisms to an advanced ecosystem for selecting next generation ecosystem alongside the non-dominated organisms are kept in the current ecosystem as a result of the comparing the fitness values of new organisms with their previous fitness values. Afterwards, the current ecosystem (n_o) and the advanced ecosystem ($4n_o$) are combined. The obtained combined ecosystem ($5n_o$) is sorted by using FNS technique to select the best n_o organisms for next generation.

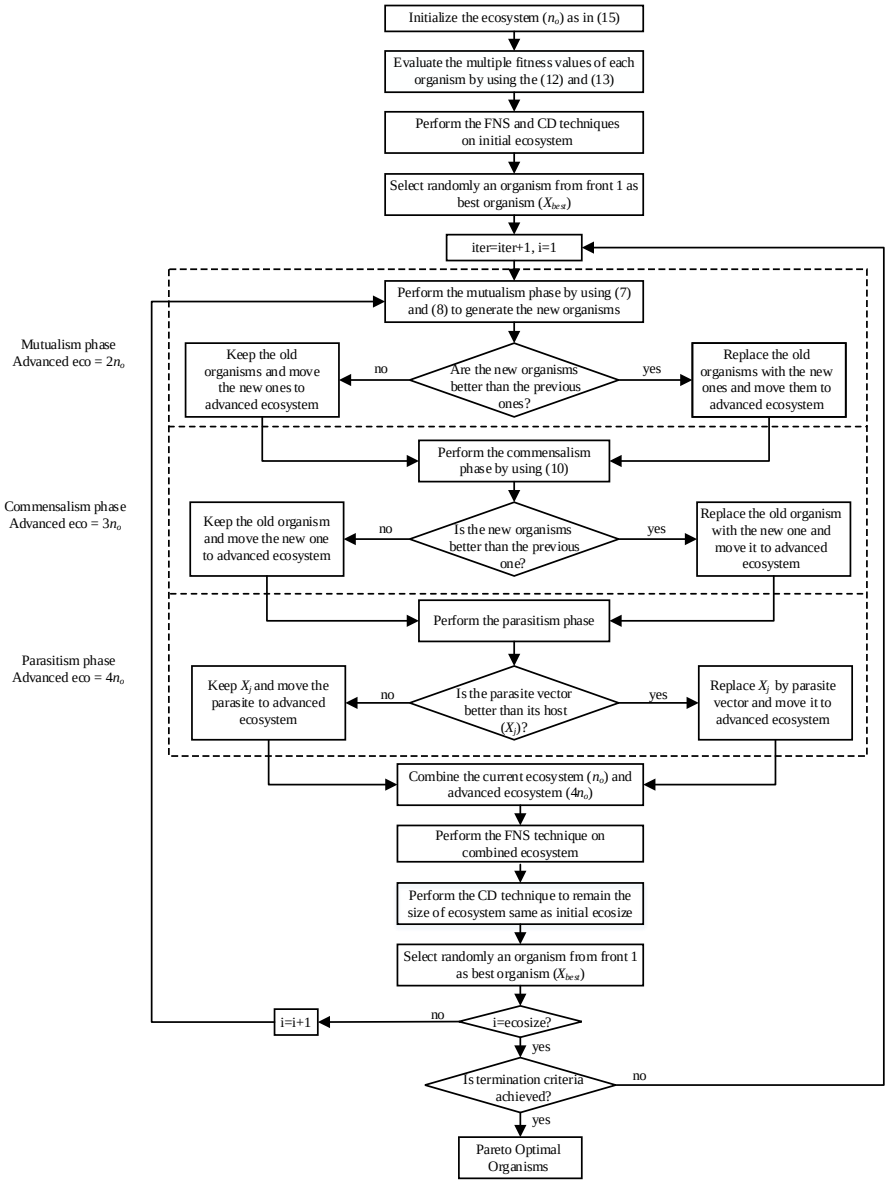


Fig. 4. The flowchart of proposed FNSMOSOS algorithm

In FNS technique, the organisms are grouped into fronts by comparing fitness values of each organism with all other organisms in the ecosystem. For determining front of each organism, the first step is to calculate domination count n_p defined the number of organisms that dominate the organism p, and S_p which is a set of organisms dominated by organism p. Each organism with $n_p = 0$ is assigned to first front (F1) also known as Pareto front and they are better than others for one objective at least. The next step is to visit S_p for each organism belonging to the Pareto front and reduce the domination count

of each organism in S_p by one. Thus, the organisms whose n_p becomes zero are assigned to second front (F2). This operation goes on until all organisms in the ecosystem are assigned to a front. A general scheme of non-dominated sorting procedure is shown in Fig. 5. More details of the fast non-dominated sorting technique can be found in [26].

After FNS technique is applied, each front beginning from the first front is assigned to the new ecosystem one by one until the ecosystem size reaches up to n_o. If the addition of an entire front to the ecosystem causes the size of the ecosystem to exceed, the best organisms in this front are selected by crowding distance technique. For this, all organisms in this front are firstly sorted in ascending order for each objective function. Thereafter, the CD values of boundary organisms with smallest (i=1) and largest (i=l) fitness values are assigned to an infinite value and for other intermediate organisms (i=2 to l-1), the CD value is calculated as normalized difference in fitness value of two neighbouring organisms (i+1 and i-1), given as in (17). This computation is performed for each objective function j (j=1,2,…,m) and the total crowding distance value of the organism i is found by summing the individual distance values corresponding to each objective function, given as in (18). For a problem involving two objective functions, computation of the crowding distance for the organism i is shown in Fig. 6.

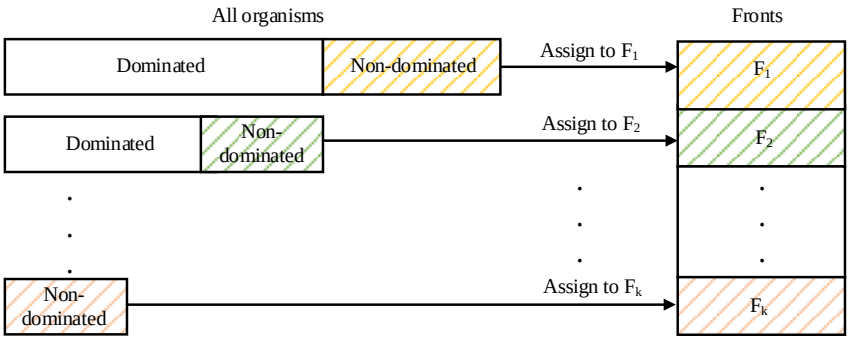


Fig. 5. A general scheme of non-dominated sorting procedure

$$d_j^i = \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{\max} - f_j^{\min}} \,. \tag{17}$$

$$d_i = d_1^i + d_2^i + ... + d_m^i \,. \tag{18}$$

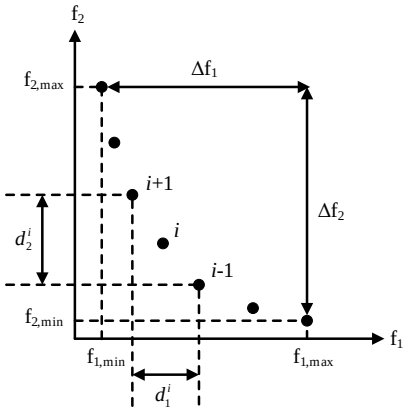


Fig. 6. Computation of the crowding distance for the organism *i*

4. Optimization of Motion

In this section, the results of the proposed methods for energy-efficient locomotion of the snake robot are presented. A five link wheel-less snake robot which has identical links is used in this study and the parameters of the robot modelled are given in Table 1.

Table 1. Model parameters of the snake robot

Parameters	Values
Number of links	$n = 5$
The length of a link	$2l = 0.18$ m
Mass of each link	$m = 0.8$ kg
Moment of inertia of each link	$J = 0.00216$ kgm^2
Friction coefficient	$c_t = 0.1$; $c_n = 10$
The parameters of the PD controller	$k_p = 20$; $k_D = 5$

4.1. Performance of the weighted sum method based MOSOS

The first method used to obtain the optimal locomotion of the snake robot is the weighted sum method based MOSOS algorithm. The initial parameters of the algorithm are set as in Table 2. The dimension of the problem should be equal to the numbers of parameters to be optimized. Thus, this parameter is determined as 3 for energy efficient locomotion of snake robot. Because the determination of the number of organisms in the ecosystem is not based on any general rule, it is empirically set to n_o =15. Similarly, the maximum iteration number is also empirically selected as N=20 by considering its effect on the finding the optimal solution. The physical constraints of joints are determined based on the servo motor (HSR-5990TG) while the lower and upper bounds of the search space are determined according to the minimum and maximum values of the

parameters of the sinusoidal motion pattern. For the scaling of P_{avg} and v_f in the objective function, their maximum values are found as v_f = 0.39 m/s and P_{avg} = 1.20 W, respectively.

Table 2. Optimization parameters

Parameters	Values
Dimension of organism	$n_d = 3$
Number of organisms	$n_o = 15$
Iteration number	N = 20
Upper bounds of the gait parameters	$\alpha_{max} = 90^\circ$, $\beta_{max} = 90^\circ$, $\omega_{max} = 210\ ^\circ/s$
The physical constraints of joints	$\phi_i^{max} = 90^\circ$, $\dot{\phi}_i^{max} = 429^\circ/s$, $u_i^{max} = 2.3$ Nm

The forward velocity and the average power consumption are obtained by changing the weight factor between 0.1 and 1 with a step size of 0.1. The effect of weight factor on the forward velocity and the average power consumption are seen in Fig. 7 (a). From this figure, it is seen that the forward velocity increases as w value increases and thus the average power consumption also increases due to the increasing velocity. The snake robot can achieve maximum forward velocity v_f =0.39 m/s with the corresponding maximum average power consumption P_{avg} =1.20 W. As it was expected, the maximum values are obtained while w=1. A set of Pareto optimal solutions obtained is seen in Fig. 7 (b). This figure indicates that the obtained solution set has a good coverage but not distributed very uniformly. This finding proves that the weights uniformly distributed cannot always present a uniform distribution of the Pareto optimal solutions.

The obtained results for the optimal gait parameters for each weight factor are also presented in Table 3. As seen in Table 3, the average power consumption of the snake robot significantly decreases from P_{avg} = 1.20 W to P_{avg} =0.42 W when the weight factor is changed from w =1 to w =*0.6.* On the other hand, there is only a small decrease in the forward velocity from v_f = 0.39 m/s to v_f = 0.34 m/s. Hence, a 65.08% reduction in the average power consumption of the snake robot can be obtained by sacrificing a 13.77% reduction in the forward speed. If a lower reduction in forward velocity is desired, by choosing the gait parameters at w= 0.8 in which the forward velocity decreases only by 4.11%, the average power consumption can be reduced by 43.27%. This table is useful for decision makers considering system requirements and can be used to find the optimal trade-off between the forward velocity of the snake robot and the average power consumption.

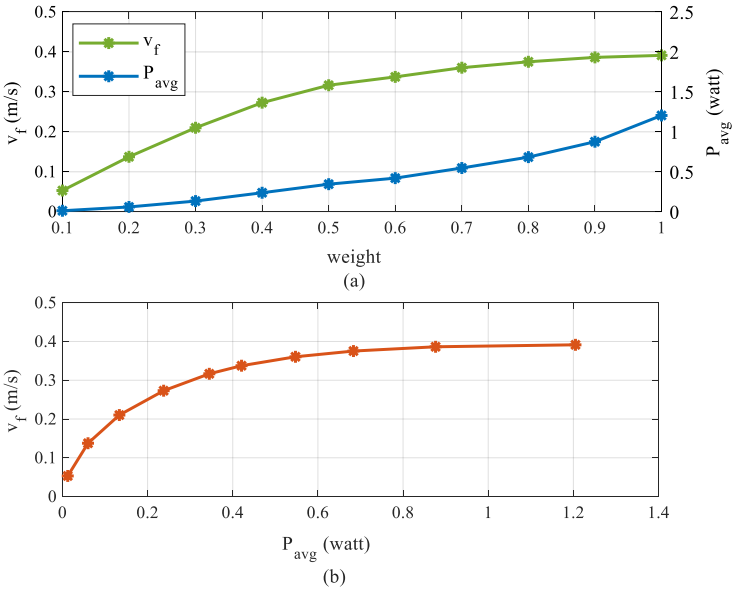


Fig. 7. (a) The effect of weight factor on the forward velocity and the average power consumption **(b)** Pareto optimal solutions

Table 3. Obtained results for the optimal gait parameters for each weight factor

w	α (deg)	ω (deg/s)	β (deg)	v_f (m/s)	P_{avg} (W)
0.1000	15.2200	70.9344	90.0000	0.0531	0.0125
0.2000	18.1374	125.4008	78.1864	0.1375	0.0601
0.3000	19.2169	145.2789	61.1747	0.2102	0.1337
0.4000	20.2889	189.7583	62.4138	0.2730	0.2377
0.5000	21.8587	208.3734	59.8795	0.3166	0.3450
0.6000	23.2913	210.0000	56.6467	0.3375	0.4207
0.7000	25.9909	210.0000	53.3036	0.3605	0.5473
0.8000	28.2939	210.0000	50.1877	0.3753	0.6835
0.9000	31.0264	210.0000	46.7528	0.3863	0.8764
1.0000	34.8682	210.0000	42.6450	0.3914	1.2049

The change of the forward velocity and average power consumption depending on the change of the optimal gait parameters in (5) are illustrated in Fig. 8. This figure shows that an increase of the parameter α also results in an increase of the forward velocity and the average power consumption. On the other hand, it is seen that the parameter β has the opposite effect of the parameter α. This means that the forward velocity and the average power consumption decrease as parameter β increases. Another important finding obtained from Fig. 8 is that the optimal value of the parameter α is in the range from 15º to 35º, while parameter β is in the range from 40º to 65º. When we examine the effect of parameter ω on the forward velocity of the snake robot, it is seen that the parameter ω is at the maximum value 210º/s for most of the weight factors.

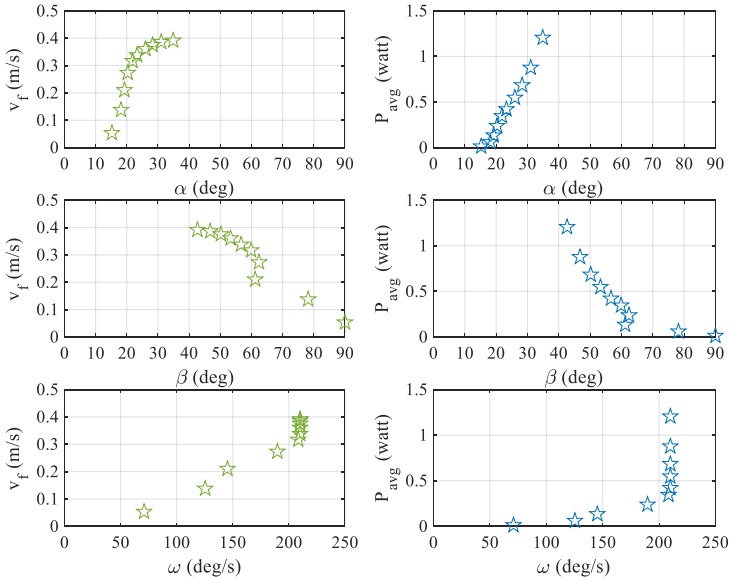


Fig. 8. The change of the forward velocity and the average power consumption versus optimal gait parameters

According to these observations, we can determine the lower and upper bounds of the parameter α and β in a narrower range and the size of the search space can be reduced from 3 to 2 by setting w to the maximum value. Thus, the optimization process applied to find optimal gait parameters for snake robot can become more efficient and less costly computation. Finally, the position of the CM of snake robot for each of the weight factor is presented in Fig. 9.

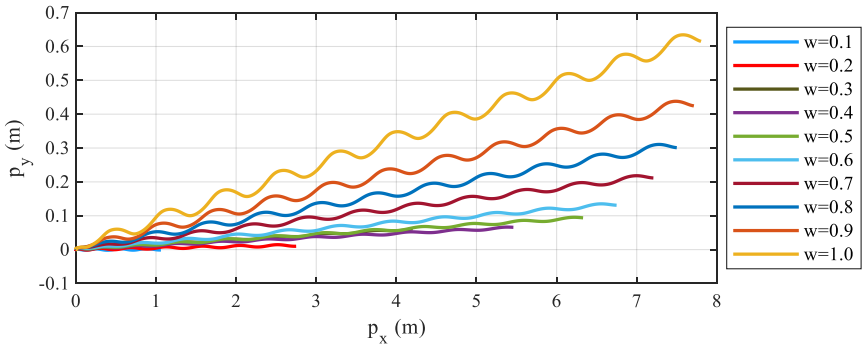


Fig. 9. The position of the CM of snake robot for each of the weight factor

4.2. Performance of the FNSMOSOS

For finding the parameters of the most efficient motion pattern for snake robot, the second proposed method in this paper is FNSMOSOS. In this algorithm, the number of organisms in the ecosystem and maximum iteration number are 100 and 20, respectively. A set of Pareto optimal solutions obtained with FNSMOSOS is seen in Fig. 10. This figure demonstrates FNSMOSOS achieves convergence to optimal Pareto front with a good diversity. When compared this Pareto front with that in Fig.7(b), it is seen that a more uniform distribution of solutions is achieved on the Pareto front with FNSMOSOS.

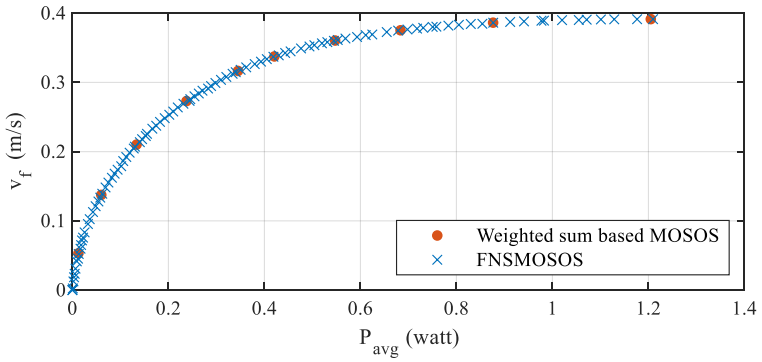


Fig. 10. Pareto optimal solutions obtained by FNSMOSOS and weighted sum based MOSOS

In this algorithm, the maximum forward velocity is obtained v_f =0.39 m/s as in the weighted sum method based MOSOS algorithm. The maximum average power consumption corresponding to maximum forward velocity is P_{avg} =1.20 W. The obtained results for the some of Pareto optimal gait parameters are given in Table 4. As seen from the table, FNSMOSOS presents more set of different solutions. Therefore, decision maker can balance between the forward velocity of the snake robot and the average power consumption by selecting the optimal gait parameters among more options according to systems requirements. For example, solution 22 in comparison with the solution 30 could be a good option by providing a 51.23% reduction in the average power consumption with only a small decrease of 6.56% in the forward speed. If the available power of the system is more limited, the average power consumption can be reduced by 74.60% by choosing the gait parameters in the solution 17. In this case, the forward velocity decreases only by 22.74%.

The effect on forward velocity and average power consumption of Pareto optimal gait parameters are presented in Fig 11. These curves are like those in Fig. 8 obtained with the weighted sum method based MOSOS algorithm. This finding indicates that the two different algorithms based MOSOS produce stable results for optimal locomotion of the snake robot.

Table 4. Obtained results for the some of Pareto optimal gait parameters

Solution	α (deg)	ω (deg/s)	β (deg)	v_f (m/s)	P_{avg} (W)
1	5.7348	41.8266	90.0000	0.0073	0.0012
2	13.7767	44.2929	86.5326	0.0321	0.0068
3	16.2800	70.4723	81.4882	0.0670	0.0189
4	17.5858	117.6275	79.4923	0.1231	0.0494
5	17.0468	124.0257	75.5258	0.1350	0.0585
6	17.0036	138.8863	73.4650	0.1552	0.0758
7	18.1607	138.9609	70.7857	0.1707	0.0898
8	19.6000	146.8214	66.5524	0.2016	0.1236
9	19.5987	149.1822	63.4268	0.2127	0.1381
10	19.8998	155.0219	60.0336	0.2312	0.1634
11	20.2750	180.2280	68.5126	0.2429	0.1834
12	19.1930	186.5953	63.2537	0.2568	0.2082
13	20.4790	190.5852	65.6480	0.2661	0.2246
14	19.9386	197.8035	63.8226	0.2757	0.2438
15	20.1859	206.7052	63.8109	0.2888	0.2725
16	20.1264	210.0000	63.3527	0.2934	0.2842
17	20.6437	210.0000	61.9566	0.3023	0.3058
18	21.9738	210.0000	61.6950	0.3143	0.3391
19	22.7583	210.0000	58.5487	0.3289	0.3868
20	23.7820	210.0000	55.7554	0.3428	0.4449
21	25.3091	210.0000	54.0794	0.3555	0.5137
22	27.1476	210.0000	53.3289	0.3656	0.5872
23	28.9861	210.0000	53.7887	0.3709	0.6459
24	29.3437	210.0000	50.1804	0.3784	0.7265
25	30.1980	210.0000	47.3687	0.3840	0.8229
26	31.3692	210.0000	45.7991	0.3876	0.9169
27	33.3200	210.0000	45.4232	0.3900	1.0281
28	33.9895	210.0000	44.3541	0.3909	1.0971
29	33.9215	210.0000	42.8199	0.3912	1.1423
30	34.9466	210.0000	42.8074	0.3913	1.2040

Table 5. Different environments

Environment	c_t	c_n
1	0.01	0.2
2	0.01	0.5
3	0.01	1.0
4	0.05	0.2
5	0.05	0.5
6	0.05	1.0
7	0.1	0.2
8	0.1	0.5
9	0.1	1.0

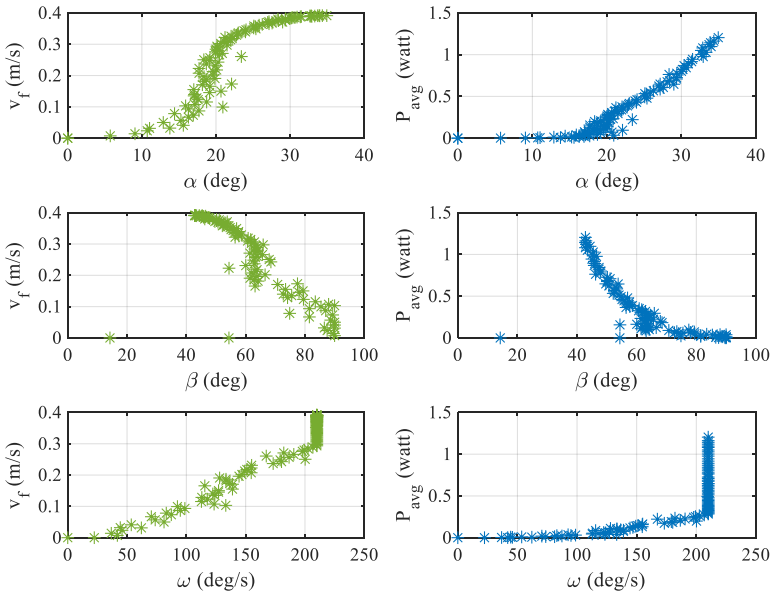


Fig. 11. The change of the forward velocity and the average power consumption versus Pareto optimal gait parameters

In this paper, the optimal gait parameters for the adaptive locomotion of the snake robot in different environment conditions are also investigated by using FNSMOSOS. For this purpose, nine different environments representing different surfaces in a quite wide range from glass to wood are used. The friction coefficients representing the environments are given in Table 5 and the obtained Pareto optimal solutions for each environment are presented in Fig. 12.

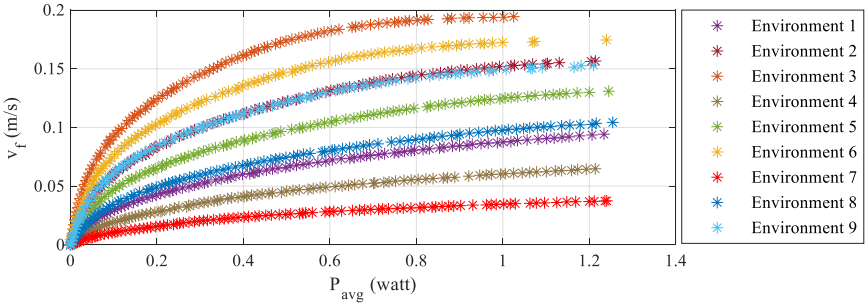


Fig. 12. Pareto optimal solutions of FNSMOSOS for each environment

As seen from Fig. 12, FNSMOSOS presents good performance for each environment in achieving optimal solutions converging to the Pareto front. As it was expected, the snake robot has reached to a higher forward velocity in environments where c_n is high such as environment 3, environment 6 and environment 9. Moreover, it is seen that if c_n is constant, the snake robot can move faster in environments where the ratio between the two friction coefficients c_n/c_t is high such as environment 3.

The effect of different friction coefficients acting in the direction normal and tangential to snake robot on the forward velocity and average power consumption according to the optimal gait parameters are presented in Fig.13-Fig.18. According to these figures, for the nine environments, the optimal value of the parameter α varies in the range between 20º and 50º, while parameter β varies in the range between 55º and 90º. Although the parameter ω varies over a wider range between 50º and 210º, the snake robot has achieved its maximum forward velocity when the parameter ω is at the maximum value 210º/s in all environments. Moreover, these figures show that snake robot should modify its gait parameters to maintain its efficient locomotion in different ground conditions. The forward velocity and average power consumption in environments where c_n changes while c_t is constant are presented in Fig. 13, Fig. 14, and Fig. 15 according to optimal α, β and ω parameters, respectively. According to Fig. 13 and Fig. 15, the optimal α and ω parameters increase as c_n decreases because the motion in slippery environments where c_n is low requires more friction force. Based on these results, an important finding is concluded that snake robot should move with greater amplitude and frequency to get more friction force in forward direction in such environments. On the other hand, Fig 14 shows that the optimal β parameter decreases in the same environment conditions. Similarly, Fig. 16, Fig. 17 and Fig. 18 demonstrate the forward velocity and average power consumption in environments where c_t changes while c_n is constant according to optimal α, β and ω parameters, respectively. The decreasing of c_t effects the optimal gait parameters in the opposite direction of the decreasing of c_n. As c_t decreases, the optimal α and ω parameters decrease while the optimal β parameter increases. However, it is seen that the changing of c_t has less of an effect on the optimal gait parameters than the changing of c_n.

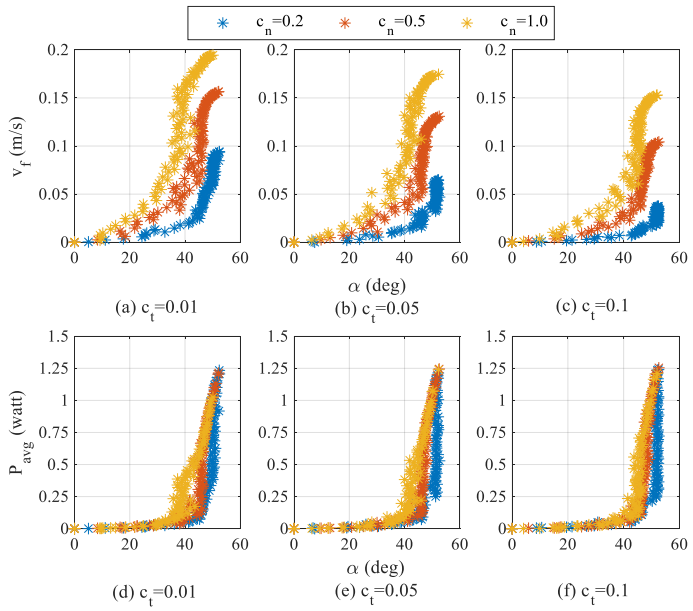


Fig. 13. The effect of different friction coefficients acting in the direction normal to snake robot on the forward velocity and average power consumption according to the optimal α parameter

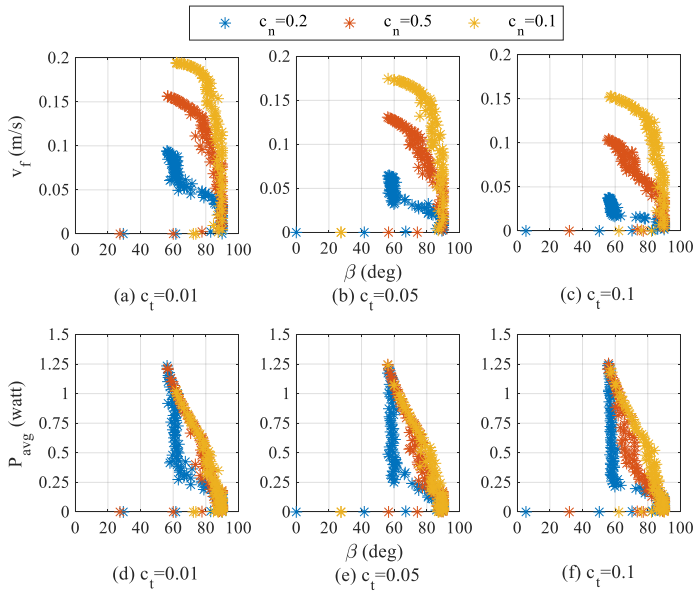


Fig. 14. The effect of different friction coefficients acting in the direction normal to snake robot on the forward velocity and average power consumption according to the optimal β parameter

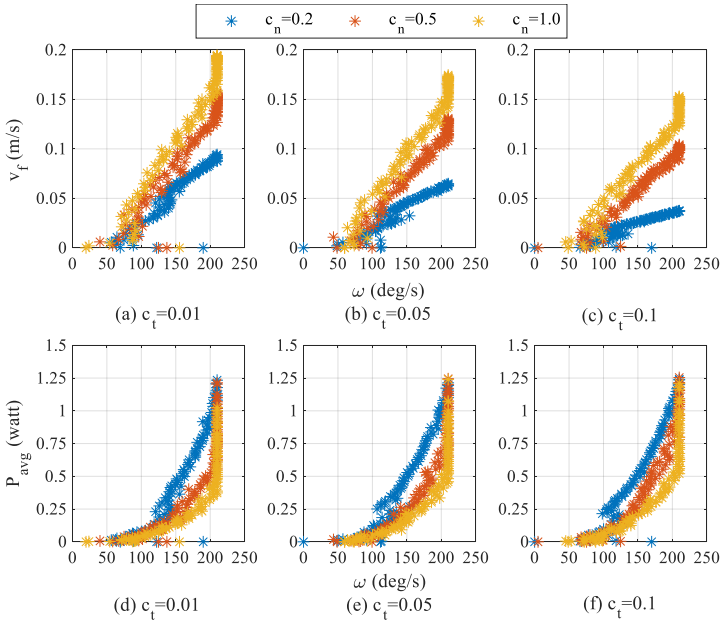


Fig. 15. The effect of different friction coefficients acting in the direction normal to snake robot on the forward velocity and average power consumption according to the optimal ω parameter

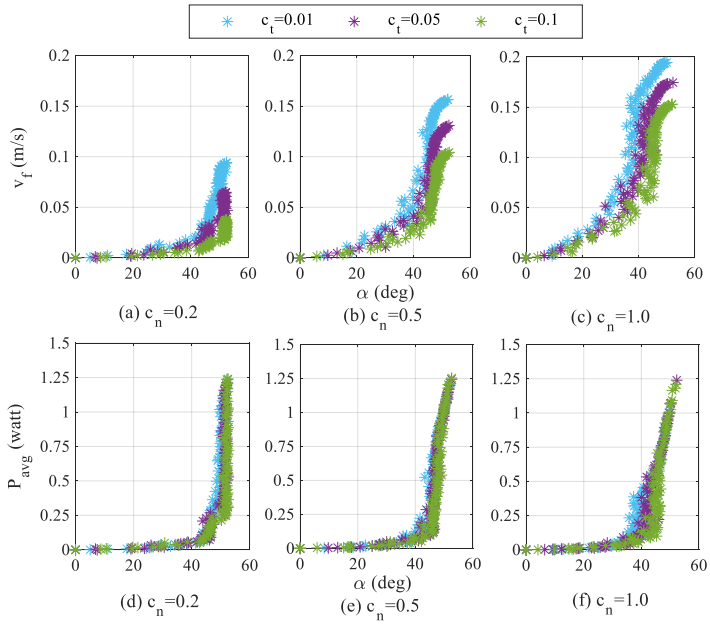


Fig. 16. The effect of different friction coefficients acting in the direction tangential to robot on the forward velocity and average power consumption according to the optimal α parameter

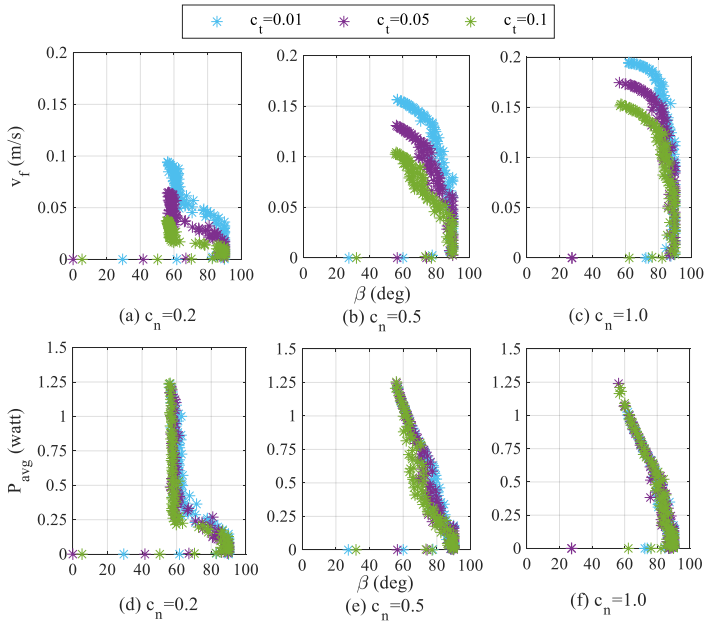


Fig. 17. The effect of different friction coefficients acting in the direction tangential to robot on the forward velocity and average power consumption according to the optimal β parameters

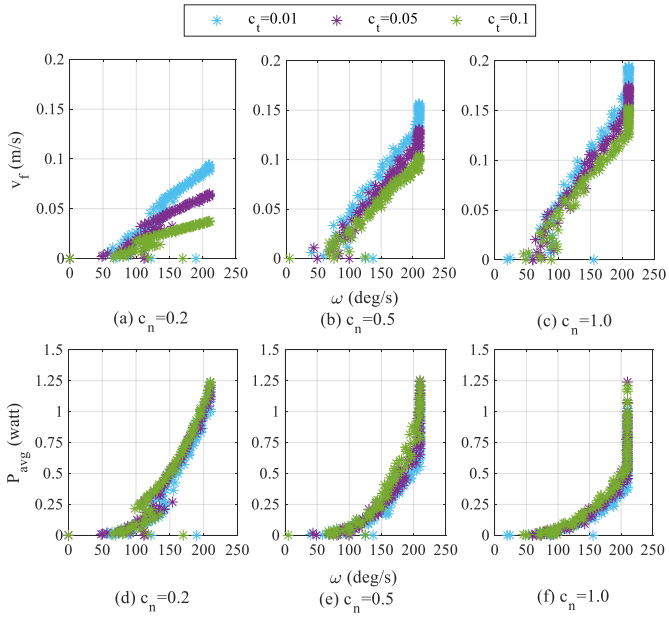


Fig. 18. The effect of different friction coefficients acting in the direction tangential to robot on the forward velocity and average power consumption according to the optimal ω parameters

5. Conclusion

This paper has investigated the optimal gait parameters giving appropriate forward velocity for the lower power consumption of the snake robot. The necessary trade-off between the forward velocity and power consumption for optimally efficient locomotion of the snake robot is obtained by using two different algorithms based MOSOS. From the obtained results, it is seen that these two algorithms produce stable results for optimal locomotion of the snake robot. However, FNSMOSOS provides a better distributed solution set when compared with the weighted sum method based MOSOS. Moreover, it generates more different solutions only in a single run. Thus, the operators can easily determine and select the optimal operational strategy from the Pareto front based on the control targets and the available power of the snake robot. In this paper, efficient locomotion of the snake robot is also investigated by considering different environments having a fairly wide friction range. The obtained results are very important to the snake robot maintaining its optimal locomotion in different environmental condition. Thus, this study is useful for developing environmental adaptability and efficient motion of the snake robot which has low motion efficiency due to its friction dependent motion.

References

1. Crespi, A., Ijspeert, A.J.: Online optimization of swimming and crawling in an amphibious snake robot. IEEE Transactions on Robotics, Vol.24, No.1, 75–87. (2008)
2. Hasanzadeh, S., Tootoonchi, A.A.: Ground adaptive and optimized locomotion of snake robot moving with a novel gait. Autonomous Robots, Vol. 28, No.4, 457-470. (2010)
3. Hasanzadeh, S., Tootoonchi, A. A.: Adaptive optimal locomotion of snake robot based on CPG-network using fuzzy logic tuner. IEEE Conference on Robotics, Automation and Mechatronics, 187-192. (2008)
4. Wu, X., Ma, S.: Adaptive creeping locomotion of a CPG-controlled snake-like robot to environment change. Autonomous Robots, Vol.28, No.3, 283-294. (2010)
5. Inoue, K., Sumi, T., Ma, S.: CPG-based control of a simulated snake-like robot adaptable to changing ground friction. IEEE/RSJ International Conference on Intelligent Robots and Systems, 1957-1962. (2007)
6. Ariizumi, R., Matsuno, F.: Dynamic analysis of three snake robot gaits. IEEE Transactions on Robotics, Vol.33, No.5, 1075-1087. (2017)
7. Kelasidi, E., Pettersen, K.Y., Gravdahl, J.T.: Energy efficiency of underwater snake robot locomotion. IEEE 23rd Mediterranean Conference on Control and Automation (MED), 1124-1131. (2015)
8. Kelasidi, E., Jesmani, M., Pettersen, K.Y., Gravdahl, J.T.: Multi-objective optimization for efficient motion of underwater snake robots. Artificial Life and Robotics, Vol. 21, No.4, 411-422. (2016)
9. Kelasidi, E., Jesmani, M., Pettersen, K.Y., Gravdahl, J.T.: Locomotion efficiency optimization of biologically inspired snake robots. Applied Sciences, Vol.8, No.1, 80. (2018)
10. Cao, Z., Zhang, D., Hu, B., Liu, J.: Adaptive path following and locomotion optimization of snake-like robot controlled by the central pattern generator. Complexity. (2019).
11. Bing, Z., Lemke, C., Jiang, Z., Huang, K., Knoll, A.: Energy-efficient slithering gait exploration for a snake-like robot based on reinforcement learning. arXiv preprint arXiv:1904.07788. (2019)

12. Bing, Z., Lemke, C., Jiang, Z., Huang, K., Knoll, A.: Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning. Neural Networks, Vol.129, 323-333. (2020)
13. Zanaty, E. A., Ghiduk, A. S.: A novel approach based on genetic algorithms and region growing for magnetic resonance image (MRI) segmentation. Computer Science and Information Systems, Vol.10, No.3, 1319-1342. (2013)
14. Du, Z., Chen, K.: Enhanced Artificial Bee Colony with Novel Search Strategy and Dynamic Parameter. Computer Science and Information Systems, Vol.16, No.3, 939–957. (2019)
15. Vidal, P. J., Olivera, A. C.: Solving the DNA fragment assembly problem with a parallel discrete firefly algorithm implemented on GPU. Computer Science and Information Systems, Vol.15, No.2, 273-293. (2018)
16. Wang, L., Wu, W., Qi, J., Jia, Z.: Wireless sensor network coverage optimization based on whale group algorithm. Computer Science and Information Systems, Vol.15, No.3, 569-583. (2018)
17. Abdullahi, M., Ngadi, M.A., Dishing, S. I., Usman, M. J.: A survey of symbiotic organisms search algorithms and applications. Neural Computing and Applications, 1–20. (2019)
18. Han, C., Zhou, G., Zhou, Y.: Binary Symbiotic Organism Search Algorithm for Feature Selection and Analysis. IEEE Access, Vol. 7, 166833-166859. (2019)
29. Tran, D.H., Cheng, M.Y., Prayogo, D.: A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem. Knowledge-Based Systems, Vol. 94, 132-145. (2016)
20. Baysal, Y.A., Altas, I.H.: Optimally Efficient Locomotion of Snake Robot. 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Novi Sad, Serbia, 1-6. (2020)
21. Saito, M., Fukaya, M., Iwasaki, T.: Serpentine locomotion with robotic snakes. IEEE Control Systems Magazine, Vol.22, No.1, 64–81. (2002)
22. Liljebäck, P., Pettersen, K.Y., Stavdahl, Ø., Gravdahl, J. T.: Snake robots: Modelling, Mechatronics, and Control. Springer Science & Business Media. (2012)
23. Hu, D., Nirody, J., Scott, T., Shelley, M.: The mechanics of slithering locomotion. Proceedings of the National Academy of Sciences, Vol.106, No.25, 10081–10085. (2009)
24. Hirose, S.: Biologically inspired robots: snake-like locomotors and manipulators. Oxford university press. (1993)
25. Cheng, M.Y., Prayogo, D.: Symbiotic Organisms Search: A new metaheuristic optimization algorithm. Computers and Structures, Vol. 139, 98-112. (2014)
26. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, 182–197. (2002)

Yesim A. Baysal received the B.Sc.E and the M.Sc.E degrees in Electrical and Electronics Engineering from Karadeniz Technical University (KTU), Turkey, in 2012 and 2016, respectively. She has had the position of a full time Research Assistant in Electrical and Electronics Engineering Department at KTU since 2012. She is currently working toward his PhD degree in electrical engineering at KTU. Her research interests include robotics and intelligent control systems.

Ismail H. Altas received his B.Sc.E in Electrical Engineering from Yildiz University, and M.Sc.E from Karadeniz Technical University (KTU), Turkey, in 1985 and 1988, respectively. He obtained his Ph.D. degree from the University of New Brunswick, Canada, in 1993. He is currently a full time Professor in Electrical and Electronics

Engineering Department at KTU. He was awarded as the best outstanding faculty member in engineering for the year 1997 at KTU. He is a member of IEEE Power Engineering, Industrial Electronics, Systems, Man and Cybernetics, Control Systems, and Computational Intelligence Societies. He has been a member of the Chamber of Electrical Engineers in Turkey. He works on intelligent control of power systems and utilization of renewable energy.

Received: February 22, 2021; Accepted: December 01, 2021.

Computer Science and Information Systems 19(1):379–396 https://doi.org/10.2298/CSIS210218063D

On the effectiveness of Gated Echo State Networks for data exhibiting long-term dependencies

Daniele Di Sarli, Claudio Gallicchio, and Alessio Micheli

Department of Computer Science
University of Pisa
Pisa, Italy
daniele.disarli@phd.unipi.it
{gallicch,micheli}@di.unipi.it

Abstract. In the context of recurrent neural networks, gated architectures such as the GRU have contributed to the development of highly accurate machine learning models that can tackle long-term dependencies in the data. However, the training of such networks is performed by the expensive algorithm of gradient descent with backpropagation through time. On the other hand, reservoir computing approaches such as Echo State Networks (ESNs) can produce models that can be trained efficiently thanks to the use of fixed random parameters, but are not ideal for dealing with data presenting long-term dependencies. We explore the problem of employing gated architectures in ESNs from both theoretical and empirical perspectives. We do so by deriving and evaluating a necessary condition for the non-contractivity of the state transition function, which is important to overcome the fading-memory characterization of conventional ESNs. We find that using pure reservoir computing methodologies is not sufficient for effective gating mechanisms, while instead training even only the gates is highly effective in terms of predictive accuracy.

Keywords: echo state networks, gated recurrent neural networks, reservoir computing.

1. Introduction

Several approaches are possible for learning functions over temporal domains. Recurrent Neural Networks (RNNs) represent an effective neural architecture for temporal tasks, with applications in many different domains [20]. When it comes to the training algorithm for RNNs, we distinguish two major approaches.

The first approach is *reservoir computing* [21,31], in which the neural network is studied as a dynamical system and involves the encoding of the input sequence and its temporal features into a fixed size vector in the state space. The peculiar characteristic of the reservoir computing approach is that the weights of the RNN are not trained: instead, only the weights associated to a simple stateless readout layer get trained. This allows for very fast and efficient trainings. A widely known instance of reservoir computing model is represented by Echo State Networks (ESN) [17,16]. ESNs and other reservoir computing models are tightly connected to the concepts of Markovian bias [30,11], fading-memory [12], and contractivity [16], which are fundamental characteristics of their state dynamics.

On the other hand, the second and most popular approach involves training *all* network weights by gradient methods, namely gradient descent. The flexibility of this approach allowed the emergence of architectural variants which, while maintaining the computational

power of simple RNNs, can make training easier over data exhibiting long-term dependencies [5]. Instances of such variants are LSTM [14] and GRU [7], which thanks to the introduction of so-called gating mechanisms allow the network to remember or discard from the internal state selected information about the input sequences (or, from the point of view of the gradient computation, can alleviate the problem of gradient vanishing [5]). While definitely effective in terms of predictive accuracy, gated network models still require gradient descent for training, which is often much more computationally expensive than the reservoir computing approaches.

In recent years, there has been an increasing research interest regarding alternative solutions for maintaining information over long time spans in recurrent models. An example is the application of the Learning-to-Learn paradigm and neuronal adaptation to spiking neural networks in the context of reservoir computing methodologies [29,2]. In this paper we focus on the following question: *can typical gating mechanisms be embedded within efficient reservoir computing networks, and within what degree of effectiveness?*

In ESNs, whose recurrent dynamics are completely untrained, it is not immediate to extend the architecture with gate-like mechanisms. In this work we investigate whether it is possible to extend the architecture of an ESN by introducing gating mechanisms, and we analyze the results in terms of training efficiency and predictive performance. The goal is to make progress towards efficient neural models which improve the predictive performance with respect to the current reservoir computing state-of-the-art, while maintaining most of the efficiency.

In short, we summarize here the key contributions of this work which are:

- the extension of the ESN architecture with the use of gating mechanisms, which we denote as *Gated ESN*;
- the theoretical analysis of the conditions associated to the fading memory of the network dynamics of the Gated ESN;
- a critical experimental analysis of the Gated ESN; and
- the suggestion of likely paths towards effective gated reservoir computing models.

Regarding the organization of the manuscript, it is laid out as follows: we start by discussing a few related works in Section 2 before introducing, in Section 3, the two main approaches from the state-of-the-art literature regarding RNN training, namely ESNs and GRUs. In Section 4 we describe the model that we study (Gated ESN), and we provide a theoretical analysis for the conditions related to memory and stability in Section 5. Then, in Section 6 we report the methodology and the results regarding our experimental analysis, whose implications are discussed in Section 7. Finally, in Section 8 we draw the conclusions.

2. Related Works

The process of extending the architecture of ESNs with gating mechanisms has been first investigated in our previous work [9] and, concurrently and independently, in [32]. The two works share a similar idea, which consists of extending the state transition function of an ESN to include the same gating mechanisms of a GRU, while keeping all the parameters in the gates untrained. Both works investigate the behavior of such network

when trained by conventional reservoir computing techniques (as opposed to the typical approach of training GRUs by using gradient descent).

While the underlying idea from [9] and [32] is similar, in [9] and in the current work we also take care to (1) consider the fundamental details regarding the initialization strategy, (2) perform an experimental evaluation over a dataset that makes it possible to evaluate more clearly the actual influence of the gates, and (3) perform a more extensive evaluation of the competing reservoir computing models, which led to important insights about the feasibility of the approach. Moreover, the current paper further extends our previous work [9] to include (a) the development of a theoretical analysis of the state dynamics in the proposed models, (b) the analysis and discussion of the agreement between the theoretical results and the experimental measurements, (c) the expansion of the hyperparameter search for the experiments, (d) the collection and discussion of additional measurements for the activation of the gates, and (e) the reporting and discussion of additional measurements regarding the weight matrices.

Finally, in the *Gating ESN* model [1] from earlier literature, it is employed a combination of many parallel instances of ESNs, each initialized with different hyperparameters, and each trained separately on the same task. A gating network then learns to select which instance to use for any given time-step. While the name of our proposed model may bear similarity with the *Gating ESN*, the approaches are radically different: instead of instantiating many different sub-models, we aim to enrich the dynamics of the state of a single ESN by explicitly introducing gated units (cells) in the spirit of architectures such as GRU [7] and LSTM [14].

3. Background

In this section we briefly describe the models that serve as the basis of our study: in Section 3.1 we describe the Echo State Networks (ESNs), a reservoir computing approach for modeling sequences, while in Section 3.2 we describe the popular approach of Gated Recurrent Units (GRUs), representing one of several existing variants of gated RNNs.

3.1. Echo State Networks

Among the different instances of the general framework of RNNs, ESNs [16,17] represent an efficient approach for sequence modeling thanks to the use of a distinctive perspective for the study of the recurrent network. As reservoir computing models, in ESNs there is a sharp distinction between the recurrent part of the network, which is referred to as *reservoir*, and the output layer, which is called *readout*. The *reservoir* is studied as a dynamical system and is responsible for embedding the input sequence into a high dimensional state space of fixed size. The key characteristic is that the reservoir does not get trained. Instead, the weights in the reservoir are initialized from a random distribution that allows to meet a mathematical property for stability. We will discuss this property in the final part of this section. The *readout* is typically implemented as a linear regression model. Since it is the only part of the model that gets trained, it is possible to obtain closed-form solution to the regression problem.

The architecture of an ESN has three main dimensions: the number of input units (N_U), the number of units in the reservoir (N_R) and the number of output units in the

readout (N_Y). Then, the ESN can be applied to sequences $\mathbf{u}(1), \ldots, \mathbf{u}(T) \in \mathbb{R}^{N_U}$ of any length T.

The state of the reservoir of the network at each time step t, which is denoted as $\mathbf{x}(t) \in \mathbb{R}^{N_R}$, is computed as

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{0}, \\ \mathbf{x}(t) &= \tanh\left(\mathbf{W}_{in}\mathbf{u}(t) + \hat{\mathbf{W}}\mathbf{x}(t-1)\right), \end{aligned} \tag{1}$$

where $\mathbf{W}_{in} \in \mathbb{R}^{N_R \times N_U}$ is the input-to-reservoir weight matrix, and $\hat{\mathbf{W}} \in \mathbb{R}^{N_R \times N_R}$ is the recurrent reservoir-to-reservoir weight matrix.

Instead of being tuned by a training process, the matrices $\mathbf{W}_{in}$ and $\hat{\mathbf{W}}$ are fixed after being randomly initialized. As part of the random initialization process, $\mathbf{W}_{in}$ is multiplied by a real scaling hyperparameter. The matrix $\hat{\mathbf{W}}$, instead, is initialized so that its norm $\|\hat{\mathbf{W}}\|$ or spectral radius $\rho(\hat{\mathbf{W}})$ (the largest eigenvalue in absolute value) meets the conditions for stability given in [16]. We report a sufficient condition for stability at the end of this section.

After the states for the input sequence have been collected, the output is computed as

$$\mathbf{y}(t) = \mathbf{W}_{out}\mathbf{x}(t), \tag{2}$$

with $\mathbf{W}_{out} \in \mathbb{R}^{N_Y \times N_R}$.

Leaky ESN – A simple but effective variant of the basic ESN is denoted as *leaky ESN*. In the leaky ESN, the neurons in the reservoir are leaky-integrators [18] which act as lowpass filters. Their leaking rate is considered a hyperparameter of the model, and as such is chosen and fixed at model selection time. For a leaky ESN, the state transition function of the reservoir is defined as

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{0}, \\ \mathbf{x}(t) &= (1-a)\,\mathbf{x}(t-1) + a\tanh\left(\mathbf{W}_{in}\mathbf{u}(t) + \hat{\mathbf{W}}\mathbf{x}(t-1)\right), \end{aligned} \tag{3}$$

where $a \in \mathbb{R}$ is the leaking rate, under the constraint that $0 < a \leq 1$.

Training – The characteristic of ESNs and its variants such as leaky ESNs is that the weights in the reservoir are not trained. As such, the only parameters that need to be optimized are those in the readout, i.e., the matrix $\mathbf{W}_{out}$. Since the readout is limited to a linear computation, a closed-form solution to the convex minimization of the error can be obtained by algorithms such as ridge regression. In practice, first the input data is fed to the reservoir and the N_{train} states that need to be classified are collected column-wise into a matrix $\mathbf{X} \in \mathbb{R}^{N_R \times N_{train}}$. Then, the readout is trained by finding a solution to the following least squares problem:

$$\min_{\mathbf{W}_{out}} \|\mathbf{W}_{out}\mathbf{X} - \mathbf{Y}_{tg}\|_2^2. \tag{4}$$

In Equation 4, $\mathbf{Y}_{tg} \in \mathbb{R}^{N_Y \times N_{train}}$ indicates the column-wise concatenation of the target vectors. A regularized solution to Equation 4 can be computed in closed-form as follows:

$$\mathbf{W}_{out} = \mathbf{Y}_{tg}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda_r\mathbf{I})^{-1}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix, and $\lambda_r \in \mathbb{R}^+$ is the regularization parameter which can be chosen by model selection.

Echo State Property – To guarantee the stability conditions that allow the untrained state dynamics to encode meaningful representations of the data, the reservoir must meet the so-called Echo State Property (ESP) [16,33]. The ESP is related to asymptotic properties of the reservoir and intuitively states that, for sufficiently long input sequences, the state in which the network ends up should only depend upon the input itself. In other words, the initial conditions of the network should not influence its long-term dynamics. For ESNs with hyperbolic tangent activation functions, a sufficient condition for the ESP is $\|\hat{\mathbf{W}}\|_2 < 1$ (see [16]).

3.2. Gated Recurrent Units

The problems associated to gradient descent training with backpropagation through time over long input sequences [5] led to the development of gated RNN models such as LSTM [14] and GRU [7]. The gates are simple mechanisms that are able to dynamically squash to zero the individual components of the possibly multidimensional signal to which they are applied. In the case of GRU, the state transition function contains two gates which are called *reset gate* and *update gate*. The activations of such gates at time t are denoted as respectively $\mathbf{r}(t) \in \mathbb{R}^{N_R}$ and $\mathbf{z}(t) \in \mathbb{R}^{N_R}$. Informally, a gate is said to *open* or *close* depending on the values of its activations, which vary in $(0, 1)$.

Intuitively, the purpose of the gates in the GRU is to open and close to regulate the flow of information within the state: the reset gate can zero out unnecessary information from the previous state $\mathbf{x}(t-1)$, while the update gate can merge information from the previous state and the current candidate state $\mathbf{h}(t)$ into the new state $\mathbf{x}(t)$. More in detail, the recurrent state $\mathbf{x}(t)$ of a GRU at a given time step t is computed as:

$$
\begin{aligned}
&\mathbf{x}(0) = \mathbf{0}, \\
&\mathbf{r}(t) = \sigma(\mathbf{W}_{in}^{r}\mathbf{u}(t) + \hat{\mathbf{W}}^{r}\mathbf{x}(t-1)) \\
&\mathbf{z}(t) = \sigma(\mathbf{W}_{in}^{z}\mathbf{u}(t) + \hat{\mathbf{W}}^{z}\mathbf{x}(t-1)) \\
&\mathbf{h}(t) = \tanh(\mathbf{W}_{in}\mathbf{u}(t) + \hat{\mathbf{W}}(\mathbf{r}(t) \odot \mathbf{x}(t-1))) \\
&\mathbf{x}(t) = \mathbf{z}(t) \odot \mathbf{x}(t-1) \; + \; (1 - \mathbf{z}(t)) \odot \mathbf{h}(t).
\end{aligned}
\tag{6}
$$

Here, we have $\mathbf{r}(t), \mathbf{z}(t), \mathbf{h}(t), \mathbf{x}(t) \in \mathbb{R}^{N_R}$, and in particular $\mathbf{r}(t), \mathbf{z}(t) \in (0, 1)$ due to the sigmoidal activation function.

From a given state of the GRU, a prediction can be obtained by means of any kind of differentiable readout layer, such as a linear one. The whole model (including the behavior of the gates) can be trained end-to-end by backpropagation through time.

4. Gated ESN

The capability of ESNs to discriminate between different input sequences depends on the guarantees given by the ESP [16] described in Section 3.1. The ESP is related to the *fading-memory* property of the ESN, i.e., in a properly initialized ESN any information from the initial conditions of the reservoir will be asymptotically washed out with time. It is easy to see how the fading memory property, if not properly controlled, can

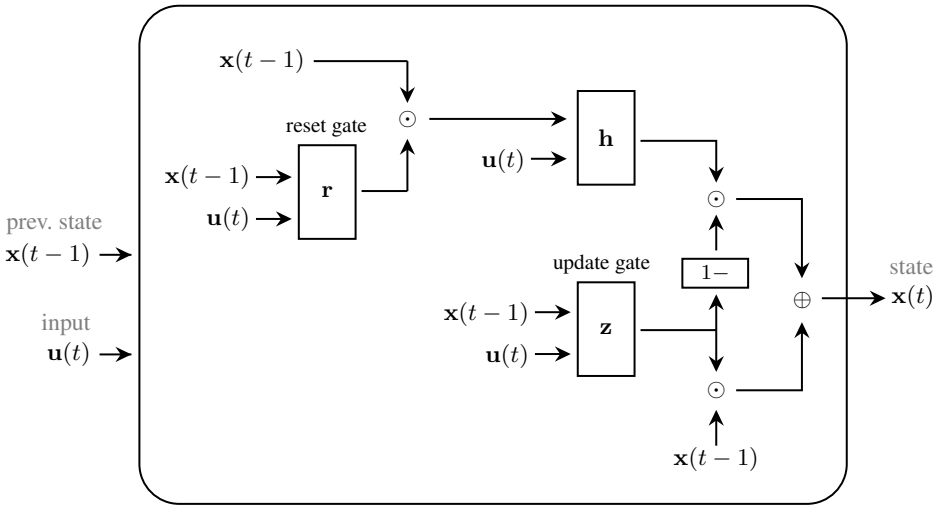


Fig. 1. Depiction of the recurrent cell of *Gated ESN* (and *Gated ESN RZ*) for a generic time step t. Symbols $\odot$ and $\oplus$ respectively denote the elementwise product and the sum of two vectors. While the architecture is identical to the one of a GRU, in *Gated ESN* the parameters controlling the activations of $\mathbf{r}(t), \mathbf{z}(t)$, and $\mathbf{h}(t)$ are not trained (hence only the readout is trained, which is not depicted here). For *Gated ESN RZ*, instead, in addition to the readout also the parameters controlling $\mathbf{r}(t)$ and $\mathbf{z}(t)$ are trained while the dynamics of $\mathbf{h}(t)$ remain untrained

also represent a dramatic limitation by introducing a Markovian bias [11] in the model. For example, imagine a task characterized by long input sequences in which the key information for performing correct predictions is often located near the beginning of the sequences. In such cases, the Markovian bias may prevent the readout to access such information, thus reducing the learning capability of the model. One may argue that it is always possible to optimize the amount of memory of the model (e.g., by increasing the number of recurrent units) so that distant information will not be lost, however this strategy does not allow to generalize to different sequence lengths. What is needed is a way for ESNs to dynamically and selectively remember or forget parts of a sequence while preserving their generalization capabilities.

We investigate whether it is possible to employ gating mechanisms in order to improve the predictive performance of ESNs in such contexts. To this aim, we make use of a gated architecture paired with a reservoir computing training methodology. In particular, we define two models which only differ for their training method. In fact, both models borrow the same state transition function of the GRU (Equation 6) as illustrated in Fig. 1. However, in the first model the recurrent part is fully untrained, while in the second one we make partial use of training for the gates. The whole model is illustrated in Fig. 2 by showing its unfolding in time. The details of the recurrent cells for Gated ESN and Gated ESN RZ are described in Sections 4.1 and 4.2.

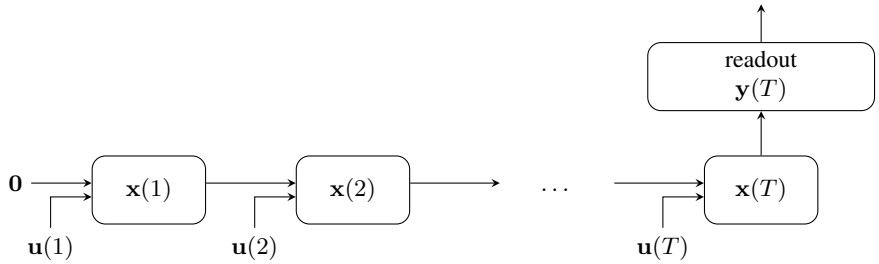


Fig. 2. Unfolding in time of the Gated ESN and Gated ESN RZ architecture for a sequence of length T. Each state $\mathbf{x}(t)$ is computed by the recurrent cell illustrated in Fig. 1. Since in this work we are concerned with the classification of a sequence given the last time step, we only show that configuration.

Table 1. Schematic view of the different matrices involved in Gated ESN, Gated ESN RZ and GRU. On the right we report whether a given matrix is tuned as part of the training procedure

Matrix	Shape	Description	Trained?		
			Gated ESN	Gated ESN RZ	GRU
$\mathbf{W}_{in}^{r}$	$N_R \times N_U$	input to reset gate	–	✓	✓
$\hat{\mathbf{W}}^{r}$	$N_R \times N_R$	reservoir to reset gate	–	✓	✓
$\mathbf{W}_{in}^{z}$	$N_R \times N_U$	input to update gate	–	✓	✓
$\hat{\mathbf{W}}^{z}$	$N_R \times N_R$	reservoir to update gate	–	✓	✓
$\mathbf{W}_{in}$	$N_R \times N_U$	input to reservoir	–	–	✓
$\hat{\mathbf{W}}$	$N_R \times N_R$	reservoir to reservoir	–	–	✓
$\mathbf{W}_{out}$	$N_Y \times N_R$	readout	✓	✓	✓

4.1. Gated ESN

We denote the first variant as *Gated ESN*. This can be considered a pure reservoir computing model in the sense that its reservoir is fully untrained and its inner mechanics (which resemble those of a GRU) are completely irrelevant for the training algorithm of the readout. In particular, all matrices in the reservoir (including those in the gates) are randomly initialized and then rescaled according to the value of corresponding hyperparameters, just like what happens in a standard ESN. The only parameters that get trained are those in the linear readout, as highlighted in Table 1, with no difference with respect to what happens in a standard ESN. In fact, the parameters of the readout can be obtained by ridge regression.

4.2. Gated ESN RZ

We denote the second variant as *Gated ESN RZ*. The name comes from the fact that in this case, the behavior of the reset and update gates (*R* and *Z*) is learned from the data. Due to the multiple nonlinearities separating the output of the model with the parameters of the gates, the training algorithm of choice is gradient descent with backpropagation through time for both the parameters of the gates ($\mathbf{W}_{in}^{r}$, $\mathbf{W}_{in}^{z}$, $\hat{\mathbf{W}}^{r}$, and $\hat{\mathbf{W}}^{z}$) and, jointly,

those of the readout ($\mathbf{W}_{out}$). The other matrices of the reservoir, namely $\mathbf{W}_{in}$ and $\hat{\mathbf{W}}$, are randomly initialized, rescaled and then kept fixed as in *Gated ESN*. A summary of which are the matrices that get trained is available in Table 1.

5. Contractivity conditions of the gated reservoir

In this section we provide an analysis of the state dynamics for a gated reservoir. In particular, we seek conditions to escape from the inherent fading-memory behavior of conventional reservoir computing approaches. To this end, we derive a bound that the reservoir matrices must satisfy when the state transition function is non-contractive. The results are insightful for the initialization and the analysis of such systems, as they enable us to characterise the contractivity (and the resulting fading-memory regime) of such gated architectures. We focus our analysis over the architecture of GRU, regardless of whether it is trained (as, precisely, in GRU), untrained (as in Gated ESN) or partially trained (as in Gated ESN RZ). Thus, the results apply to all these models after initialization and, where applicable, training.

For ease of notation, in this section let us hide the explicit time dependency and denote with $\mathbf{u}$ and $\mathbf{x}$ respectively a generic input vector and state vector. The state transition function of the reservoir will be represented by the function $\tau : \mathbb{R}^{N_U} \times \mathbb{R}^{N_R} \to \mathbb{R}^{N_R}$. Moreover, we denote with $\left(\frac{\partial y(x)}{\partial x}\right)_{f(x)}$ the partial derivative of y with respect to x while considering $f(x)$ as a constant.

For the ESP to hold, it can be shown that it is sufficient for the state transition function to be contractive. Contractivity is defined as:

$$\begin{gathered}\exists C \in \mathbb{R}, \quad 0 \leq C < 1, \quad \forall \mathbf{u} \in \mathbb{R}^{N_U}, \quad \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^{N_R} : \\ \left\|\tau(\mathbf{u}, \mathbf{x}) - \tau(\mathbf{u}, \mathbf{x}')\right\| \leq C \left\|\mathbf{x} - \mathbf{x}'\right\|,\end{gathered} \tag{7}$$

i.e. τ must be Lipschitz continuous with constant $C < 1$. Equation 7 can also be formulated in terms of the derivative of τ, in fact:

$$\begin{gathered}\forall C \in \mathbb{R}, \quad C \geq 0, \quad \forall \mathbf{u} \in \mathbb{R}^{N_U}, \quad \forall \mathbf{x} \in \mathbb{R}^{N_R} : \\ \tau \text{ is } C\text{-Lipschitz w.r.t. } \mathbf{x} \quad \iff \quad \left\| \frac{\partial \tau(\mathbf{u}, \mathbf{x})}{\partial \mathbf{x}} \right\| \leq C.\end{gathered} \tag{8}$$

We now study the characterization of the asymptotic stability of the state dynamics for a GRU. First, in Lemma 1 we show that for a GRU whose weights are within a given bound, the contractivity of the state transition function is guaranteed. Then we use the aforementioned bound to derive the main result of Proposition 1, which gives a necessary condition for a state transition function that is non-contractive.

Lemma 1. *Let τ be the state transition function of a GRU as defined in Equation 6, and let $z_{max} = \max_t \left\|\mathbf{z}(t)\right\|_\infty$. A sufficient condition for the contractivity of τ is:*

$$\|\hat{\mathbf{W}}\|_2 \left(1 + \|\hat{\mathbf{W}}^r\|_2\right) + 2\|\hat{\mathbf{W}}^z\|_2 + z_{max} < 1. \tag{9}$$

Proof. We write the state transition function $\tau(\mathbf{u}, \mathbf{x})$ as

$$\begin{aligned} \mathbf{r} &= \sigma(\mathbf{W}_{in}^{r}\mathbf{u} + \hat{\mathbf{W}}^{r}\mathbf{x}) \\ \mathbf{z} &= \sigma(\mathbf{W}_{in}^{z}\mathbf{u} + \hat{\mathbf{W}}^{z}\mathbf{x}) \\ \mathbf{h} &= \tanh(\mathbf{W}_{in}\mathbf{u} + \hat{\mathbf{W}}(\mathbf{r} \odot \mathbf{x})) \\ \tau(\mathbf{u}, \mathbf{x}) &= \mathbf{z} \odot \mathbf{x} \ + \ (1 - \mathbf{z}) \odot \mathbf{h}, \end{aligned} \tag{10}$$

Then we can compute the 2-norm of the partial derivative of τ with respect to x as follows:

$$\begin{aligned} \left\| \frac{\partial \tau(\mathbf{u}, \mathbf{x})}{\partial \mathbf{x}} \right\|_2 &= \left\| \frac{\partial \tau(\mathbf{u}, \mathbf{x})}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} + \frac{\partial \tau(\mathbf{u}, \mathbf{x})}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{x}} + \left(\frac{\partial \tau(\mathbf{u}, \mathbf{x})}{\partial \mathbf{x}} \right)_{\substack{\mathbf{h} \\ \mathbf{z}}} \right\|_2 \\ &= \Big\| \operatorname{diag}(1 - \mathbf{z}) \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \\ &\quad + \operatorname{diag}\big((\mathbf{x} - \mathbf{h}) \odot \mathbf{z} \odot (1 - \mathbf{z})\big) \hat{\mathbf{W}}^{z} \\ &\quad + \operatorname{diag}(\mathbf{z}) \Big\|_2 \\ &\leq \|1 - \mathbf{z}\|_\infty \left\| \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right\|_2 \\ &\quad + \|(\mathbf{x} - \mathbf{h}) \odot \mathbf{z} \odot (1 - \mathbf{z})\|_\infty \|\hat{\mathbf{W}}^{z}\|_2 \\ &\quad + \|\mathbf{z}\|_\infty \\ &\leq \|1 - \mathbf{z}\|_\infty \|\hat{\mathbf{W}}\|_2 \left(1 + \|\hat{\mathbf{W}}^{r}\|_2\right) \\ &\quad + \|(\mathbf{x} - \mathbf{h}) \odot \mathbf{z} \odot (1 - \mathbf{z})\|_\infty \|\hat{\mathbf{W}}^{z}\|_2 \\ &\quad + \|\mathbf{z}\|_\infty \\ &\leq \|\hat{\mathbf{W}}\|_2 \left(1 + \|\hat{\mathbf{W}}^{r}\|_2\right) + 2\|\hat{\mathbf{W}}^{z}\|_2 + \|\mathbf{z}\|_\infty \\ &\leq \|\hat{\mathbf{W}}\|_2 \left(1 + \|\hat{\mathbf{W}}^{r}\|_2\right) + 2\|\hat{\mathbf{W}}^{z}\|_2 + z_{max}. \end{aligned} \tag{11}$$

where $\operatorname{diag}(\mathbf{v})$ (for a generic vector $\mathbf{v}$) is the diagonal matrix with the entries of $\mathbf{v}$ on the main diagonal, and $\|\mathbf{v}\|_\infty = \max_i |v_i|$ is the infinity norm of vector $\mathbf{v}$.

Equation 11 provides an upper bound to the derivative of τ. From Equation 8 it follows that when this bound is less than unity, τ is contractive. □

While the contractivity of the state transition function ensures that the ESP is satisfied, the whole idea of a gated architecture is for the state dynamics to not be restricted to contractive trajectories. This would allow the network to relax the strong Markovian bias discussed at the beginning of Section 4 and escape from the strict fading memory behaviour. Then, the initialization strategy must ensure that the network is outside of a strictly contracting regime. From the bound of Lemma 1 we can obtain a necessary condition for having a non-contractive state transition function, as reported in Proposition 1.

Proposition 1. *Let τ be the state transition function of a GRU as defined in Equation 6, and let $z_{max} = \max_t \|\mathbf{z}(t)\|_\infty$. If τ is non-contractive, then it holds:*

$$\|\hat{\mathbf{W}}\|_2 \left(1 + \|\hat{\mathbf{W}}^{r}\|_2\right) + 2\|\hat{\mathbf{W}}^{z}\|_2 + z_{max} > 1. \tag{12}$$

Proof. The statement follows straightforwardly from the result in Lemma 1 (by negation). □

In other words, the result in Proposition 1 states that if the GRU is outside of the fading memory regime, then Equation 12 must be satisfied. We can use this bound as a means to verify the contractivity conditions of the different models under consideration, and possibly as a strategy for the initialization of the weights. Note that the presence of the term $z_{max} \in (0, 1)$ in Equation 12 suggests that a network on the edge of a strictly contractive regime could be able to dynamically enter and exit such regime by means of the activations of the update gate.

6. Experimental analysis

In this section we describe in detail the experimental evaluation of the gated models introduced in Sections 4.1 and 4.2. In particular, we test our hypothesis (i.e., can gates provide advantages to reservoir computing models?) on a Natural Language Processing task which has been specifically chosen for the presence of long-term dependencies, and thus for its potential to clearly highlight the effect of the gating mechanisms. Note that the application of ESNs to Natural Language Processing tasks, whose data by their nature can often include long-term dependencies, has been quite limited: to the best of our knowledge there are only a few of such works [26,25,28,10].

6.1. TREC Dataset

We have chosen to empirically assess our model over real-world data exhibiting clear long-term dependencies. A good fit for a dataset exhibiting these characteristics is the TREC dataset for the Question Classification task[1] [19], which is a commonly used benchmark for evaluating Natural Language Processing systems. The TREC dataset deals with the task of classifying a number of input sentences, written in English, into one of 6 classes that indicate their broad topic (i.e. whether they ask about a person, a location, a number, a human being, a description or an entity). The output classes are represented in our models as one-hot encoded vectors. While the dataset also contains more detailed fine-grained classes, here we only focus on the 6 commonly used coarse-grained classes.

To support our model validation methodology, the dataset has been split in three folds: training, validation and test. The test fold is directly provided by the authors of the dataset [19] and contains 500 labeled questions. The other fold provided by the authors of the dataset, composed of 5452 labeled questions, was partially used for training and partially for validation. In fact, we have split this fold by the commonly used "80/20 rule", where 80% of the instances (chosen at random) are used for training and the other 20% for validation. This yields a training set of 4362 questions and a validation set of 1090 questions, with similar class distributions between the two sets (we did not perform an explicit stratification).

We have performed tokenization of the input questions, so that we could assign a word embedding to each token. In particular, we represented each token by a pretrained FastText embedding vector for the English language, with 300 dimensions [13]. Whenever

[1] `http://cogcomp.org/Data/QA/QC/`

Table 2. The total number of trainable parameters across the models is kept constant by controlling the size N_R of the state

	ESN	Gated ESN	Leaky ESN	Gated ESN RZ	GRU
Trainable params	19386	19386	19386	19386	19386
N_R	3230	3230	3230	29	20

a word that does not have a corresponding embedding in FastText is encountered, we use a random vector of the same shape instead. This vector is different for each missing word.

6.2. Experimental methodology

All models have been selected after a randomized hyperparameter search of 60 iterations by employing a hold-out validation set. Then, the selected models are retrained over the union of the training and validation sets, and their performance on the test set is measured and averaged across 10 trials, each with different random initializations of the parameters. Where needed (i.e., when employing gradient descent), the data in the training set has been shuffled.

To provide a fair and rigorous comparison, we made sure to keep the total number of trainable parameters uniform between all models by controlling the number of recurrent units, as shown in Table 2. In the literature this is a commonly used strategy for comparing RNNs, since forcing the same number of units for all models would lead to misleading results [8].

We point out that several architectural modifications can be introduced to significantly boost the predictive performance of an ESN on this task, as shown in our previous work [10]. For example, a bidirectional architecture [6,27] can easily help to capture most of the important information in this task [10]. However, in this work we deliberately consider only the simplest architectures in order to focus on the improvements brought by the gates.

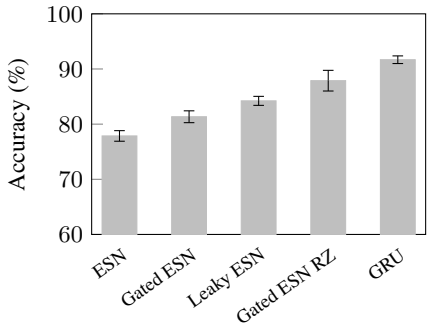
All experiments have been carried out on a NVIDIA Tesla V100 GPU.

Initialization – All reservoir matrices for the ESN, Gated ESN and Leaky ESN have been randomly initialized by sampling from $\mathcal{U}(-1, 1)$ and then rescaled according to the hyperparameters. The same initialization scheme has been used for Gated ESN RZ; here and in GRU, however, the entries for the matrices which are tuned by gradient descent are directly sampled from $\mathcal{U}\left(-1/N_R, 1/N_R\right)$ without further rescaling.

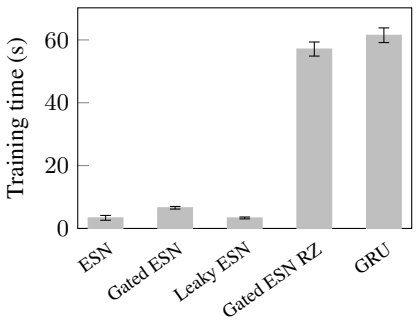
Training algorithm – The pure reservoir computing models (i.e., ESN, Gated ESN, and Leaky ESN) only involve the training of a linear output layer. For this reason, in such cases we employ ridge regression. The Gated ESN RZ and the GRU are trained by gradient descent. In Gated ESN RZ, we avoid the computation of the gradients associated to the matrices that are kept fixed ($\mathbf{W}_{in}$ and $\hat{\mathbf{W}}$).

6.3. Results

In Fig. 3 we have reported the predictive accuracy and training time for the investigated variants of ESNs and for a fully trained GRU. Here we start our analysis with some



(a) Predictive performance. The highest accuracy is reached by the fully trained model, while the basic ESN is the baseline over which the improvements are built



(b) Training time. As soon as backpropagation is used (last two bars), even if only partially, the very fast training times of the pure reservoir computing models grow significantly.

Fig. 3. Test set results for the experimental comparison on the Question Classification task, with standard deviations. *Left:* Test set accuracy *Right:* Training times. The gates in Gated ESN are able to improve the predictive performance of the ESN, but training their parameters (Gated ESN RZ) seems necessary. Unfortunately, using backpropagation for this training process drastically increases the training time.

general considerations, leaving to a later moment more specific considerations on the results obtained by architectures with untrained or trained gates.The first three models in Fig. 3a have a completely untrained reservoir and as such they all exhibit the same number of recurrent units *and* trainable parameters. Nonetheless, a distinct difference in predictive accuracy can be observed between these models, which hints at the importance of more advanced reservoir state transition functions. The importance of reservoir computing models is clearly highlighted when comparing the time required for training the parameters (see Fig. 3b): the pure reservoir computing models (the first three, i.e., ESN, Gated ESN, and Leaky ESN) provide a remarkable efficiency advantage with respect to the other two (Gated ESN RZ and GRU).

As expected, the best performing model is the GRU, in which all the parameters are trained by gradient descent and backpropagation through time. Also as expected, the basic ESN displays the lowest level of accuracy. While the number of trainable parameters is the same in these two models, their striking difference in accuracy is due to their different biases with respect to the data. In the task under consideration, the most important input words for producing a correct prediction are often located at the beginning of the sentences: due to the fading memory property of the ESN, their contribution has a very low influence on the final states of the network, which are those observed by the classifier [10].

Untrained gates dynamics – For what concerns the improvements in predictive performance brought by the Gated ESN architecture with untrained gates, it can be observed from Fig. 3 that there is a significant increase in accuracy with respect to an ESN. However, the simpler model which uses leaky-integrator neurons (Leaky ESN) produces better results. To better understand this trend, we take a step further in our analysis and in Ta-

Table 3. Statistics about the activations of the gates for Gated ESN. Mean, standard deviation and maximum value are computed by aggregating across both the unit dimension and the time dimension

$\mathbf{r}(t)$			$\mathbf{z}(t)$		
mean	std. dev	max	mean	std. dev	max
0.4997	0.0761	0.9993	0.5000	0.0108	0.6488

Table 4. Average spectral radius and norm of the reservoir matrices after initialization and training (where applicable). For different models, the matrices can have different size (see Table 2). Also note that the value of $a = 0.04$ has been chosen by model selection in $\mathcal{U}(0, 1)$

	State		Reset gate		Update gate	
Model	$\rho(\hat{\mathbf{W}})$	$\|\hat{\mathbf{W}}\|$	$\rho(\hat{\mathbf{W}}^r)$	$\|\hat{\mathbf{W}}^r\|$	$\rho(\hat{\mathbf{W}}^z)$	$\|\hat{\mathbf{W}}^z\|$
ESN	1.35	76.38	–	–	–	–
Gated ESN	5.30	297.92	0.28	15.70	0.06	3.37
Leaky ESN ($a = 0.04$)	0.02	1.16	–	–	–	–
Gated ESN RZ	7.62	9.18	8.16	14.57	7.59	25.31
GRU	4.52	10.06	1.69	7.86	2.78	11.91

ble 3 we report the main statistics about the activations of the gates over time. According to the measurements in Table 3 the gates (and especially the *update* gate) are not being fully exploited. In fact, from Table 3 it can be inferred that the matrices in the update gate have been rescaled by the procedure of model selection so that it behaves roughly like a constant, in this case $\mathbf{z}(t) \approx 0.5 \cdot \mathbf{1}\ \forall t$. This can be deduced by observing the low standard deviations for the gate activations. Thus, the behavior of the Gated ESN is actually approximating on average the one of a Leaky ESN. In the upper part of Table 4 we have reported the average norm of the recurrent matrices for the different models. It can be observed that Gated ESN tends to values of $\|\hat{\mathbf{W}}\|$ that are significantly higher than the other untrained models (ESN and Leaky ESN). This by itself brings Gated ESN well within the bound provided by Proposition 1, which gives a necessary condition for the non-contractivity of the state transition function and thus allows state dynamics that are outside of the fading memory regime which is typical of ESNs. The values of $\|\hat{\mathbf{W}}^r\|$ and $\|\hat{\mathbf{W}}^z\|$ are made irrelevant for the bound by the higher-than-unity value of $\|\hat{\mathbf{W}}\|$.

For completeness, in Table 4 we also report the average spectral radius of the matrices, as it represents a reference parameter for the initialization of the recurrent matrices in reservoir computing-based networks. Most networks (all except the Leaky ESN) exhibit a value of $\rho(\hat{\mathbf{W}}) > 1$, which is noteworthy because for basic ESNs, $\rho(\hat{\mathbf{W}}) < 1$ represents a traditionally used bound for the initialization of ESNs in practical applications (even though it is not a sufficient condition for the ESP). This hints to the fact that the networks are trying to escape from the fading memory regime implied by their contractive state transition function, but in the case of the non-gated models it is impossible to dynamically do so.

Trained gates dynamics – While using gates with untrained parameters appears to be effective only to a limited extent, applying learning as in *Gated ESN RZ* drastically im-

proves the predictive performance (Fig. 3a). In particular, even though the training is only applied to the parameters in the gates and the dynamics are still mainly determined by random weights, the accuracy of *Gated ESN RZ* dominates over all pure reservoir computing models. Moreover, in this case a relatively small size of the reservoir is sufficient to obtain good performance (29 units instead of 3230, as indicated in Table 2). However, from the measurements reported in Fig. 3b it is clear that the introduction of backpropagation through time also causes a severe increase in the training time. This makes the approach of training the gates via the specific algorithm of backpropagation through time unappealing in practice, as the efficiency advantages coming from the reservoir computing approach are vanishing.

Regarding the average matrix norms reported in the bottom part of Table 4, notice how even though matrix $\hat{\mathbf{W}}$ for the GRU is trained, it ends up having a very similar norm to the one of Gated ESN RZ, in which $\hat{\mathbf{W}}$ is untrained (10.06 for GRU versus 9.18 for Gated ESN RZ). In addition, we can observe how both models respect the bound from Proposition 1, i.e. the necessary condition for the non-contractivity of the state transition function. This does not mean that the state transition function is never contractive. On the contrary, it is likely that the recurrent dynamics are mainly contractive, which is needed in order to provide meaningful data representations to the readout. However, in the case of Gated ESN RZ, the untrained dynamics of the reservoir are able to occasionally exit the contractive (i.e., fading memory) regime thanks to the activations of the gates, thus allowing the network to relax its Markovian bias and to increase its memory capacity. The same happens in GRU, even though by means of a fully adapted state transition function.

7. Discussion

We have shown how there exist tasks with particular characteristics such that simple ESNs, despite their exceptional efficiency, do not compete in accuracy with the more popular and expensive alternatives such as GRU. In such cases, a state transition function that is able to give different weights to different parts of the input can have an important impact on the predictive performance of the model.

Equipping the reservoir of the ESN with gating mechanisms while maintaining its weights untrained does not appear to be sufficient for a meaningful increase in predictive performance. What seems to be effective, instead, is maintaining a mostly untrained dynamics but injecting a training signal into the gates. The benefits of such approach are twofold. On one hand, training only the gates allows to employ much smaller reservoirs than what would be necessary in an ESN, and a reservoir of a given size can also generalize better with respect to longer sequences.[2] On the other hand, the approach has the potential of reducing the training time compared to what would be required for a GRU.

Currently, due to its relatively low efficiency the algorithm of backpropagation through time is not suited to train the parameters of the gates. However, the constrained model that we propose under the name of Gated ESN RZ has the potential for allowing the use of less conventional training algorithms that may be more efficient in this case. One of the problems with the use of backpropagation through time for Gated ESN RZ is that, for all time steps, the gradients need to flow through $\mathbf{h}(t)$ and $\mathbf{x}(t)$ anyway, regardless of the

[2] For example, in principle the network is allowed to discard any irrelevant time step in the input sequence without alterations of the reservoir state.

fact that the parameters $\mathbf{W}_{in}$ and $\hat{\mathbf{W}}$ are *not* trained. This adds a significant cost to the computation of the gradients, especially on larger reservoirs.

Considering the potential impact of gated reservoir computing, innovative methods for training the gates are needed. One likely candidate is represented by the class of local algorithms which could *bypass* the chain of gradients by injecting a training signal directly into the gates, a concept similar to the direct feedback alignment in feedforward networks [24]. An instance of such algorithms is the biologically inspired Hebbian learning, which must be modulated by an error signal to allow supervised learning. However, replacing backpropagation through time requires to employ alternative techniques for addressing the problem of *credit assignment*, or *distal reward* [23]: how did each individual synapse contribute to the final prediction of the model, especially in case of long delays before the error signal is available? The impact of this problem is clearly evident in the case of classification problems, in which the error signal is only available at the end of each sequence. For addressing the credit assignment problem, variants of biologically motivated mechanisms of *eligibility traces* [15] can be used, which can also be compatible with approximations of the gradients that would be computed by backpropagation through time [4,3].

In the literature, reward-modulated Hebbian learning has already been successfully employed for training all parameters of a recurrent neural network [22]. However, in order to exploit the untrained discrimination capabilities of the reservoir computing approach we see as a promising method that of not training the whole network, but instead only steering the trajectories of the state of the reservoir through the use of gate or gate-like mechanisms trained by variants of the above-mentioned approach.

Efficient alternatives for training RNNs are needed: the suggested approach of employing both untrained and trained dynamics could represent a tool for allowing the already efficient ESNs to effectively tackle problems which today represent a hurdle due to the presence of long-term dependencies.

8. Conclusion

In this paper we have discussed the introduction of gated mechanisms in the architecture of reservoir computing neural networks. We have started by presenting the limitations of reservoir computing models such as ESNs. Their fading memory characteristic, which is a strength in certain situations, can become a weakness when dealing with data presenting long-term dependencies. We have thus proposed a reservoir computing model, Gated ESN, and its variant Gated ESN RZ, for overcoming those limitations by the use of gating mechanisms.

To allow the Gated ESN to escape a strict fading memory regime, we have derived a general bound that links the weight matrices of all GRU-based gated models (GRU, Gated ESN, and Gated ESN RZ) to their ability to escape such regime. This gives a means of initializing or verifying these networks for the desired behaviour.

We have performed an experimental comparison between the different models under consideration by testing the generalization performance on a Question Classification task that was chosen for its suitedness to highlight the effect of the gates. We have discovered that gates *can* indeed provide advantages even to reservoir computing models. In addition, we have shown that the use of backpropagation through time drastically increases the

time required for training. We have then verified that the experimental results match the theoretical bound that we have derived.

While the results of this work provide insights about gated reservoir computing, we have also critically discussed the reasons why we believe a pure gated reservoir computing model to be ineffective in practice. We have then suggested directions to produce efficient gated models that join reservoir computing techniques with trained gate dynamics, with an important focus on local training algorithms.

Looking ahead, we believe that the key for efficient and effective RNN models is to be found in the form of an interplay between a suitable gated architecture and a suitable local training algorithm.

Acknowledgments. This work has been partially supported by the European Union's Horizon 2020 Research and Innovation program, under project TEACHING (Grant agreement ID: 871385), URL `https://www.teaching-h2020.eu`, and by the project BrAID under the Bando Ricerca Salute 2018 – Regional public call for research and development projects aimed at supporting clinical and organisational innovation processes of the Regional Health Service – Regione Toscana.

References

1. Babinec, S., Pospichal, J.: Gating echo state neural networks for time series forecasting. In: ICONIP (1). Lecture Notes in Computer Science, vol. 5506, pp. 200–207. Springer (2008)
2. Bellec, G., Salaj, D., Subramoney, A., Legenstein, R.A., Maass, W.: Long short-term memory and learning-to-learn in networks of spiking neurons. In: NeurIPS. pp. 795–805 (2018)
3. Bellec, G., Scherr, F., Hajek, E., Salaj, D., Subramoney, A., Legenstein, R.A., Maass, W.: Eligibility traces provide a data-inspired alternative to backpropagation through time. In: Neuro AI Workshop, NeurIPS (2019)
4. Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., Maass, W.: A solution to the learning dilemma for recurrent networks of spiking neurons. bioRxiv p. 738385 (2019)
5. Bengio, Y., Simard, P.Y., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Networks 5(2), 157–166 (1994)
6. Bianchi, F.M., Scardapane, S., Løkse, S., Jenssen, R.: Bidirectional deep-readout echo state networks. In: ESANN (2018)
7. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014. pp. 1724–1734 (2014)
8. Collins, J., Sohl-Dickstein, J., Sussillo, D.: Capacity and trainability in recurrent neural networks. In: ICLR (Poster). OpenReview.net (2017)
9. Di Sarli, D., Gallicchio, C., Micheli, A.: Gated echo state networks: a preliminary study. In: INISTA. pp. 1–5. IEEE (2020)
10. Di Sarli, D., Gallicchio, C., Micheli, A.: Text classification by untrained sentence embeddings. Intelligenza Artificiale 14(2), 245–259 (2020)
11. Gallicchio, C., Micheli, A.: Architectural and Markovian factors of echo state networks. Neural Networks 24(5), 440–456 (2011)
12. Gonon, L., Ortega, J.P.: Fading memory echo state networks are universal. Neural Networks (2021)
13. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)

14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)
15. Izhikevich, E.M.: Solving the distal reward problem through linkage of stdp and dopamine signaling. Cerebral cortex 17(10), 2443–2452 (2007)
16. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks – with an erratum note'. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report (2001)
17. Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science 304(5667), 78–80 (2004)
18. Jaeger, H., Lukosevicius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky-integrator neurons. Neural Networks 20(3), 335–352 (2007)
19. Li, X., Roth, D.: Learning question classifiers. In: 19th International Conference on Computational Linguistics, COLING 2002 (2002)
20. Lipton, Z.C., Berkowitz, J., Elkan, C.: A critical review of recurrent neural networks for sequence learning. CoRR 1506.00019 (2015)
21. Lukosevicius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Comput. Sci. Rev. 3(3), 127–149 (2009)
22. Miconi, T.: Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. Elife 6, e20899 (2017)
23. Minsky, M.: Steps toward artificial intelligence. Proceedings of the IRE 49(1), 8–30 (1961)
24. Nøkland, A.: Direct feedback alignment provides learning in deep neural networks. In: NIPS. pp. 1037–1045 (2016)
25. Popov, A., Koprinkova-Hristova, P., Simov, K., Osenova, P.: Echo state vs. lstm networks for word sense disambiguation. In: International Conference on Artificial Neural Networks. pp. 94–109. Springer (2019)
26. Ramamurthy, R., Stenzel, R., Sifa, R., Ladi, A., Bauckhage, C.: Echo state networks for named entity recognition. In: ICANN (Workshop). Lecture Notes in Computer Science, vol. 11731, pp. 110–120. Springer (2019)
27. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45(11), 2673–2681 (1997)
28. Simov, K.I., Koprinkova-Hristova, P.D., Popov, A., Osenova, P.: Word embeddings improvement via echo state networks. In: INISTA. pp. 1–6. IEEE (2019)
29. Subramoney, A., Scherr, F., Maass, W.: Reservoirs learn to learn. CoRR abs/1909.07486 (2019)
30. Tiňo, P., Hammer, B., Bodén, M.: Markovian bias of neural-based architectures with feedback connections. In: Perspectives of Neural-Symbolic Integration, Studies in Computational Intelligence, vol. 77, pp. 95–133. Springer (2007)
31. Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. Neural Networks 20(3), 391–403 (2007)
32. Wang, X., Jin, Y., Hao, K.: A gated recurrent unit based echo state network. In: IJCNN. pp. 1–7. IEEE (2020)
33. Yildiz, I.B., Jaeger, H., Kiebel, S.J.: Re-visiting the echo state property. Neural Networks 35, 1–9 (2012)

Daniele Di Sarli has conducted research on Recurrent Neural Networks and Reservoir Computing at the Department of Computer Science, University of Pisa, Italy. The main focus of his research concerns the study of the effectiveness of Reservoir Computing approaches.

Claudio Gallicchio is an Assistant Professor of Machine Learning at the Department of Computer Science of the University of Pisa, Italy. His research is based on the fusion

of concepts from Deep Learning, Recurrent Neural Networks, and Randomized Neural Systems.

Alessio Micheli is an Associate Professor at the Department of Computer Science, University of Pisa, Italy. His main research lines are in the field of Machine Learning and Neural Networks, with a pioneering research activity since the end of 90's for learning in structured domains (sequence, tree and graph data).

Received: February 18, 2021; Accepted: August 31, 2021.

Computer Science and Information Systems 19(1):397–414 https://doi.org/10.2298/CSIS210313058S

A Comparison of Deep Learning Algorithms on Image Data for Detecting Floodwater on Roadways*

Salih Sarp[1], Murat Kuzlu[2], Yanxiao Zhao[1], Mecit Cetin[2], and Ozgur Guler[3]

[1] College of Engineering, Virginia Commonwealth University,
Richmond, USA
{sarps, yzhao7}@vcu.edu
[2] Batten College of Engineering & Technology,
Old Dominion University, Norfolk, VA, USA
{mkuzlu, mcetin}@odu.edu
[3] eKare, Inc., Fairfax, VA, USA
oguler@ekare.ai

Abstract. Object detection and segmentation algorithms evolved significantly in the last decade. Simultaneous object detection and segmentation paved the way for real-time applications such as autonomous driving. Detection and segmentation of (partially) flooded roadways are essential inputs for vehicle routing and traffic management systems. This paper proposes an automatic floodwater detection and segmentation method utilizing the Mask Region-Based Convolutional Neural Networks (Mask-R-CNN) and Generative Adversarial Networks (GAN) algorithms. To train the model, manually labeled images with urban, suburban, and natural settings are used. The performances of the algorithms are assessed in accurately detecting the floodwater captured in images. The results show that the proposed Mask-R-CNN-based floodwater detection and segmentation outperform previous studies, whereas the GAN-based model has a straightforward implementation compared to other models.

Keywords: Floodwater detection; Mask-R-CNN; GAN; object detection and segmentation.

1. Introduction

Monitoring and sensing roadway conditions automatically are critical not only for self-driving vehicles but also for informing the traveling public. Due to various environmental and weather-related factors (e.g., heavy snow, rain, storm surge), roadway conditions can dramatically change, and a given road segment may be inundated or may operate under a reduced capacity. The vulnerability of roadways to floodwater is increasingly becoming a major concern for numerous communities, especially for those living in the coastal regions, since recurrent flooding occurs more frequently due to sea-level rise, storm surge, heavy rain, and tidal flooding [1]. As many societal functions depend on a functioning transportation infrastructure, i.e., routing of emergency vehicles and delivery of goods and services to support commerce, there needs to be a scalable and effective system to monitor or predict the inundations

* This is an extended version of a conference paper, INISTA 2020.

on roadways [2]. The main goal of this paper is to contribute to the development of such a system by showing how image-based sensing and detection techniques could be utilized to detect floodwater present on the roadways.

Over the last five years, object detection and segmentation have been evolving rapidly, where new approaches are being invented, and new application areas are emerging. Previous studies for the detection and segmentation of floodwater focused on remote sensing methods, which leverage aerial photographs and radar data [3-4]. A survey of large areas can be done with these methods, whereas detailed information about an area, especially a roadway, needs a more localized approach. Synthetic Aperture Radar (SAR) images are used by Kang et al. [5] with Fully Convolutional Network (FCN) implementation. These approaches lack important local information, such as the type, severity, and extent of floodwater, which are crucial for safe driving. Ultrasonic rangefinder and passive infrared temperature sensors are utilized by Mousa et al. [6] to detect floods in urban cities with Artificial Neural Network (ANN). Their implementation requires special sensor placement, which will increase the cost and effort to detect and segment the floodwater. Authors in [7] propose a flood detection model where Region-Based Convolutional Neural Networks (R-CNN) architecture is used to preprocess the images.

In this paper, the recently proposed Mask-R-CNN and Generative Adversarial Networks (GAN) [8] methods will be utilized to detect and segment the floodwater on roadways. Mask-R-CNN is built on the Faster-R-CNN algorithm and improves the segmentation performance [9]. Mask-R-CNN is a deep learning algorithm belonging to the Region-Based Convolutional Neural Networks (R-CNN) family of object detection and semantic segmentation models. As the latest evolution in the R-CNN family, Mask-R-CNN fuses localization, classification, and segmentation in a compact and fast algorithm. GAN algorithm is another deep learning algorithm that is widely used in a variety of applications. It is used in a conditional setting to segment flooded areas in the images. The implementation of Mask-R-CNN and GAN models on detecting and segmenting the floodwater on roadways has not been investigated previously. The main contributions of this study include: (i) Applying Mask-R-CNN and GAN models to both detect whether a given image contains floodwater and, if so, to segment the image to delineate the floodwater; and (ii) Demonstrating that the proposed approaches yield more accurate results than alternative approaches. The proposed deep learning models do not require manual extractions of any features and are tested on various images collected under different conditions.

The remainder of this paper is organized as follows. Section 2 introduces the background of proposed algorithms Mask-R-CNN and GAN. Section 3 presents the methodology. Section 4 includes data collection steps and the preparation of the dataset for processing. Results and related discussions are examined in Section 5. Conclusions and future work are provided in Section 6.

2. Background

Image segmentation is one of the essential processes in the image processing field [10]. Image segmentation tasks include the division of the image into various regions by utilizing similar and specific properties in the image [11]. Deep learning techniques

contribute significantly to image processing, especially in image segmentation. In this section, R-CNN and GAN methods will be discussed in detail.

2.1. Regions with CNN features (R-CNN)

One of the common model families for object detection and segmentation is R-CNN [12]. Four different methods have evolved over the years, namely R-CNN, Fast-R-CNN, Faster-R-CNN, and lastly, Mask-R-CNN.

R-CNN, Regions with CNN features, was developed with a three-step structure in 2014 [13]. The first part of this structure is the selective search, which proposes bounding boxes of around 2000 regions called 'Region of Interest' (RoI) [13]. These proposed regions are fed to CNN to compute the features of each proposed RoI. This step requires very high computing power and is time-consuming. Class-specific linear Support Vector Machines (SVM) are employed for the classification of each region.

The Fast-R-CNN method, Fast Region-based Convolutional Neural Network, is an improved version of R-CNN, which requires a long time for object detection. The calculation of the ConvNet forward-pass independently for each RoI is the main reason why R-CNN is slow [14]. The Fast-R-CNN takes an image and produces a convolutional (Conv) feature map from the entire image, which is shared for each RoI. On the other hand, R-CNN calculates the feature map for each RoI, which takes a lot of time. Fast-R-CNN then extracts the feature vector for the corresponding RoI, which is then fed to a fully connected (FC) layer. FC layer has two outputs: SoftMax probability output for classification and a real-valued output for the position of the bounding box of classes.

The Faster-R-CNN is developed to speed up Fast-R-CNN. The significant drawback of Fast-R-CNN is the separate generation of region proposals. Girshick [14] indicates that the generation of region proposals can be implemented by Conv feature maps used in Fast-R-CNN. Region Proposal Networks (RPNs) are developed as an attention mechanism by Ren et al. [15] and fused with Fast-R-CNN, which harnesses shared computation, to speed up the object detection. The dual structure of Faster-R-CNN decreased the time needed for object detection.

2.2. Generative Adversarial Networks (GAN)

GAN algorithm is developed by Goodfellow [8] that consists of two hostile networks, namely Generator (G) and Discriminator (D) networks. These two networks are simultaneously trained to pick the underlying statistical data distribution of the training set.

Different GAN models are proposed after its first introduction for various image-related applications. One of the most used GAN models, DCGAN, is proposed by Radford et al. [16]. DCGAN utilizes the feature extraction capabilities of CNNs to generate natural images. The elimination of the fully connected layer, use of batch normalization, replacing pooling layers with strided and fractional-strided convolutions improve the produced image quality.

The use of GAN in a conditional setting is implemented by forcing the generated images to follow certain data distribution. Conditional Generative Adversarial Networks (cGAN) models have been used to successfully tackle a wide variety of problems [17]. Future state prediction [18], image manipulation by user constraints [19], super-resolution [20], inpainting [21], and style transfer [22] are some of the application areas that generate significant results.

3. Methodology

This study employs Mask-R-CNN and cGAN based architectures to segment the inundated areas on roadways. These two deep learning techniques will be implemented separately using the same dataset. Our study will reveal that both models have a high potential to achieve successful segmentation results.

3.1. Mask-R-CNN

Mask-R-CNN is the fourth generation of the R-CNN object detection family by adding a third branch that implements fine pixel-to-pixel alignment for segmentation. The Faster-R-CNN framework is modified to predict an object mask [23]. The RoI pooling layer is changed with the Region of Interest Align (RoIAlign) layer to align the extracted features precisely with the input [9]. A fully convolutional neural network is applied to each RoI to produce a segmentation mask. Mask and class predictions are also decoupled to improve instance segmentation [24]. As shown in Fig. 1, localization, classification, and segmentation are the three tasks that Mask-R-CNN implements.

The Mask-R-CNN has a two-stage procedure that consists of a region proposal network and three networks fed by RoI Align, as seen in Fig. 1. The proposed Mask-R-CNN model in this paper is built utilizing a Feature Pyramid Network (FPN) [25] for feature extraction and adopted ResNet101 [26] as the backbone in the first stage. This CNN has four layers of convolution and one deconvolution layer. The RoIAlign layer keeps the size of the feature map the same and avoids misalignment by avoiding quantization. Anchors are used for region proposals. They are preassigned, and our model follows the same steps as in [9]. The model checks if there is any object inside the anchor, then it moves to each pixel in the feature map. After refining the anchor coordinates, it returns the bounding boxes as object proposals. Multi-task loss (L), bounding-box loss (L_{box}), classification loss (L_{cls}), and mask loss (L_{mask}) are identical to the previous studies [9-15]. The multi-task loss is defined as $L = L_{box} + L_{cls} + L_{mask}$ [27]. The Tensorboard visualization tool is used to track these loss metrics. The features are fed into three networks. The mask prediction branch has four convolutions and one deconvolution layer followed by the ReLU activation function to predict flood masks. The returned predicted mask has a size of 28x28 to decrease the computational complexity, which provides faster instance segmentation. On the other hand, image segmentation on downscaled images (28x28) could result in artifacts after being scaled up for inference. There are two common FC layers on the classification and bounding-box branch. This assists the classification network to classify only the inside of the bounding box. The final FC layers consist of 2 layers.

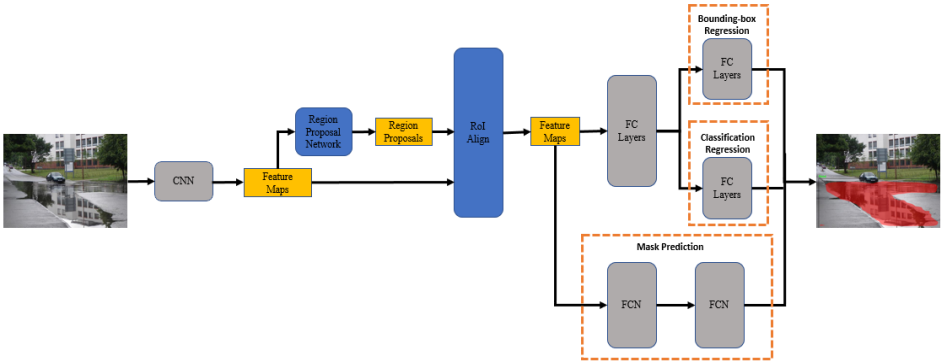


Fig. 1. Framework of the proposed method

3.2. Conditional Generative Adversarial Networks (cGAN)

Conditioning the output to a given input is achieved by cGAN, which evolves from the vanilla GAN model. In the vanilla GAN model, generator and discriminator networks form the GAN architecture. These two separate networks are trained simultaneously to compete with each other. The generator network uses the given random uniform or Gaussian noise (z) to mimic the training data distribution for generating the fake image (y), G: z $\rightarrow$ y [28]. The generated images by the generator network are then classified as real or fake by the discriminator network during the training stage, i.e., D: y $\rightarrow$[0, 1]. This two-player zero-sum game continues until satisfactory image generation is achieved [29]. Training of GAN algorithms is subject to research, and it is a continuing effort to handle efficient training of GAN models [30]. After the training, the generator network is used for the generation of new images.

The cGAN architecture utilizes an additional conditioning input image (x) which provides further benefits in addition to the vanilla GAN architecture. cGAN model outshines other GAN-based architectures because of its simplicity, fast and straightforward implementation. Generator network gets input image (x) and noise (z) to produce an image (y) that shares similar characteristics with the input image and similar data distribution with the training set, G: {x, z}$\rightarrow$ y [17]. Discriminator network also observes the input image (x) so as to determine if the output of the generator image (y) is real or fake, D: {x, y}$\rightarrow$ [0,1] [31]. Generator network design is based on U-Net [32] encoder-decoder with skip connections architecture. Discriminator network utilizes PatchGAN architecture to increase the generated image quality [17].

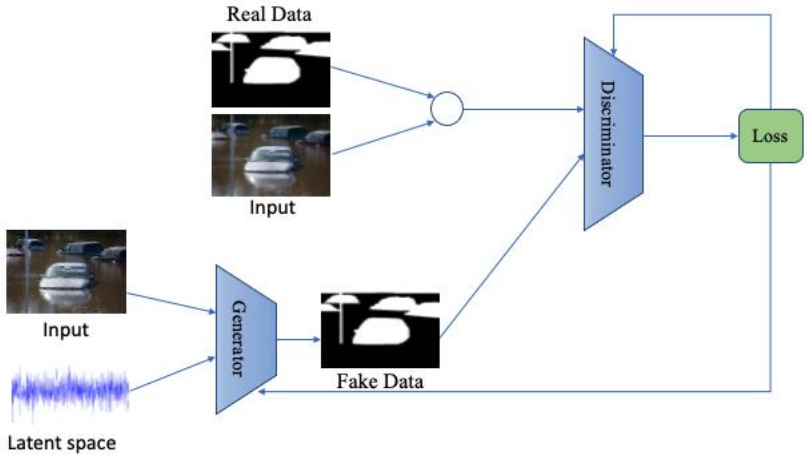


Fig. 2. Framework of the proposed method

The GAN algorithm utilizes four different loss functions during the training to generate decent fake images. D_real and D_fake loss functions for real and fake samples directly update the discriminator network. Generator network weights are updated with two loss functions: adversarial loss (G_GAN) from the discriminator network and L1 loss (G_L1). Training and update of the discriminator network are straightforward, whereas generator network training is done indirectly via the discriminator network. This is one of the reasons why the training of GAN is difficult [33]. Adversarial loss objective is depicted as:

$$\mathrm{G\_GAN(G, D)} = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}\left[\log\left(1 - D\big(x, G(x,z)\big)\right)\right] \quad (1)$$

The overall objective loss function is expressed with regularizing parameter (λ) as:

$$G^* = argmin_G max_D G\_GAN(G,D) + \lambda G\_L1(G) \quad (2)$$

4. Data Collection

The data collection, preparation, validation, and training environment details will be presented in this section.

4.1. Data Collection and Preparation

The flood image dataset, which includes urban, suburban, and natural areas, is manually labeled for training and testing [34]. The variety of scenes improves the applicability of the algorithm implemented in this paper. The size of the images and labels is shaped as 385x512 pixels for the Mask-R-CNN-based model and the cGAN based model. Flood

images are hand-labeled at the pixel level where flooded areas have the pixel value of one, and the rest of the image has the pixel value of zero. MATLAB labeler tool is used for the labeling. Around 500 images contain floodwater, while another 250 images are dry. Twenty-five (25) crowdsourced images from the internet are used for testing. Some of the images used in this study can be seen below in the result section. Red-colored areas are detected and segmented as floodwater.

4.2. Environment

We implemented the proposed Mask-R-CNN model using the Keras deep learning and Tensorflow libraries with Python 3.6. We used the pre-trained weights for MS COCO and Matterport's code [35] as a starting point. Our model is trained using an Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40 GHz with 128 GB memory and NVIDIA Quadro K4200 with 4 GB dedicated and 64 GB shared memory. We trained our model for 20 epochs which took approximately 24 hours. We use a learning rate of 0.001 where the original paper used 0.02 because it causes weights to blow up in Tensorflow. The rest of the hyperparameters were the same as those in the original Mask-R-CNN article [14].

cGAN based model is implemented using PyTorch machine learning library with Python 3.8. Pix2pix framework [18] is utilized to create the proposed cGAN-based segmentation model. The study with the cGAN based model is realized using Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz with 32 GB memory and NVIDIA GeForce GTX 1650 Ti GPU with 4 GB dedicated and 16 GB shared memory. We trained our model for 200 epochs which took approximately 2 hours. We choose a batch size of 20 to fully leverage GPU power. That's why the training of the GAN algorithm took less time. Adam optimizer and a learning rate of 0.002 are employed during the training.

4.3. Validation

The performances of the algorithms are evaluated based on four common metrics: accuracy, precision, recall, and F1-score [36]. Since a binary classification problem is being solved, each pixel will fall into one of the two possible categories: "dry" indicating no floodwater and "flood" for the opposite. The performance metrics are defined below :

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1score = 2x\frac{Precision\ x\ Recall}{Precision + Recall} \tag{6}$$

Where True Positive (TP) is the number of correctly predicted flood pixels; False Positive (FP) is the number of pixels that are incorrectly classified as dry; True Negative (TN) is the number of correctly predicted flood pixels; False Negative (FN) is the number of pixels that are incorrectly classified as dry.

5. Result and Discussion

This section will present and discuss the outputs of the Mask-R-CNN and cGAN based models in terms of performance. The dataset consisting of 441 flood and 238 dry images is divided into training (75%) and validation (25%) subsets.

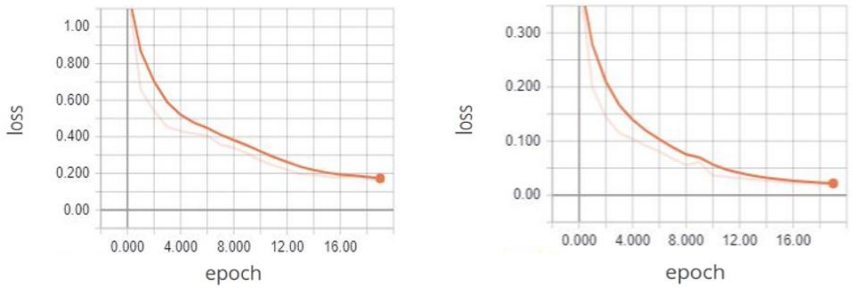
5.1. Results of Mask-R-CNN based model

The proposed Mask-R-CNN-based model encapsulates three different networks. The three networks address predicting the bounding box of the RoI, the classification, and the instance segmentation. All these three networks are trained together during training.

Accuracy is the measurement of correctly classified instances, which is depicted in equation (3). According to the results from the testing samples, the floodwater/dry classification accuracy of the Mask-R-CNN model is 99.2%, and the segmentation accuracy of the flooded area is 93.0%. Compared to cGAN based model and a recent study [2], Mask-R-CNN yields better accuracy in segmentation and classification tasks.

The loss function is used to evaluate how well the machine learning algorithms perform in the training phase. The model weights are regulated to minimize the loss function. Figs. 3 and 4 show the loss graphs of the total, bounding box, classification, and segmentation mask loss. Each figure shows the number of epochs on the X-axis and the output of the loss function on the Y-axis.

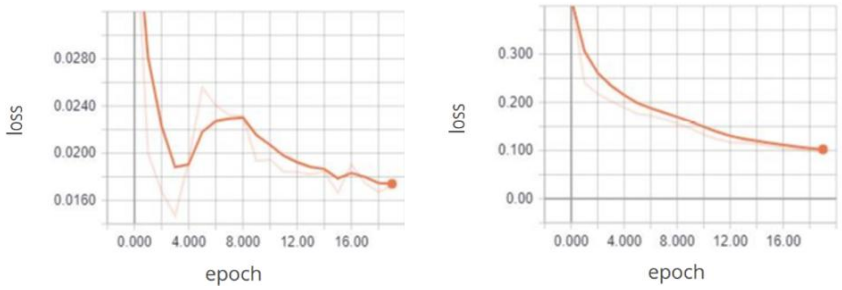
Furthermore, the bold line shows a smoothed version of the actual output shown in a dimmer line for better readability. The overall downward trend of the curves shows the adequate nature of the network modeling flood detection, which is a sign for the model learning the inherent pattern of flooding in the images.



a) b)

Fig. 3. (a) Multi-task and (b) bounding box loss graphs

The loss graphs indicate that the model successfully learns the properties of each image. Fig. 3 and Fig. 4 show the same decreasing nature.



a) b)

Fig. 4. (a) Classification and (b) segmentation mask loss graphs

The validation loss is another metric like training loss, but it is not used to update the weights. It is used for monitoring if the model generalizes well. Checking validation loss helps to avoid overfitting. If the validation loss does not improve for a long time, model training will need to be stopped early. The total validation loss is shown in Fig. 5. for 30 epochs. After 20 epochs, the loss tends to increase, indicating the potential for overfitting.

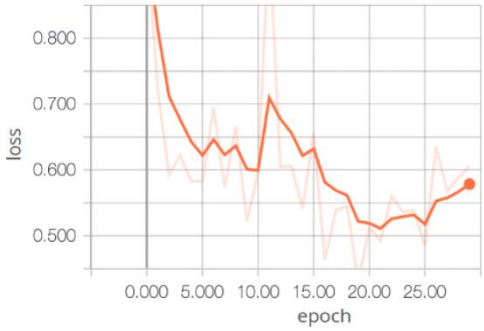
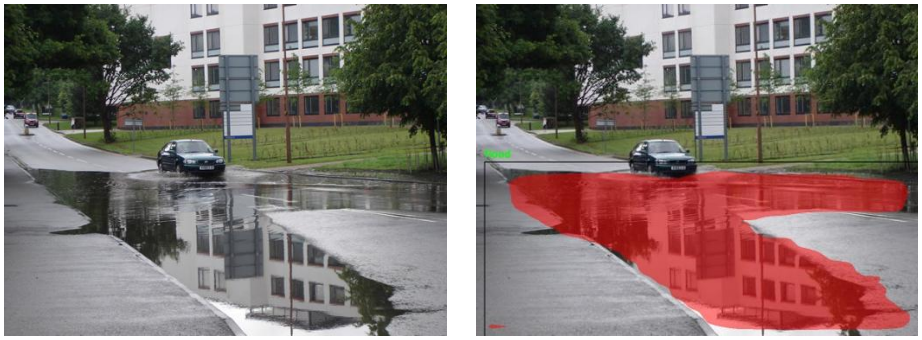


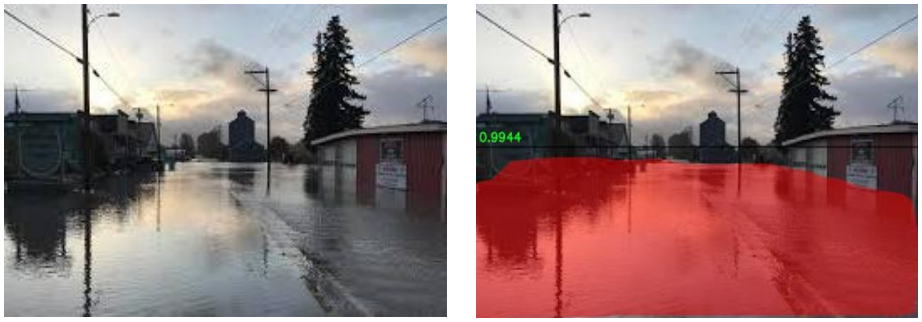
Fig. 5. Total validation loss graph for 30 epochs

Fig. 6 to Fig. 10 depict flood segmentation and original images side by side. The predicted region, as floodwater by the Mask-R-CNN, is marked in red. Since the model first determines a bounding box and then segments within this box, there may be straight lines demarcating the floodwater boundaries in the scene. Overall, the model captures the floodwater boundaries reasonably well.



(a) (b)

Fig. 6. (a) Original and (b) segmented image of a suburban scene. Segmented flood regions are overlaid red



(a) (b)

Fig. 7. (a) Original and (b) segmented image of a suburban scene. Segmented flood regions are overlaid red

(a) (b)

Fig. 8. (a) Original and (b) segmented image of a roadway with a red overlay

(a) (b)

Fig. 9. (a) Original and (b) segmented image of an urban scene with a red overlay

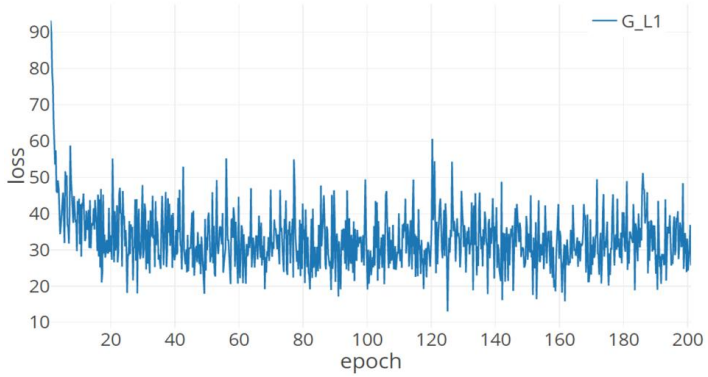
(a) (b)

Fig. 10. (a) Original and (b) segmented image of a natural scene with a red overlay

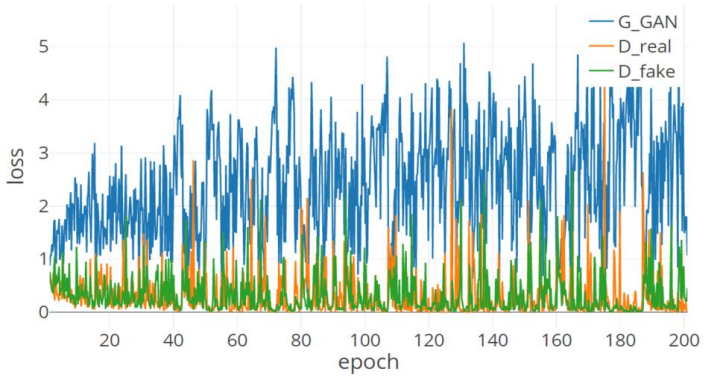
5.2. Results of cGAN-based model

The cGAN-based model has distinctive training characteristics as a different framework (PyTorch) was used in comparison to the Mask-R-CNN-based model (Tensorflow). There are four different loss functions that are used in the GAN model, D_real, D_fake, G_GAN, and G_L1. D_real and D_fake loss functions for real and fake samples directly update the discriminator network. Generator network weights are also updated with two loss functions, adversarial loss (G_GAN) from discriminator network and L1 loss (G_L1) [37]. The training of discriminator and generator networks is a zero-sum game and causes a non-converging problem [38]. Therefore, the training progress cannot be understood by the loss graph alone. The loss graphs of cGAN based model extracted from the visdom virtualization package are indicated in Fig. 11.

The segmentation accuracy of the cGAN based model is 67%. The performance of the cGAN model is low in contrast to previous studies and the proposed Mask-R-CNN model regarding the accuracy parameter. This is an expected result as GAN models need more images to converge. Sample images from the test set are shown in Figs. 12-16.

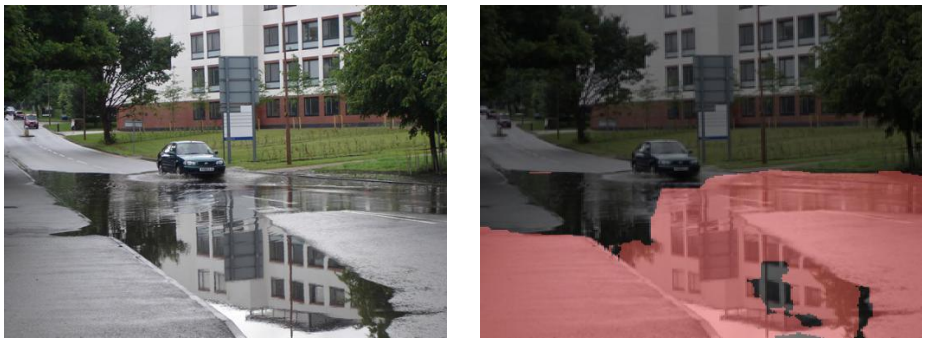


(a)



(b)

Fig. 11. Graph of L1 loss (a) and other three different loss functions (b)



(a) (b)

Fig. 12. (a) Original and (b) segmented image of a suburban scene. Segmented flood regions are overlaid red

(a) (b)

Fig. 13. (a) Original and (b) segmented image of a suburban scene. Segmented flood regions are overlaid red

(a) (b)

Fig. 14. (a) Original and (b) segmented image of a roadway with a red overlay.

(a) (b)

Fig. 15. (a) Original and (b) segmented image of an urban scene with a red overlay.

(a) (b)

Fig. 16. (a) Original and (b) segmented image of a natural scene with a red overlay.

5.3. Comparison of the models

Table 1 compares a recent study [39] with the proposed Mask-R-CNN and GAN models. Our Mask-R-CNN model has better precision, recall, and F1-score results over the Fully Convolutional Neural Network (FCN) based on a pre-trained VGG-16 network [2].

Table 1. Model comparison with a previous study

Model	Precision	Recall	F1-score
Mask-R-CNN	0.96	0.96	0.96
GAN	0.88	0.73	0.80
FCN [2]	0.92	0.90	0.91

Results indicate that Mask-R-CNN achieves better accuracy over cGAN based model and previously proposed models [2]. Object reflections have adverse effects on segmentation accuracy. Training our model on a larger dataset with more reflection cases will increase the model's accuracy. The downscaling of images, as explained previously, to reduce computational complexity may also decrease segmentation accuracy. Instance segmentation on downscaled images provides faster results, which is essential for autonomous vehicles.

The Mask-R-CNN performs segmentation on a rescaled (smaller-28x28) version of the original image (385x512). This has the side effect that the segmentation pixel resolution is very coarse. Figs. 8 and 9 show this artifact very well. The headlights of the car are labeled as a flood in Fig. 8. This could be improved by modifying the Mask-R-CNN to work on a similar size to the original image. However, the downside of increased image resolution is that learning will only converge with a much bigger image dataset because the number of parameters would increase exponentially.

cGAN-based model suffers from a class imbalance problem as it could segment the flooded areas in the lower half of the image better. When the same image in Fig. 9 and

Fig. 15 are compared, cGAN based model achieves better segmentation for the flooded areas around the car. We envision that the cGAN-based model has significant room for improvement if a sufficient dataset is provided.

6. Conclusion

This paper presents state-of-the-art deep learning algorithms, Mask-R-CNN and cGAN, to the segmentation of inundated sections of the roadways, on an image dataset consisting of urban, suburban, and natural scenes. The models are trained to detect floodwater in these images and delineate its boundaries using segmentation techniques. The proposed Mask-R-CNN model achieves 99% and 93% accuracy for floodwater detection and segmentation tasks, respectively. The graphs of the loss functions of the three encapsulated networks, which are classification, bounding box detection, and segmentation mask networks, indicate the efficient training of the model.

The proposed cGAN based model underperforms in comparison to Mask-R-CNN and previous FCN based models. The main reason for the low performance of the cGAN based model is the limited dataset size, which was insufficient to gather data distribution of the training set by the GAN algorithm. However, its straightforward implementation, fast, and apparent structure provide significant potential. With the increasing importance of data collection, the underlying dataset size problem could be resolved. Mask-R-CNN also has many benefits, but it is expensive in terms of time and memory consumption.

As a future study, the floodwater depth prediction will be investigated. The flood level will be determined using nearby objects, such as cars and traffic light poles. In addition to flood depth prediction, the speed of flood detection algorithms will be investigated as well. The dataset size will be increased to get a better result. Finally, water reflections will be studied to improve flood segmentation accuracy.

References

1. W. V. Sweet and J. Park, “From the extreme to the mean: Acceleration and tipping points of coastal inundation from sea-level rise,” Earth's Future, vol. 2, no. 12, pp. 579–600, Dec. 2014.
2. C. Sazara, M. Cetin, and K. M. Iftekharuddin, "Detecting floodwater on roadways from image data with handcrafted features and deep transfer learning," 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 2019, pp. 804-809.
3. N. M. Robertson and T. Chan, "Aerial image segmentation for flood risk analysis," 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, 2009, pp. 597-600.
4. Klemas V. 2014. Remote sensing of floods and flood-prone areas. J. Coastal Res. 31(4):1005–1013.
5. Kang, Wenchao, Yuming Xiang, Feng Wang, Ling Wan, and Hongjian You. "Flood Detection in Gaofen-3 SAR Images via Fully Convolutional Networks." Sensors 18.9 (2018): 2018. Web.

6. Mousa, M., Xiangliang Zhang, & Claudel. (2016). Flash Flood Detection in Urban Cities Using Ultrasonic and Infrared Sensors. IEEE Sensors Journal, 16(19), 7204-7216.
7. Megan A. Witherow, Cem Sazara, Irina M. Winter-Arboleda, Mohamed I. Elbakary, Mecit Cetin & Khan M. Iftekharuddin (2019) Floodwater detection on roadways from crowdsourced images, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 7:5-6, 529-540, DOI: 10.1080/21681163.2018.1488223
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial networks. arXiv preprint arXiv:1406.2661.
9. He, Kaiming, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. "Mask R-CNN." IEEE Transactions on Pattern Analysis and Machine Intelligence PP.99 (2018): 1. Web.
10. Manisha, M. and Mithra, K.S., Various Image Segmentation Techniques: A Review.
11. Tongbram, S., Shimray, B.A., Singh, L.S. and Dhanachandra, N., 2021. A novel image segmentation approach using fcm and whale optimization algorithm. Journal of Ambient Intelligence and Humanized Computing, pp.1-15.
12. Weng, L., Object Detection for Dummies Part 3: R-CNN Family. 2017. http://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html
13. Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." ArXiv.org (2014): ArXiv.org, Oct 22, 2014. Web.
14. Girshick, Ross. "Fast R-CNN." ArXiv.org (2015): 27. Web.
15. Shaoqing Ren, R., Kaiming He, Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE Transactions on Pattern Analysis and Machine Intelligence 39.6 (2017): 1137-149. Web.
16. Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
17. Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
18. Y. Zhou and T. L. Berg. Learning temporal transformations from time-lapse videos. In ECCV, 2016.
19. J.-Y. Zhu, P. Kra ̈henbu ̈hl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image mani-fold. In ECCV, 2016.
20. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Te- jani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single im- age super-resolution using a generative adversarial network. In CVPR, 2017.
21. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In CVPR, 2016.
22. C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. ECCV, 2016.
23. Johnson, J.W., 2018. Adapting mask-rcnn for automatic nucleus segmentation. arXiv preprint arXiv:1805.00500.
24. Cheng, B., Wei, Y., Shi, H., Feris, R., Xiong, J., and Huang, T., 2018. Revisiting rcnn: On awakening the classification power of faster rcnn. In Proceedings of the European conference on computer vision (ECCV) (pp. 453-468).
25. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S., 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).
26. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
27. Zimmermann, R.S. and Siems, J.N., 2019. Faster training of Mask R-CNN by focusing on instance boundaries. Computer Vision and Image Understanding, 188, p.102795

28. Huang, H., Yu, P.S. and Wang, C., 2018. An introduction to image synthesis with generative adversarial nets. arXiv preprint arXiv:1803.04469.
29. Ge, H., Xia, Y., Chen, X., Berry, R., and Wu, Y., 2018. Fictitious gan: Training gans with historical models. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 119-134).
30. Sarp, S., Kuzlu, M., Wilson, E. and Guler, O., 2021. WG2AN: Synthetic wound image generation using generative adversarial network. The Journal of Engineering, 2021(5), pp.286-294.
31. Bakkay, M. C., Rashwan, H. A., Salmane, H., Khoudour, L., Puig D., and Ruichek, Y., "BSCGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 4018-4022.
32. Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
33. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., 2016. Improved techniques for training gans. In Advances in neural information processing systems (pp. 2234-2242).
34. Sazara, Cem; Cetin, Mecit; Iftekharuddin, Khan (2019), "Image Dataset for Roadway Flooding," Mendeley Data, v1http://dx.doi.org/10.17632/t395bwcvbw.1
35. W. Abdulla. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. 2017. Accessed on: February 20, 2020. [Online]. Available: https://github.com/matterport/Mask_RCNN.
36. 36. Sarp, S., Kuzlu, M., Wilson, E., Cali, U. and Guler, O., 2021. The Enlightening Role of Explainable Artificial Intelligence in Chronic Wound Classification. Electronics, 10(12), p.1406.
37. Sarp, S., Kuzlu, M., Pipattanasomporn, M. and Guler, O., 2021. Simultaneous wound border segmentation and tissue classification using a conditional generative adversarial network. The Journal of Engineering, 2021(3), pp.125-134.
38. Goodfellow, I., 2016. NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160.
39. Sarp, S., Kuzlu, M., Cetin, M., Sazara, C. and Guler, O., 2020, August. Detecting Floodwater on Roadways from Image Data Using Mask-R-CNN. In 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)(pp. 1-6). IEEE.

Salih Sarp (Student Member IEEE) received the B.S degree from Dogus University, Istanbul, Turkey, in 2014, and the M.Sc. degree in electrical and computer engineering from George Washington University, Washington DC, USA, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Virginia Commonwealth University, USA. His research interests include machine learning, embedded systems and internet of things.

Murat Kuzlu (Senior Member IEEE) joined Old Dominion University (ODU) of Electrical Engineering Technology Department as an Assistant Professor in 2018. He received his B.Sc., M.Sc., and Ph.D. degrees in Electronics and Telecommunications Engineering in 2001, 2004, and 2010, respectively. From 2005 to 2006, he worked as a Global Network Product Support Engineer at the Nortel Networks, Turkey. In 2006, he joined the Energy Institute of TUBITAK-MAM (Scientific and Technological Research Council of Turkey-Marmara Research Center), where he worked as a senior researcher.

Before joining ODU, he worked as a Research Assistant Professor at Virginia Tech's Advanced Research Institute. His research interests include smart grid, demand response, smart metering systems (AMR, AMI, AMM), home and building energy management system, co-simulation, wireless communication and embedded systems.

Yanxiao Zhao (Senior Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA, USA, in 2012. She is currently an Associate Professor with the Electrical and Computer Engineering Department, Virginia Commonwealth University. Her research interests include, but not limited to: the Internet of Things (IoT), 5/6G communications, cyber security, wireless networks, including cognitive radio networks, vehicular networks, wireless autonomous networks, wireless body area networks, software-defined networks, device-to-device (D2D) communications, wireless energy harvesting, and power management and communications in smart grid. Her research has been supported by NSF, NASA, and Air Force. She has published over 70 papers in prestigious journals and international conferences. She has been actively organizing international conferences by serving as a TPC chair, publicity chair, and TPC member. She was a recipient of the Best Paper Award for three international conferences: WASA2009, ChinaCom2016, and ICMIC2019.

Mecit Cetin earned his M.S. degree in Civil Engineering and Ph.D. degree in Transportation Engineering from Rensselaer Polytechnic Institute (RPI), Troy, NY, in 1999 and 2002, respectively. Dr. Cetin has joined the Civil and Environmental Engineering Department at Old Dominion University (ODU) as an assistant professor in August 2008. Prior to that, he had worked as an assistant professor for four years in the Department of Civil and Environmental Engineering at the University of South Carolina (USC), Columbia, SC. Dr. Cetin conducts research in various areas including mining big transportation data, modeling and simulation of traffic operations, congestion pricing, freight transportation, sustainable transportation, traffic signal control, probe vehicle technologies, and system state estimation in transportation networks. Dr. Cetin is currently directing various research projects as the principle investigator totaling more than $1.5M. Dr. Cetin has been working with traffic data from various sensor types, including loops, radar, vehicle classification sensors, Bluetooth, accelerometers within mobile devices, video cameras, GPS, weigh-in-motion sensors, etc., to predict network conditions both on freeways and arterials. His publications include 41 articles in archived journals and 72 refereed international conference proceedings.

Özgür Güler is an imaging scientist specializing in 3D wound imaging and computer vision. Prior to eKare, he was a researcher at the Sheikh Zayed Institute (SZI) for Pediatric Surgical Innovation in Washington DC, where he developed the segmentation and classification algorithms that laid the groundwork of the eKare inSight system. Dr. Güler received his PhD from the Medical University Innsbruck in Austria with focus on image-guided diagnosis and therapy, MS in Computer Science with focus on image-guided surgery and BS in Computer Science from Leopold-Franzens University Innsbruck.

Received: March 13, 2021; Accepted: August 31, 2021.

Computer Science and Information Systems 19(1):415–433 https://doi.org/10.2298/CSIS210211057K

An Approach for Selecting Countermeasures against Harmful Information based on Uncertainty Management

Igor Kotenko, Igor Saenko, Igor Parashchuk, and Elena Doynikova

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS),
39, 14th Liniya, 199178, St. Petersburg, Russia
{ivkote, ibsaen, parashchuk, doynikova}@comsec.spb.ru

Abstract. Currently, one of the big problems in the Internet is the counteraction against the spread of harmful information. The paper considers models, algorithms and a common technique for choosing measures to counter harmful information, based on an assessment of the semantic content of information objects under conditions of uncertainty. Methods of processing incomplete, contradictory and fuzzy knowledge are used. Two cases of the algorithm implementation to eliminate the uncertainties in the assessment and categorization of the semantic content of information objects are analyzed. The first case is focused on processing fuzzy data. The second case is based on using an artificial neural network. An experimental evaluation of the proposed technique have shown that the use of both cases makes it possible to eliminate uncertainties of any type and, thereby, to increase the efficiency of choosing measures to counter harmful information.

Keywords: harmful information, assessment, countermeasures, semantic content, information objects, uncertainty.

1. Introduction

At present the Internet and social networks, which can be represented as large sets of interconnected digital network information objects, are becoming one of the most important threats to personal, public and state information security. This determines the need to protect the individual, society and the state from information that spreads through information and telecommunication networks and is capable to harm the health of citizens or motivate them to unlawful behavior. For example, the United States has laws that protect children's Internet and protect children from harmful content posted on the Internet. In the UK, Canada and many other countries, systems are used to block blacklisted sites with harmful content. However, the presence of such systems responsible for blocking harmful content on the Internet and social networks does not mean that the problem of protecting against harmful information has been solved. At the moment, the detection of harmful sites and messages and the formation of black lists is carried out, as a rule, in manual mode.

In scientific and methodological terms, the problem of protection from harmful information has only a small number of scientific and technical solutions. We can say that the methodology for countering harmful information is at the initial stage of

development and implementation. This is fully true for the task of choosing measures to counter harmful information. The solution to this problem should be based on solutions for the development and application of content analysis tools, as well as software and hardware for detecting, evaluating and countering harmful information. At the same time, the concept of harmful (detrimental, dangerous, destructive) information means such information that is prohibited from being distributed on the Internet or social networks by current legislation.

Determination of reliable estimates of digital network content requires that, in order to increase the objectivity of its analysis and make adequate decisions to counter harmful information, data processing is carried out taking into account their uncertainty. This task is of particular relevance for making decisions and choosing specific measures to counter harmful information in real-life conditions. Consequently, models, algorithms and methods for evaluating information objects, as well as choosing means of countering harmful information should underlie the operation of systems for intelligent analytical processing of digital network content. The main purpose of such systems should be the detection, analysis and counteraction of harmful information.

Systems for intelligent analytical processing of network information objects can perform many different functions and consist of many different components. In particular, the components of distributed scanning of information objects, as well as their classification and categorization in accordance with the categories (or types) of harmful information established by law, are mandatory. However, the uncertainty of information available in information objects leads to a significant decrease in the efficiency of these components. Therefore, the component of eliminating the ambiguity of the semantic content of information objects, as well as the component for choosing countermeasures, should also be included among the basic components of such systems. In this regard, the purpose of this work is to clarify the functionality of the uncertainty elimination component and the countermeasures selection component, determine their interrelationships in the analysis of information flows, and develop models, algorithms and methods for selecting measures to counter harmful information based on an assessment of the semantic content of information objects under uncertainty.

The research was firstly presented at International Conference on INnovations in Intelligent SysTems and Applications (INISTA) 2020 [1]. In this paper we detailed and extended the description of countermeasure selection techniques and provided listing of the countermeasure selection algorithm. Besides we have added the second computational experiment for eliminating incompleteness and inconsistency of source data while in the research provided at INISTA 2020 we demonstrated only the first computational experiment that allows eliminating fuzziness.

The paper is organized as follows. Section 2 reviews the related works on selection of countermeasures against harmful information considering uncertainty of observation data. Section 3 provides a general algorithm for uncertainty elimination. Section 4 discusses the methods, models, techniques and algorithms for selection of countermeasures against harmful information. Section 5 describes two computational experiments and obtained results. Section 6 gives conclusions and future research directions.

2. Related Work

Despite the fact that in recent years some solutions on individual components of this kind of protection systems [2-20] have been suggested, they are either at the initial stage of development and implementation, or do not implement the full range of expected capabilities. So, in [2-7] the mechanisms for detecting and counteracting harmful information in network information objects are considered. These papers set out solutions for determining reliable estimates of digital network content. The mechanisms considered in them are based on methods of information classification, methods of intelligent data processing and spam filtering. However, these mechanisms are not focused on working in conditions of semantic uncertainty of information content.
The works [8-10] consider various methods of analyzing social networks to detect and select measures to counter harmful information. In [8], algorithms for searching by event description, identifying users of various networks, and searching by user groups are often used to detect harmful information. Methods for quantitative and qualitative assessment of information impacts in social networks, based on tabular and graphical tools for representing metrics and calculating metrics, are discussed in [9]. In [10] approaches to determine the demographic characteristics of users of social networks are considered. However, since there are other sources of unwanted information in addition to social networks, these approaches cannot be considered universal.

Many works suggest using traffic analysis methods based on the classification of web pages to detect and counteract unwanted information. Thus, in [11], methods based on content analysis of the internal properties of web pages are considered. In [12, 13], it is proposed to use a binary classifier based on identifying groups of internal properties of HTML documents, which is used to train systems for classifying web pages. Methods for training classifiers in order to detect and counter malicious information based on a combination of significant functions of web pages are discussed in [14]. However, all the methods presented in [11-14] are focused on the analysis of web content. Thus, they also cannot cover all sources of harmful information and types of countermeasures.

In some works, it is proposed to implement methods for detecting and countering harmful information based on the results of evaluating the semantic content of information objects using algorithms for classifying web content topics. For example, the papers [15, 16] describe an approach to searching for harmful information based on URL addresses. The advantage of this approach is to reduce the many characteristics of malicious information that need to be analyzed, which entails a reduction in the range of countermeasures. Another popular approach is to use it to analyze links in web content. In [17, 18], based on this approach, a procedure for hierarchical classification of web content is proposed. However, the application of these methods is limited. An interesting method proposed in [19] is the search and extraction of meaningful text from tags with the subsequent application of the classifier to the obtained samples. The same approach in combination with methods of counteracting harmful information is mentioned in [20]. But their application takes a significant amount of time.

Taking into account real conditions in the process of identifying and countering harmful information requires the use of modern approaches, in which the processing and assessment of the properties of harmful information is carried out under conditions of uncertainty. In some works, these approaches are based on methods, models and algorithms for eliminating uncertainty. In [21-24], approaches are considered in which the processing of uncertain information of various types, as well as decision support, are

implemented using artificial neural networks. In [25, 26], it is offered to use fuzzy sets for these purposes. Neural fuzzy networks are proposed to be used for obsolescence of uncertainty in [27]. However, it should be noted that the application of these methods to detect and counteract malicious information is a rather difficult task. On the other hand, a great advantage of these methods is that they allow you to choose measures to counter harmful information based on an assessment of the semantic content of information objects in conditions of uncertainty. These approaches will form the basis of the solutions considered in this paper.

Thus, the analysis of known approaches, methods and solutions for the detection and counteraction of harmful information shows that reliable control of the semantic content of information objects is a complex process that requires the combined use of various mechanisms. However, the task of developing existing methods for choosing measures to counter harmful information is still relevant. Methods for detecting and countering harmful information should be focused on processing poorly formalized (incomplete, inconsistent and fuzzy) data. It is necessary to use additional expert opinions and dynamic (changing) knowledge. The solutions discussed below are focused on the implementation of such approaches.

3. General Algorithm for Eliminating Uncertainty

Information objects are natural language text, media content, embedded parts of other information objects, executable scripts, domain names, IP addresses, etc. Solving the problem of assessing and categorizing the semantic content of information objects is an important stage for detecting harmful information, making decisions and countering harmful information. Elimination of ambiguity (namely, fuzziness, incompleteness and inconsistency) when solving this problem is a necessary condition for its successful solution in real subject areas, when the initial data are exposed to various factors of uncertainty. Such factors, for example, include noise when measuring, modeling, or observing the attributes of the semantic content of information objects. These attributes can be textual, graphic, numeric, logical, ordinal, nominal, etc. Among the various types of uncertainty, the most significant are ambiguity (fuzziness) and insufficiency (incompleteness, inconsistency) of the initial data.

Uncertainty inherent in the initial data of the tasks of detecting, evaluating and making decisions on counteracting harmful information can arise due to the non-stationarity of the information flow, fuzziness, incompleteness and inconsistency in identifying features of harmful information, the dynamics of the security system, the impact of destabilizing (often antagonistic) environmental factors, and also due to the presence of ambiguity of goals and the inconsistency of the tasks of detecting and countering harmful information.

The common algorithm for eliminating the uncertainty of the initial data for the problem of identifying harmful properties of information will be called the algorithm for eliminating uncertainty for assessing and categorizing the semantic content of information objects based on methods of processing fuzzy, incomplete and contradictory knowledge. This algorithm includes the following steps:

1. Input undefined harmful information's features and type of uncertainty (fuzziness or incompleteness and inconsistency).

2. If harmful information's features are fuzzy specified go to step 3 (steps 3-7 allow the experts to specify weight matrices of harmful information's features in advance), otherwise go to step 10.
3. Specification of the fuzzy harmful information's features system and initial membership functions of fuzzy sets.
4. Matching of the expert opinions on adding of the specific information objects' semantic content features to the harmful information's features set.
5. If there is one expert (i=1), go to 9 else go to 6.
6. Specification of the membership functions of fuzzy sets by the next expert (i=i+1).
7. Calculation of the common experts' opinion on adding of the specific information objects' semantic content features to the harmful information's features set (disjunctive sum).
8. If there is the next expert, go to 6 else 9.
9. Final choice of the specific information objects' semantic content features for the harmful information's features set (based on the max of preference function).
10. Generation of weight matrices W_m and W_{Y} for two-layer artificial neural network.
11. Activation of artificial neural network's input layer.
12. Initial setting of neurons of artificial neural network's output layer.
13. Bringing of input layer neurons to the state of second layer neurons.
14. Calculation of states of output layer neurons.
15. If artificial neural network is stable go to step 16 else go to step 13.
16. Sum values of weight coefficients.
17. Final choice of the specific information objects' semantic content features for the harmful information's features set (based on the max value of elements of sum output weight coefficients' vector).
18. Output the results: final harmful information's features considering uncertainty elimination in scope of fuzzy and conflicting knowledge processing.

The algorithm for eliminating uncertainty in the assessment and categorization of the semantic content of information objects is based on the use of the mathematical theory of fuzzy sets [25, 26]. The central link of the algorithm is a decision support mechanism for adding the analyzed fuzzy characteristics of information in digital network content to the set of features of malicious information (steps 3–9 of the algorithm). The criterion for the harmfulness of the processed semantic content in its assessment and classification is the excess of the features of dubious information characteristics in the semantic content of information objects of the threshold value (α-level of the preference function). The analyzed attributes of the semantic content include the presence, quantity, or nomenclature (severity) of questionable informational characteristics. The identification of the fuzzy character of harmful information is implemented based on the opinions of experts. To determine the subjective measure of confidence that this information belongs to a fuzzy set of characteristics of malicious information, membership functions are used. To combine several subjective confidence measures (that is, the opinions of several experts), mathematical operations of addition, union, intersection and disjunctive sum of fuzzy sets are used.

The algorithm also implements the elimination of uncertainty in the assessment and categorization of the semantic content of information objects using artificial neural networks. The neural network mechanism [21-24] for searching and predicting the

relationship between the features of the semantic content of information objects is specified in steps 10-17 of the algorithm. The purpose of this mechanism is reasonable adding of the analyzed incomplete and inconsistency features of information within digital network content to harmful information's features set. The principle of operation of this mechanism is that if there is at least one function that is guaranteed to be included in the set of functions of malicious information, it is possible to generate an input set of functions that takes into account incomplete and incompatible relationships of all functions. After that, an artificial neural network is put into operation. With its help, it is possible to obtain at the output a set of features of malicious information with coefficients (elements) characterizing the weight (severity) of features of malicious information. These coefficients make it possible to evaluate and classify dubious information as harmful.

The output result of the algorithm is a given system of features of the semantic content of information objects, which uniquely determines whether or not a particular information is harmful. Thus, the developed algorithm makes it possible, in conditions of uncertainty, to identify harmful information, as well as to form the initial data for making a decision on counteracting harmful information.

4. Models, Algorithm and Technique for Selecting Countermeasures

This section describes the developed technique for counteracting against harmful information and the related models and algorithm.

The technique is based on the decision-making theory, including multi criteria optimization methods. Input of the technique is as follows: (1) harmful objects; (2) available countermeasures. The main stages of the technique are: (1) generating models of information objects, information system, countermeasures and counteracting process; (2) selecting countermeasures. The output of the technique is the set of selected countermeasures.

We specified the following models considering subjects and objects participating in countermeasure selection process: information system model, information object model, countermeasure model, model of counteracting against harmful information.

An information system *IS*, where counteracting is implemented, is Internet. It incorporates objects *IO* and communication means *IC*. Thus, information system model is specified as follows: $IS = (IO, IC)$. The objects can be physical or informational. The developed model is limited with information objects. Communication is interaction between two or more objects and/or subjects. Communication between the information objects can be determined based on the existing references. In the developed model a subject can be represented as some information object, for example, a profile in social network or on some web site (in this case communication is determined based on the profile's friends and groups), or as a property of information object, for example a counter of visitors for the web page. Communication can be physical or logical. The developed model is limited with logical communications represented as follows: $IC \subseteq IO \times IO$.

We specify information object *IO* as follows:

$$IO = <size, role, hltype, type, state, ioaud, saud>,$$

where

size – a size of information object, it can take values {*s*, *m*, *l*}, where *s* – small, *m* – medium, *l* – large;

role – a role of information object, it can take values {*s*, *r*, *u*}, where *s* – sender, *r* – recipient, *u* – user;

hltype – a type of information object, it can take values {*h*, *n*}, where *h* – harmful objects, *n* – not harmful objects;

type – a more detailed type of information object, it can take values {*ter*, *hea*, *por*, *dru*, *cru*, *none*}, where *ter* – an information object containing public calls for terrorism and extremism; *hea* – the information objects containing information harmful for people's (especially children's) health, and moral and spiritual development; *por* – an information object with pornography propaganda; *dru* – the information objects containing information on ways of development, production and use of drugs and committing suicide, as well as swearing; *cru* – an information object containing direct calls for violence and cruelty (e.g. war), ethnic and religious hatred, or hostility in the content information; *none* – not applicable (if *hltype* is *n*);

state – a state of compromise of information object in case of harmful information impact, it can take values {*compr*, *nonc*}, where *compr* – object is compromised by the harmful information, *nonc* – object is not compromised;

ioaud – audience of the information object that is an array of links on information objects that are linked with the sender by communications and that are recipients of the objects (can be null);

saud – real number (if there is a counter of visitors of information object) or expert assessment of subjects who are the recipients of the object (can be 0).

Information objects can be classified into small objects IO_s, medium objects IO_m and large objects IO_l depending on their size. We consider post, message, comment, media object, etc. as IO_s, web page, group, channel, etc. as IO_m and information system, web site, social network, messenger, etc. as IO_l. These classes are related as follows: $IO_s \subset IO_m \subset IO_l$.

Each information object has a role. The roles are as follows: sender (subset *Rs* of *IO*), recipient (subset *Rr* of *IO*), and user (subset *Ru* of *IO*). *Rs* propagate harmful (or not) information. *Ru* incorporates all information objects that are connected with the considered information objects via *IC* and form the audience *Au* of information object (*ioaud*). Some users represented with information objects are the recipients of information object: $Rr \in Ru$, $Rr \in Au$. A subset *Rr* can be empty. Other users that form the rest part of the *Au* get information object unintentionally.

Information object can be harmful or not. Harmful object if its role is sender (namely *role*=*s* and *hltype*=*h*) affects the audience as follows:

1. Audience compromise state becomes *compr* (this is relevant for *ioaud* and *saud*).
2. Harmful information propagation, i.e. the part of the audience $Rr \in ioaud$ becomes senders *Rs*. While some information objects counted by *saud* can also become senders, it is difficult to trace if they have propagated harmful information.

Thus, harmful object affects information system state as follows: $\{IO^{k_i}, IC^{k_i}\}$ becomes $\{IO^{k_j}, IC^{k_j}\}$, where k_i – previous harmful object number, k_j – current harmful

object number. For *m* known information objects from IO^{kj} (*m*=[0;*M*], where *M* – number of elements in *ioaud* of this harmful object) the following parameters are changed: *role* becomes *s*, *hltype* becomes *h*, *type* becomes *ter*, *hea*, *por*, *dru* or *cru* (depending on considered harmful object type), and *state* becomes *compr*. It should be noticed that number of compromised information objects are $saud_{\Sigma} = |ioaud|_{k_i} + saud_{k_i} + |ioaud|_{k_j} + saud_{k_j}$, while $saud_{k_j} = \sum_{m} saud$.

The countermeasures should eliminate impacts on the audience and stop harmful information propagation. The countermeasures can be taken against:

1. Information object with sender role. Such measures should be taken if an audience of the harmful object is huge and information object is contrary to the laws of the country. In this case the following countermeasures can be taken: removal (or block); a warning.
2. Information objects from the audience of harmful information object. Such measures can be taken in case of low popularity of the sender to stop harmful information propagation or prevent access to harmful information. In this case the countermeasure of informing type can be taken. Thus, to stop harmful information propagation a warning about an illegality of content and responsibility for its distribution can be provided; to prevent access to harmful information a warning that content is for the appropriate age category can be provided.

We specify countermeasure *rm* from *RM* (set of countermeasures) as follows:

$$rm=<rm\_class, rm\_type, rm\_cost, rm\_ef>,$$

where

rm_class – class of countermeasure (barrier, disguise, informing or enforcement);
rm_type – size of the information object (small, medium, or large);
rm_cost – countermeasure cost;
*rm_rol*e – role of information object (sender or recipient);
rm_ef – efficiency of the countermeasure;
rm_cd – collateral damage from the countermeasure implementation.

Countermeasure cost is represented by the weight that depends on the countermeasure intrinsic cost, cost of implementation and maintenance, considering complexity of implementation and maintenance and required resources.

Efficiency of the countermeasure is represented as weight that depends on the ratio of recipients that won't be compromised in case of countermeasure implementation to the common number of recipients that would be compromised otherwise.

Collateral damage is represented as weight that depends on additional losses in case of countermeasure implementation, for example financial losses in case of web site blocking.

The countermeasure model is used to specify counteracting model. The countermeasure affects information system state: {*IO*, *IC*} become $\{IO^l, IC^l\}$, where *l* – countermeasure number. For *j* information objects from IO^l (*j*=[0;*N*], where *N* –number of elements in *ioaud* of the harmful object that were affected by the countermeasure), an information object is deleted or its following parameters are modified: *role* become *r* or *u*, *hltype* become *n*, *type* become *none*, and *state become nonc*. For *d* connections from IC^l (*d*=[0;*D*], where *D* – number of links between harmful object and connected objects

that were affected by the countermeasure), an information connection is deleted. Besides, $saud_{\Sigma}$ decreased.

The specified models are used to formalize the algorithm of counteracting against harmful information. We set the following requirements to the counteraction algorithm: (1) it should consider size of harmful information; (2) it should consider harmful information audience (size and age); (3) it should select the countermeasures that provide maximum efficiency and have minimum cost.

In scope of the counteraction algorithm development the input data for countermeasure selection (that are the output data of the previous stages) are as follows: size of information object (*size* parameter of object's model), role of information object (*role* parameter of object's model), high level type of information object (*hltype* parameter of object's model), and detailed type of information object (*type* parameter of object's model). Thus, we have information to satisfy the first requirement.

To satisfy the rest two requirements to the counteraction algorithm some additional information is required. Thus, to consider harmful information's audience (information objects that are linked with the sender *ioaud* and not linked recipients of the object *saud*) additional harmful information propagation algorithm is developed. It is based on search of linked objects and changing of their *state* to *compromised*. Size and age of the harmful information's audience is calculated considering these compromised objects and their traffic (using counters).

Countermeasure efficiency (that is specified in the countermeasure model as *rm_ef*) is calculated as ratio of recipients that won't be compromised in case of countermeasure implementation, both number of information objects from *ioaud* with *state compr* and *saud*, to the common number of recipients that would be compromised otherwise. While countermeasure cost (*rm_cost* in the countermeasure model) is specified by the experts manually. Besides, in counteraction algorithm class of countermeasure (that is selected depending on the harmful information type) and size of the information object are considered. The pseudocode of the counteraction algorithm is provided below:

```
1. Input io from IO where state=compr, hltype=h, role=s.
2. Input io class, io type.
3. Calculate direct ioaud for io.
4. Determine saud size, saud age for io.
5. Calculate propagated ioaud for io.
6. Determine propagated saud size, saud age for io.
7. Determine cms for io considering role, class, type, propagated
ioaud, propagated saud size, propagated saud age, rm_class,
rm_type.
8. For each countermeasure c from cms:
8.1.     Determine rm_cost.
8.2.     Determine rm_ef.
9. Select scms from cms with min rm_cost and max rm_ef.
10. Output scms.
```

Step 7 of the algorithm above is implemented based on the set of rules. The following classes of countermeasures can be outlined:

1. Barrier, namely, filtering of information objects and blocking of information sources using software. This measure can be implemented by the information object that has sender role (e.g. filtering of messages on the web site) and recipient role (e.g. parental control software, filtering options within operation system) as well.

2. Disguise (or distraction) can be implemented on the part of sender by adding distracting content, e.g. message or picture.
3. Informing, should be implemented on the sender part to motivate the recipients to avoid information object. For example, it can be implemented using warning message about illegality of the content, or age category of content.
4. Enforcement are the measures implemented as the result of laws, such as deleting of information or user blocking that can be implemented on the sender side, or web-site blocking that can be implemented on the domain management level.

On step 7 of the algorithm the rule-based technique for countermeasures list determination is used. It outputs the set of possible countermeasures *cms* considering *role* of information object (sender *s*, recipient *r*, or user *u*), *size* of information object (small *s*, medium *m*, large *l*), *type* of information object (public calls for terrorism and extremism *ter*, information harmful for people's health *hea*, pornography propaganda *por*, information on ways of development, production and use of drugs and committing suicide *dru*, direct calls for violence and cruelty *cru*, or *none*), and total audience size *ioaud* and *saud* and age, and *rm_class* (barrier, disguise, informing or enforcement) and *rm_type* (*size* of the information object *small*, *medium*, or *large*). Examples of rules for step 7:

- "if *role* = *s* and *size* = *s* and *type* = *ter* and total audience size<3000 and age>18 select countermeasures where *rm_class=disguise* or *informing* and *rm_type=small*";
- "if *role* = *u* and *size* = *s* and *type* = *hea* and total audience size<3000 and age>18 select countermeasures where *rm_class=barrier* or *informing* and *rm_type=small*".

If more than one countermeasure *c* is selected, on steps 8-9 of the proposed algorithm the multicriteria optimization is used.

5. Experiments and Discussion

In order to test the feasibility of implementing the proposed algorithms for eliminating uncertainties and choosing countermeasures, a computational experiment was carried out to refine the features of malicious information based on the mathematics of fuzzy sets. Below we consider the operation of a branch of this algorithm, which uses methods for processing fuzzy knowledge (namely, calculating a disjunctive sum).

There is initial set of fuzzy specified harmful information's features. The expert opinions are specified. That is initial membership functions of fuzzy sets that characterize preliminary, fuzzy specified harmful information set, for example:

$$\tilde{\tilde{X}} = [\Delta\tilde{\tilde{x}}_{chil}|\mu(\Delta\bar{x}_{\text{chil}});\ \Delta\tilde{\tilde{x}}_{\text{terr}}|\mu(\Delta\bar{x}_{\text{terr}});\Delta\tilde{\tilde{x}}_{\text{porn}}|\mu(\Delta\bar{x}_{\text{porn}});\ \Delta\tilde{\tilde{x}}_{\text{drug}}|\mu(\Delta\bar{x}_{\text{drug}});\ \Delta\tilde{\tilde{x}}_{\text{war}}|\mu(\Delta\bar{x}_{\text{war}})]^{\text{T}}, \quad (1)$$

where

$\Delta\bar{x}_{\text{chil}}(k)$ – abnormal deviation of the average amount of information harmful for people's (especially children's) health, and moral and spiritual development;

$\Delta\bar{x}_{\text{terr}}(k)$ – abnormal deviation of the average amount of information containing public calls for terrorism and extremism in traffic;

$\Delta\bar{x}_{\text{porn}}(k)$ – abnormal deviation of the average amount of information with pornography propaganda;

$\Delta\bar{x}_{\text{drug}}(k)$ – abnormal deviation of the average amount of information containing data on ways of development, production and use of drugs and committing suicide, as well as swearing; and

$\Delta\bar{x}_{\text{war}}(k)$ – abnormal deviation of the average amount of direct calls for violence and cruelty, ethnic and religious hatred, or hostility in the content information;

μ– membership function of fuzzy set, can take values from 0 to 1.

The disjunctive sum of two fuzzy sets $\widetilde{A}$ and $\widetilde{B}$ characterizing the opinions of the first and second experts, accordingly, is specified using unions and intersections as follows:

$$\tilde{A} \oplus \tilde{B} = (\tilde{A} \cap \bar{\tilde{B}}) \cup (\bar{\tilde{A}} \cap \tilde{B}), \tag{2}$$

where $\bar{\tilde{A}}$ and $\bar{\tilde{B}}$– complements of these fuzzy sets.

The membership function for *j*-th harmful information's feature looks as follows:

$$\forall x_j \in \overline{1,\ldots,5:} \quad \mu_{\tilde{A}\oplus\tilde{B}}(x_j) = \max\{[\min\{\mu_{\tilde{A}}(x_j),\ 1-\mu_{\tilde{B}}(x_j)\}];$$
$$[\min\{1-\mu_{\tilde{A}}(x_j),\ \mu_{\tilde{B}}(x_j)\}]\}.$$

Opinion of the first expert (A) about the assessment and categorization of each feature from the listed above (1) as harmful information's feature can be represented as fuzzy set:

$$\tilde{A} = \{\Delta\tilde{\bar{x}}_{chil}|0{,}3;\ \Delta\tilde{\bar{x}}_{\text{terr}}|0{,}1; \Delta\tilde{\bar{x}}_{porn}|0{,}1;\ \Delta\tilde{\bar{x}}_{drug}|0{,}5;\ \Delta\tilde{\bar{x}}_{war}|0{,}2\}.$$

Opinion of the second expert (B) about the assessment and categorization of each feature from the listed above as harmful information's feature can be represented as similar fuzzy set:

$$\tilde{B} = \{\Delta\tilde{\bar{x}}_{chil}|0{,}7; \Delta\tilde{\bar{x}}_{\text{terr}}|0{,}9; \Delta\tilde{\bar{x}}_{porn}|0{,}4; \Delta\tilde{\bar{x}}_{drug}|0{,}5;\ \Delta\tilde{\bar{x}}_{war}|0{,}4\}.$$

The complements of these fuzzy sets are as follows:

$$\bar{\tilde{A}} = \{\Delta\tilde{\bar{x}}_{chil}|0{,}7; \Delta\tilde{\bar{x}}_{\text{terr}}|0{,}9; \Delta\tilde{\bar{x}}_{porn}|0{,}9; \Delta\tilde{\bar{x}}_{drag}|0{,}5; \Delta\tilde{\bar{x}}_{war}|0{,}8\};$$

$$\bar{\tilde{B}} = \{\Delta\tilde{\bar{x}}_{chil}|0{,}3;\ \Delta\tilde{\bar{x}}_{\text{terr}}|0{,}1; \Delta\tilde{\bar{x}}_{porn}|0{,}6; \Delta\tilde{\bar{x}}_{drug}|0{,}5; \Delta\tilde{\bar{x}}_{war}|0{,}6\}.$$

Intersections of these fuzzy sets are as follows:

$$\tilde{A} \cap \bar{\tilde{B}} = \{\Delta\tilde{\bar{x}}_{chil}|0{,}3;\ \Delta\tilde{\bar{x}}_{\text{terr}}|0{,}1; \Delta\tilde{\bar{x}}_{porn}|0{,}1;\ \Delta\tilde{\bar{x}}_{drug}|0{,}5;\ \Delta\tilde{\bar{x}}_{war}|0{,}2\};$$

$$\bar{\tilde{A}} \cap \tilde{B} = \{\Delta\tilde{\bar{x}}_{chil}|0{,}7;\ \Delta\tilde{\bar{x}}_{\text{terr}}|0{,}9; \Delta\tilde{\bar{x}}_{porn}|0{,}4;\ \Delta\tilde{\bar{x}}_{drug}|0{,}5;\ \Delta\tilde{\bar{x}}_{war}|0{,}4\}.$$

Finally, a union of these fuzzy sets will give the results of disjunctive summation. These results characterize aggregated opinion of two experts about the assessment and categorization of each feature from the listed above as harmful information's feature:

$$\tilde{A} \oplus \tilde{B} = (\tilde{A} \cap \bar{\tilde{B}}) \cup (\bar{\tilde{A}} \cap \tilde{B}) =$$

$$= \{\Delta\tilde{\bar{x}}_{chil}|0{,}7;\ \Delta\tilde{\bar{x}}_{\text{terr}}|0{,}9; \Delta\tilde{\bar{x}}_{porn}|0{,}4; \Delta\tilde{\bar{x}}_{drug}|0{,}5; \Delta\tilde{\bar{x}}_{war}|0{,}4\}.$$

If there are more than two experts, an opinion of the third expert is specified. The aggregated opinion of two previous experts is used as one opinion and all cycle is repeated till there are experts. As the result we get aggregated opinion of experts based on fuzzy knowledge processing.

Let us introduce threshold value of membership function describing a preference of adding the information object' semantic content features to the set of harmful information's features as $\mu^{\text{тр}} \geq 0{,}6$.

Further, for the purpose of the final selection of the features of the semantic content of information objects and their inclusion in the set of characteristics of malicious information, the maximum preference function is used.

The membership function value chart describing a criterion for assessment and categorization in fuzzy conditions is represented in Fig. 1. The results of experimental computations show that the fuzzy sets math allows eliminating this type of input data uncertainty while assessing and categorizing information objects' semantic content using the fuzzy knowledge processing methods.

The state of values of membership functions (for the considered example) should be interpreted as a forecast of the guaranteed preference of adding the specific content feature to the set of harmful information's features.

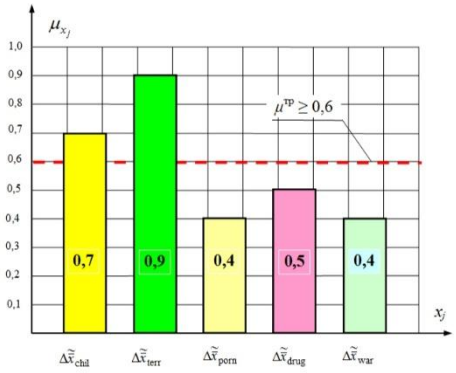


Fig. 1. The results of computational experiment

This state (see Fig. 1) for the *k*-th step of analytical processing of digital network content and for the considered example is characterized by the low preference of adding the following features to the harmful information set:

$\Delta\bar{x}_{\text{porn}}(k)$ – abnormal deviation of the average amount of information with pornography propaganda;

$\Delta\bar{x}_{\text{drug}}(k)$ – abnormal deviation of the average amount of information containing data on ways of development, production and use of drugs and committing suicide, as well as swearing; and

$\Delta\bar{x}_{\text{war}}(k)$ – abnormal deviation of the average amount of direct calls for violence and cruelty, ethnic and religious hatred, or hostility in the content information.

These directions (pornography, drugs and war propaganda) in the content do not exceed the threshold now and they are not harmful. Severity preference is given to the following features as to the most harmful (for the considered example):

$\Delta\bar{x}_{\text{chil}}(k)$ – abnormal deviation of the average amount of information harmful for people's (especially children's) health, and moral and spiritual development;

$\Delta\bar{x}_{\text{terr}}(k)$ – abnormal deviation of the average amount of information containing public calls for terrorism and extremism in traffic.

These are an abnormal deviation of the average amount of information harmful for children's health and average amount of information containing public calls for terrorism. The countermeasures should be implemented against these threats. The calculations were implemented for sample data. They characterize weight of specific feature in the tasks of harmful information detection and counteraction.

Another example of implementation of the proposed algorithms (for uncertainty elimination and countermeasure selection) is the second computational experiment on specification of harmful information features on the basis of mathematical algorithms of artificial neural networks theory.

Let us to consider an example of operation of the second branch of common algorithm for uncertainty elimination. This branch operates using methods of processing of incomplete and conflicting data using artificial neural networks [21-24].

In scope of implementation of this branch of the common algorithm we use the neural network based mathematical procedure for elimination of incompleteness and inconsistency of assessment and categorization of information objects' semantic content features. Two-layer artificial neural network is the basis of the second branch of uncertainty elimination algorithm (steps 10-17 of the algorithm described in Section 3). This branch is developed to search and forecast interconnections between the information objects' semantic content features. As a result, it allows taking the reasonable decision on including (or not) the analyzed incompletely or contradictory specified features of information circulating in digital web content into the set of features of harmful information.

Mathematical essence of the neural network-based branch of the common algorithm for uncertainty elimination (steps 10-17 of the algorithm described in Section 3) is as follows. We determine at least one feature guaranteed to be included in the set of features of harmful information, first. We construct input feature vector $\{\vec{Y}^{i}_{\text{inp}}\}$ using two-layer artificial neural network. This vector $\{\vec{Y}^{i}_{\text{inp}}\}$ considers incomplete and conflicting interconnections of all features (based on opinion of E experts). We get output harmful information's feature vector with coefficients (elements) characterizing weight (severity) of these features based on the results of solving the problem of neural network transformation (steps 10-17 of the common algorithm). The results of these computations allow assessing and categorizing information as harmful on step 18 of the common algorithm (considering incompleteness and inconsistency of input data).

The proposed model for selecting "important" (sensitive) harmful information's features in the conditions of incompleteness and inconsistency allows filtering the subjective values and obtain knowledge empirically based on experts' opinion.

Let empirical data have the form of a protocol:

$\{\vec{Y}^{i}_{\text{inp}}, \quad i = 1, \ \dots \ , \ E\}$,

where vector $\vec{Y}^{i}_{\text{inp}} = (Y^{i}_{\text{inp }1}, Y^{i}_{\text{inp }2}, \ldots, Y^{i}_{\text{inp }P})$ is vector of input features (in terms of artificial neural networks it is input vector *A*) that considers incomplete and conflicting interconnections of all $j = 1, \ldots, P$ harmful information's features according to the opinion of *i*-th expert from the set *E* of experts.

The vector characterizing "importance", for example, for each of 5 (five) previously considered harmful information's features can be common illustrative example:

$\vec{Y}^{1}_{\text{inp}} = (1, 0, 0, 1, -1)$.

This vector is character representation of the expression: "According to the opinion of the first expert "importance" of the harmful information's features is as follows: the first feature $Y_{\text{inp }1}$ (its physical meaning is $\Delta\bar{x}_{\text{chil}}$) and the fourth feature $Y_{\text{inp }4}$ (its physical meaning is $\Delta\bar{x}_{\text{drug}}$) are "important" (sensitive/valuable), the fifth feature $Y_{\text{inp }5}$ (its physical meaning is $\Delta\bar{x}_{\text{war}}$) is not "important" (not sensitive), for the rest features of harmful information (the second and the third) $Y_{\text{inp }2}$ (its physical meaning is $\Delta\bar{x}_{\text{terr}}$) and $Y_{\text{inp }3}$ (its physical meaning is $\Delta\bar{x}_{\text{porn}}(k)$) an opinion of the first expert is absent (it is equal to 0)".

For our computational experiment, assume that at a given time the feature $Y_{\text{inp }5}$ (the unit element of the input vector *A*) is guaranteed "important" (valuable/sensitive) feature of harmful information. This feature characterizes $\Delta\bar{x}_{\text{war}}$ – abnormal deviation of the average amount of direct calls for violence and cruelty, ethnic and religious hatred, or hostility in the content information. Other features of harmful information are undetermined. To get reasonable results of semantic content assessment and detect harmful information it is required to reconstruct the rest components of vector of "important" (valuable/sensitive) harmful information's features. In process of operation the two-layer artificial neural network reconstructs the rest components of vector *A*. Let us to consider this process with an example.

Suppose we are interested in the components of the vector characterizing "importance" of all harmful information's features considering that the fifth feature is obligatory for inclusion in the list of "dangerous" features, i.e. $Y_{\text{inp }5}$ value characterizing "importance" of this feature is equal to "1". Let us to pre-normalize the increments of all features relative to the scale of the activation function. Let the activation function have a stepwise form:

$$f(Y_{\text{inp}}) = \begin{cases} 1, & Y_{\text{inp}} \geq 1; \\ 0, & 0 \leq Y_{\text{inp}} < 1 \\ -1, & Y_{\text{inp}} < 0. \end{cases}.$$

Then $Y_{\text{inp }5}$ value characterizing "importance" of harmful information's feature $\Delta\bar{x}_{\text{war}}$ will correspond to the output value of fifth neuron that is equal to 1, while input vector will take the form $A = (0, 0, 0, 0, 1)$. In other words, the two-layer artificial neural network takes as input $Y_{\text{inp}} = (0, 0, 0, 0, 1)$.

Then, considering mathematical essence of the second neural network based branch of the common algorithm for uncertainty elimination (steps 10-17 of the algorithm described in Section 3), the output vector $B = (b_1, b_2, b_3, b_4, b_5)$ of the two-layer artificial neural network consequentially takes the following values:

$B(0) = f([0; 0; 0; 0; 1]) = [0, 0, 0, 0, 1]$;

$B(1) = f([0{,}667; -0{,}333; 1; 1; 0]) = [0, -1, 1, 1, 1]$;

$B(2) = f([3; -0{,}667; 4; 4; 7]) = [1, -1, 1, 1, 1]$;

$B(3) = f([3; -1{,}667; 4{,}667; 4{,}333; 7{,}667]) = [1, -1, 1, 1, 1]$;

$B(4) = f([3; -1{,}667; 4{,}667; 4{,}333; 7{,}667]) = [1, -1, 1, 1, 1]$;

$B(5) = f([3; -1{,}667; 4{,}667; 4{,}333; 7{,}667]) = [1, -1, 1, 1, 1]$.

The obtained results characterize intermediate and final dependencies of harmful information's features weight ("importance"/value/severity), i.e. characterize total preference (from the experts' point of view) of including of these features, that should be assessed in scope of detection and counteraction against harmful information, into the set of dangerous features. These results can be represented graphically as a diagram (Fig. 2).

As it can be seen from the diagram (Fig.2) the two-layer artificial neural network designed in the interests of evaluating the semantic content for the search and detection of harmful information, has stabilized after the third tact (step). Thus, using such artificial neural network containing two layers of neurons it is possible to implement assessment and short term normative weight ("importance"/value/severity) forecasting for harmful information's features in the conditions of incompleteness and inconsistency of input data.

The results of solving of the second computational experiment (example) allow constructing the vector of sensitive for the given conditions harmful information's features with a high degree of objectivity using accumulated in the neural network data. They allow selecting the volume and nomenclature of harmful information's features for including into the set of dangerous features mathematically correct and as objectively as possible. Moreover, the set of dangerous, obvious features constructed in the interests of detecting and counteracting harmful information, will be guaranteed to include such "important" (sensitive) features as $Y_{\text{inp }1}$ (its physical meaning is $\Delta\bar{x}_{\text{chil}}$), $Y_{\text{inp }3}$ (its physical meaning is $\Delta\bar{x}_{\text{porn}}(k)$), $Y_{\text{inp }4}$ (its physical meaning is $\Delta\bar{x}_{\text{drug}}$) and $Y_{\text{inp }5}$ (its physical meaning is $\Delta\bar{x}_{\text{war}}$) and won't include the feature $Y_{\text{inp }2}$ (its physical meaning is $\Delta\bar{x}_{\text{terr}}$).

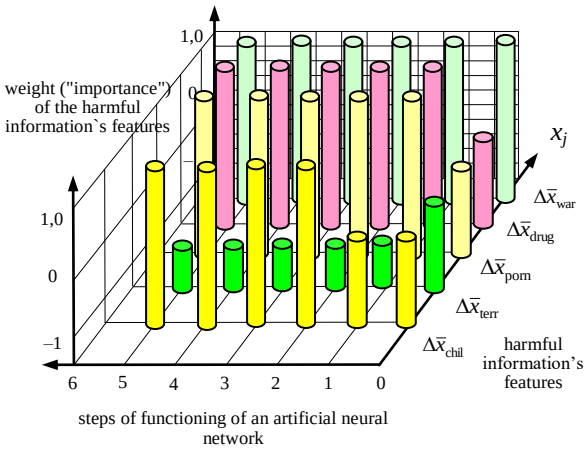


Fig. 2. Graph of the dependence of the weight ("importance") of the harmful information`s features on the cycle (step) of calculation new states of neurons of the output layer

Thus, the second computational experiment was considered. This experiment is based on the second neural network based branch of the common algorithm for uncertainty elimination (steps 10-17 of the algorithm described in Section 3). The experiment demonstrated that this algorithm allows eliminating incompleteness and inconsistency of source data. This distinguishes it from the first branch of the common algorithm for uncertainty elimination (steps 3-9 of the algorithm described in Section 3) that is considered in the first computational experiment and allows eliminating fuzziness.

The results of the computational experiments demonstrate that application of both branches of the common algorithm (described in Section 3) allows eliminating uncertainty of any type while constructing the set of dangerous explicit features for decision making in order to detect and counteract against harmful information.

6. Conclusion

The paper proposed a novel approach to developing the methodological foundations for harmful information's features assessment and decision making on counteracting against harmful information propagation considering uncertainty in data observations. These tasks were specified, and two variants of implementation of harmful information's countermeasures selection process were introduced. The stages of common algorithm for uncertainty elimination while assessing and categorizing information objects' semantic content using methods for fuzzy, incomplete and conflicting knowledge processing are specified for determination of input data for harmful information counteracting task. Thus, the common scheme of the process of eliminating uncertainty in semantic content of information objects and the selection of countermeasures against harmful information were described within the framework of

the common architecture of the system for intelligent analytical processing of network content.

We developed the models, algorithm and technique for harmful information's countermeasures selection on the basis of the proposed scheme for uncertainty elimination. The techniques include the information system model, harmful information counteracting model (including countermeasure model) and the harmful information counteracting algorithm. For the countermeasure selection we use traditional decision support theory methods and multicriteria optimization methods.

On the basis of the proposed scheme for eliminating uncertainty, models, an algorithm and a technique for selecting means of countering harmful information were developed. These solutions include an information system model, a harmful information counteracting model (including a countermeasure model), and a harmful information counteracting algorithm. To select countermeasures, traditional methods of decision support theory and methods of multicriteria optimization are used.

The future research will be devoted to enhancement of the developed algorithms and tools for harmful information's countermeasures selection. It is planned to make the proposed algorithms more universal, so that they would allow evaluating the characteristics of harmful information and choosing countermeasures taking into account both non-stochastic and probabilistic uncertainties.

Acknowledgment. The reported study was partially funded by RFBR project 18-29-22034 mk and by the budget project 0073-2019-0002.

References

1. Parashchuk, I., Doynikova, E., Saenko, I., Kotenko, I.: Selection of Countermeasures against Harmful Information based on the Assessment of Semantic Content of Information Objects in the Conditions of Uncertainty. In Proceeding of the 2020 International Conference on Innovations in Intelligent Systems and Applications (INISTA 2020), Novi Sad, Serbia, 1-7. (2020)
2. Vaismoradi, M., Turunen, H., Bondas, T.: Content Analysis and Thematic Analysis: Implications for Conducting a Qualitative Descriptive Study. Nursing & Health Sciences, Vol. 15, No. 3, 398-405. (2013)
3. Elo, S., Kyngas, H.: The Qualitative Content Analysis Process. Journal of Advanced Nursing, Vol. 62, No. 1, 107-115. (2008)
4. Krippendorff, K.: Content Analysis: An Introduction to Its Methodology. Sage Publications, California, USA. (2004)
5. Graneheim, U. H., Lundman, B.: Qualitative Content Analysis in Nursing Research: Concepts, Procedures and Measures to Achieve Trustworthiness. Nurse Education Today, Vol. 24, No. 2, 105-112. (2004)
6. Pashakhanlou, H.: Fully Integrated Content Analysis in International Relations. International Relations, Vol. 31, No. 4, 447-465. (2017)
7. Timmermans, S., Iddo, T.: Theory Construction in Qualitative Research: From Grounded Theory to Abductive Analysis. Sociological Theory, Vol. 30, No. 3, 167-186. (2012)
8. Marcus, S., Moy, M., Coffman, T.: Social Network Analysis. In: Cook, D. J., Holder, L. B. (eds.): Mining Graph Data. John Wisley & Sons, Hoboken. (2007)
9. UCINET documentation (2017). [Online]. Available: https://sites.google.com/site/ucinetsoftware/document (current February 2021)

10. Scott, J.: Social Network Analysis: Developments, Advances, and Prospects. Social Network Analysis and Mining, Vol. 1, No. 1, 21-26. (2011)
11. Qi, X., Davison, B. D.: Web Page Classification: Features and Algorithms. ACM Computing Surveys, Vol. 41, No. 2, 12:1–12:31. (2009)
12. Patil, A., Pawar, B.: Automated Classification of Web Sites using Naive Bayessian Algorithm. In Proceedings of the International Multi-Conference of Engineers and Computer Scientists, Vol. 1, Hong Kong, 466-467. (2012)
13. Kotenko, I., Chechulin, A., Shorov, A., Komashinsky, D.: Analysis and Evaluation of Web Pages Classification Techniques for Inappropriate Content Blocking. In: Perner, P. (ed.): Proceedings of the 14th Industrial Conference on Data Mining, Lecture Notes in Artificial Intelligence, Vol. 8557, 39-54. (2014)
14. Shibu, S., Vishwakarma, A., Bhargava, N.: A Combination Approach for Web Page Classification using Page Rank and Feature Selection Technique. International Journal of Computer Theory and Engineering, Vol. 2, No. 6, 897-900. (2010)
15. Kan, M.-Y., Thi, H. O. N.: Fast Web Page Classification using URL Features. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 325-326. (2005)
16. Baykan, E., Henzinger, M., Marian, L., Beber, I.: Purely URL-Based Topic Classification. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 1109-1110. (2009)
17. Dumais, S., Chen, H.: Hierarchical Classification of Web Content. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 256-263. (2000)
18. Calado, P., Cristo, M., Moura, E., Ziviani, N., Ribeiro-Neto, B., Goncalves, M. A.: Combining Link-Based and Content-Based Methods for Web Document Classification. In Proceedings of the 12th International Conference on Information and Knowledge Management, New York, New Orleans, LA, USA, 394-401. (2003)
19. Belmouhcine, A., Idrissi, A., Benkhalifa, M.: Web Classification Approach using Reduced Vector Representation Model Based on HTML Tags. Journal of Theoretical and Applied Information Technology, Vol. 55, No. 1, 137-148. (2013)
20. Kotenko, I., Chechulin, A., Komashinsky, D.: Categorisation of Web Pages for Protection against Inappropriate Content in the Internet. International Journal of Internet Protocol Technology, Vol. 10, No. 1, 61-71. (2017)
21. Kriesel, D.: A Brief Introduction to Neural Networks. Cambridge University Press, Grate Britain. (2010)
22. Mehlig, B.: Artificial Neural Networks. University of Gothenburg, Sweden. (2019)
23. Rojas, R.: Neural Networks. Springer-Verlag, Germany. (1996)
24. Parashchuk, I.: System Formation Algorithm of Communication Network Quality Factors using Artificial Neural Networks. In Proceedings of the 1st IEEE International Conference on Circuits and System for Communications, Saint-Petersburg, Russia, 263-266. (2002)
25. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Englewood Cliffs, NJ, USA. (1992)
26. Kotenko, I., Saenko, I., Ageev, S., Kopchak, Y.: Abnormal Traffic Detection in Networks of the Internet of Things Based on Fuzzy Logical Inference. In Proceedings of the XVIII International Conference on Soft Computing and Measurements, Saint-Petersburg, Russia, 5-8. (2015)
27. Kotenko, I., Parashchuk, I., Omar, T.: Neuro-Fuzzy Models in Tasks of Intelligent Data Processing for Detection and Counteraction of Inappropriate, Dubious and Harmful Information. In Proceedings of the 2nd International Scientific-Practical Conference Fuzzy Technologies in the Industry, Ulyanovsk, Russia, 116-125. (2018)

Igor Kotenko obtained the Ph.D. degree in 1990 and the National degree of Doctor of Engineering Science in 1999. He is Professor of computer science and Head of the

Laboratory of Computer Security Problems of St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). He was a project leader in the research projects from the US Air Force research department, via its EOARD (European Office of Aerospace Research and Development) branch, EU FP7 and FP6 Projects, HP, Intel, F-Secure, etc. His research results were tested and implemented in more than fifty Russian research and development projects. His main research interests are innovative methods for network intrusion detection, simulation of network attacks, vulnerability assessment, verification and validation of security policy, etc. He has chaired several International conferences and workshops, and serves as editor on multiple editorial boards.

Igor Saenko obtained the Ph.D. degree in 1992 and the National degree of Doctor of Engineering Science in 2002. He is Professor of computer science and Leading Researcher of the Laboratory of Computer Security Problems of St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). His main research interests are security policy management, access control, management of virtual computer networks, knowledge modeling soft and evolutionary computation, information and telecommunication systems.

Elena Doynikova obtained her PhD in St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS) in 2017. In 2015 she was awarded the medal of the Russian Academy of Science in area of computer science, computer engineering and automation. Currently she is a senior researcher of computer security problems laboratory, SPC RAS. Research interests: information systems security, risk analysis and security decision support methods, security metrics, data mining. She participated in many projects devoted to information systems security research.

Igor Parashchuk, Doctor of Engineering Sciences, Professor; Leading Researcher of the Laboratory of Computer Security Problems in St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), areas of scientific interest: computer network security, automated information systems, data storage and processing.

Received: March 14, 2021; Accepted: August 31, 2021.

Computer Science and Information Systems 19(1):435–453 https://doi.org/10.2298/CSIS210314062D

An Effective Method for Determining Consensus in Large Collectives ⋆

Dai Tho Dang[1,2**], Thanh Ngo Nguyen[3], and Dosam Hwang[1⋆⋆]

[1] Department of Computer Engineering,
Yeungnam University, Gyeongsan 38541, Republic of Korea
daithodang@ynu.ac.kr, dshwang@yu.ac.kr
[2] Vietnam - Korea University of Information and Communication Technology,
The University of Danang, Danang, Vietnam
ddtho@vku.udn.vn
[3] Department of Applied Informatics, Faculty of Computer Science and Management,
Wrocław University of Science and Technology, 50-370 Wrocław, Poland
thanh-ngo.nguyen@pwr.edu.pl

Abstract. Nowadays, using the consensus of collectives for solving problems plays an essential role in our lives. The rapid development of information technology has facilitated the collection of distributed knowledge from autonomous sources to find solutions to problems. Consequently, the size of collectives has increased rapidly. Determining consensus for a large collective is very time-consuming and expensive. Thus, this study proposes a vertical partition method (VPM) to find consensus in large collectives. In the VPM, the primary collective is first vertically partitioned into small parts. Then, a consensus-based algorithm is used to determine the consensus for each smaller part. Finally, the consensus of the collective is determined based on the consensuses of the smaller parts. The study demonstrates, both theoretically and experimentally, that the computational complexity of the VPM is lower than 57.1% that of the basic consensus method (BCM). This ratio reduces quickly if the number of smaller parts reduces.

Keywords: large collective, consensus, algorithm, computational complexity.

1. Introduction

Rapid development in information technology has facilitated the use of distributed knowledge from autonomous sources to find solutions to problems [1]. One such example is social networks. Social media platforms, such as Twitter, Facebook, Instagram, and Wikipedia, have revolutionized communication among individuals, groups, and organizations. Exploiting the data generated from social network sites is helpful for both individuals and organizations, such as businesses for marketing, sales, customer support, and public relations. One example of knowledge created by collectives of users is Wikipedia.

⋆ This is an extended version of the article titled "A New Approach to Determine 2-Optimality Consensus for Collectives". In: Fujita H., Fournier-Viger P., Ali M., Sasaki J. (eds) Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices. IEA/AIE 2020. Lecture Notes in Computer Science, vol 12144, Springer.

⋆⋆ Corresponding authors

It is currently the most extensive online encyclopedia collection, with over 54 million articles available in more than 312 languages. Data from social media are considered sources of knowledge [2], and organizations and individuals are increasingly looking for ways to benefit from the collective intelligence of these sources [3]. Another example is Internet of Things (IoT). It has given rise to large amounts of continuous data collected from the physical world [4], [5]. IoT has pervasively penetrated most areas of human life, such as homes, cities, industry, organizations, agriculture, hospitals, and healthcare [6], [7], [8]. Its applications collect data for their aims, such as decision making, system performance boosting, optimal management of resources [9]. This leads to the continuous growth of collectives [10].

The rapid development of other fields has also contributed to the increase in the size of collectives; one such field is biology, where technological advances have allowed researchers to gather unprecedented amounts of data. The amount of biological data is rapidly increasing. Over the last decade, the amount of produced data has doubled almost every seven months [11]. Advances in computational sciences and communication technologies have allowed biologists to share data [12].

Consensus determination has a significant role in computer science, automatic control, social sciences, and biology [13], [14], [15], [16]. Consensus determination is based on collective members' knowledge states. However, the knowledge states in a collective are often inconsistent; thus, consensus determination is complex [17]. The Consensus method is an efficient tool to solve this problem [18].

Consensus determination is an NP-hard problem [16], [18], [19], and many heuristic algorithms have been used to find consensus for different knowledge structures [18], [20]. The complexity of most such algorithms is $O(n^2)$ or larger [16], [18], [19]. For large collectives, determining consensus is very time-consuming and expensive. This study considers determining consensus for large collectives.

This study is an expanded version of our earlier conference paper [21]. In that paper, we proposed an algorithm for determining the 2-Optimality consensus for a large binary collective, the vertical partition method (VPM). First, this method vertically divides the collective into many small parts. Second, it uses a brute-force algorithm to determine the optimal consensuses of these parts. Finally, these consensuses are used to determine the consensus of the whole collective. The approach reduces the time complexity of the brute-force algorithm, and the optimal consensuses of the smaller parts can be used to find consensus in a collective. An experiment showed that the VPM is 99.94% and 99.89% faster if we vertically partition the collective into three and two parts. However, this was only a case study with a binary collective and brute-force algorithm. The two most fundamental problems of the VPM have not been solved. The first is the computing of the computational complexity of the VPM. The second is proving the efficiency of the VPM for determining consensus in large collectives in general. In this study, we deal with these two problems. The contributions of this study are as follows:

- We propose the VPM and develop a general mathematical model for the VPM.
- The computational complexity of the VPM is computed as a function of the collective sizes of the smaller parts.
- We prove that the computational complexity of the VPM is lower than 57.1% that of the BCM. This ratio reduces quickly if the number of smaller parts reduces.
- The efficiency of the VPM was measured experimentally through a case study.

The remainder of this paper is organized as the following. We present some related concepts of this study in Section 2. In Section 3, the VPM is described. The computational complexity of the VPM is calculated in Section 4. The capability of the VPM is demonstrated in Section 5. In Section 6, we investigate the efficiency of the VPM through a case study. Finally, conclusions and future work are shown in Section 7.

2. Related works

Nowadays, collective intelligence is attracting researchers from many fields, such as biology [13], computer science [22], and automatic control [23].

In computer science, the consensus problem has been investigated in distributed computing [13], multi-agent systems [25], [26], IoT [27], etc. In recent years, collective intelligence has become a promising research area, attracting increasing interest from researchers and organizations. Axiomatic, optimization, and constructive methods have been used to address the consensus problem.

The axiomatic method was first proposed by K. Arrow under seven conditions [27]. It employs simple structures, such as partial order linear order. Nguyen introduced a set of ten postulates for consensus choice functions [17]. However, no consensus choice functions satisfy all postulates concurrently. The postulates 1-Optimality and 2-Optimality have an important role because if one consensus satisfies one of these two postulates, it will satisfy most of the others.

The constructive method solves consensus problems based on the structure of elements and the relation between elements. The relation between elements may be a distance function or preference relation between elements. Many structures of elements have been investigated, such as n-tree [13], ordered partitions [20], disjunction and conjunction Structures [29], binary vectors [30], and ontology [31], [32]

The optimization approach defines consensus choice functions, which are usually based on optimality rules. Optimality rules include the global optimality rule, Condorcet's optimality rule, and maximal similarity rules [18].

Let *U* denote a finite set of objects that represent all potential knowledge states of the same subject. Symbol 2^U denotes the powerset of *U*, which includes the set of all subsets of *U*. Let $\prod_k(U)$ be a set of all k-element subsets of set *U* for $k \in \mathcal{N}$ (where $\mathcal{N}$ is the set of natural numbers), and let

$$\prod(U) = \bigcup_{k\in\mathcal{N}} \prod_k(U)$$

A set $X \in \prod(U)$ is called a collective. The macrostructure of the set *U* is a distance function $d : U \times U \rightarrow [0, 1]$ that satisfies the nonnegative, reflexive, and symmetrical conditions. Pair *(U,d)* is called the distance space [18].

For a given collective $X \in \prod(U)$, the consensus of *X* is found by:

- Postulate 1-Optimality if: $d(x^*, X) = min_{y\in U} d(y, X)$
- Postulate 2-Optimality if: $d^2(x^*, X) = min_{y\in U} d^2(y, X)$

where x^* is the consensus of X, $d(x^*, X)$ is the sum of the distances from x^* to collective members, $d^2(x^*, X)$ is the sum of the squared distances from x^* to collective members.

The postulates 1-Optimality and 2-Optimality have an important role in finding consensus. Determining consensuses that meet one of the two postulates are often NP-hard problems [16], [18], [19]. For example, the Kemeny ranking is an NP-hard problem, even for only four votes [14], [33]. Heuristic algorithms have been applied for this task. Over 104 algorithms and combinations have been introduced [14], and their complexities are often $O(m^2)$ or larger.

Consensus determination of large collectives is widespread in medicine and bioinformatics. Many consensus problems must be solved in these two fields, such as gene prediction, protein structure prediction, and disease-related gene ranking. One example is the consensus ranking. A large collective of gene lists of regulation, expression, correlation, interaction can be extracted from data mining results, such as disease-related genes and protein-protein interactions, and disease-related genes. Thus, it is important to rank such data. Given m rankings of n elements, the complexities of the algorithms are $O(n^3m)$, $O(mn + n^2)$, and $O(n^2m)$ [34]. The second example is determining consensus for DNA structure. In [35], algorithms were introduced to determine the 2-Optimality consensus for this structure. The last example is the multiple structure alignment problem. The complexity of the best algorithm to solve this problem is $O(n^2k^2)$, where k is the maximum length of n proteins [36].

For group decision making (GDM) problems, many consensus algorithms have been proposed for various knowledge structures. Many algorithms have been introduced for hesitant fuzzy linguistic structures. In [37], the authors proposed a new method for measuring the difference between two hesitant fuzzy linguistic term sets. Based on this measure, an algorithm was proposed to resolve the hesitant linguistic GDM problem's consensus problem. This algorithm obtains optimally adjusted individual opinions in hesitant linguistic GDM. Its computational complexity is $O(mn^2)$, where n is the number of experts, and m is the number of alternatives to be assessed. In [38], Wu and Xu first defined a new consistency measure. A new algorithm was then presented to improve the consistency index for a given hesitant fuzzy linguistic preference relation. It has a computational complexity of $O(mn^2)$. In [39], the concept of a possibility distribution was introduced. The authors proposed some aggregation operators, such as the hesitant fuzzy linguistic weighted average operator and the hesitant fuzzy linguistic ordered weighted average operator, based on the possibility distributions. A consensus measure was then defined, and a consensus reaching process was presented. The complexity of this algorithm is $O(n^2)$.

The consensus problem has also been of interest in economic [40], [41], [42]. Algorithms for investment strategy design for a multiagent system that supports investment decisions on the stock market were presented in [41]. Based on decisions generated by agents, the supervisor agent uses a consensus method to generate a satisfactory rate of return and reduce the level of risk associated with investing in a financial instrument. The complexity of this algorithm is $O(nm^2)$, where n is the size of the set of decisions and m is the number of decision elements.

3. Vertical Partition Method (VPM)

The basic consensus method (BCM) directly determines consensus based on the primary collective X [15]. In other words, it determines consensus based on the knowledge states

of all members in the collective X. If the collective size is large, the VPM is often very time-consuming and expensive.

Instead of using the algorithm to determine the consensus based on the collective X as the BCM, the VPM applies the algorithm for smaller parts of the collective X to reduce the computational complexity. First, the primary collective is vertically partitioned into small parts. Then, a consensus-based algorithm is applied to determine consensus for each smaller part. Finally, the consensus of the collective X is determined based on the consensuses of the smaller parts. The procedure of the VPM is illustrated in Fig. 1.

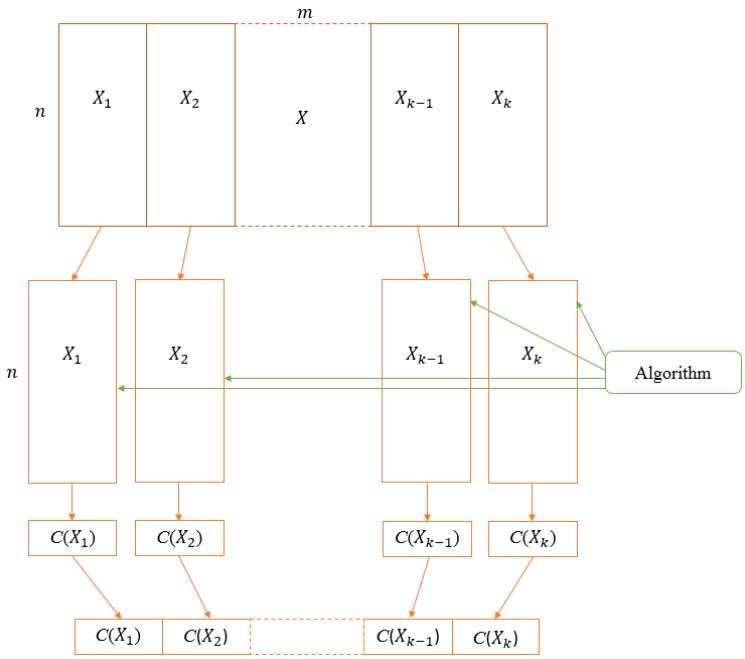


Fig. 1. Schema of the VPM.

Let a large collective X contain n members, where the length of each member is m. The VPM with k parts to determine consensus for the collective X is described as follows:

– **Step 1**: Use the vertical partition to divide the collective X into k disjointed parts $X_1, X_2, ..., X_k$ that satisfy the following:

$$U_1 \cup U_2 \cup \ldots \cup U_k = X$$

$$U_1 \cap U_2 \cap \ldots \cap U_k = \emptyset$$

$$|length(X_i) - length(X_j)| = 1 \text{ or } |length(X_i) - length(X_j)| = 0$$

for $1 \leq i, j \leq k$.

- **Step 2**: Determine consensuses for $X_1, X_2, \ldots, X_k$ as $C(X_1)$, $C(X_2), \ldots, C(X_k)$, respectively.
- **Step 3**: Determine consensus $C(X)$ by combining $C(X_1), C(X_2), ..., C(X_k)$ sequentially:

$$C(X) = C(X_1)C(X_2)...C(X_k)$$

Note that the number of smaller parts k is a natural number that satisfies:

$$2 \leq k \leq \lfloor \frac{m}{2} \rfloor \tag{1}$$

Under this condition, the VPM is very general and flexible.

4. Computational Complexity of the VPM

Let $CVPM(m, m_1, m_2, ...m_k)$ represent the computational complexity of the VPM, where $m, m_1, m_2, \ldots, m_k$ are the lengths of $X, X_1, X_2, \ldots, X_k$, respectively. We can calculate $CVPM(m, m_1, m_2, ...m_k)$ based on the computational complexity of the steps.

Let $O(g(m))$ represent the computational complexity of partitioning the collective X into smaller parts, $O(f(l))$ represent the computational complexity of determining consensus for a smaller part with length l, and $O(h(m))$ represent the computational complexity of generating consensus for the collective X by combining the consensuses of parts $X_1, X_2, \ldots, X_k$. The computation of $CVPM(m, m_1, m_2, ...m_k)$ is detailed as follows:

- In step 1, the collective X with the length of m is vertically partitioned into k smaller parts $X_1, X_2, \ldots, X_k$. The computational complexity of this task is $O(g(m))$.
- In step 2, the complexity of finding the consensuses of k smaller parts X_i $(i = \overline{1,k})$ is computed as the following:

$$O(f(m_1)) + O(f(m_2)) + ... + O(f(m_k))$$

The difference between the lengths of members of any two smaller parts is not larger than 1. The length of the smaller parts X_i $(i = \overline{1,k})$ are $\lfloor \frac{m}{k} \rfloor$ or $\lfloor \frac{m}{k} \rfloor + 1$. The number of smaller parts with the length $\lfloor \frac{m}{k} \rfloor$ is $k - (m - k \times \lfloor \frac{m}{k} \rfloor)$, and the number of smaller parts with the length $\lfloor \frac{m}{k} \rfloor + 1$ is $m - k \times \lfloor \frac{m}{k} \rfloor$. We have

$$O(f(m_1)) + O(f(m_2)) + ... + O(f(m_k))$$

$$= (k - (m - k \times \lfloor \frac{m}{k} \rfloor) \times O(f(\lfloor \frac{m}{k} \rfloor)) + (m - k \times \lfloor \frac{m}{k} \rfloor \times O(f(\lfloor \frac{m}{k} \rfloor) + 1)$$

- In step 3, the complexity of generating consensus for the collective X by combining the consensuses of $X_1, X_2, ..., X_k$ is $O(h(m))$.

Thus

$$CVPM(m, m_1, m_2, ...m_k) = O(g(m)) + (k - (m - k \times \lfloor \frac{m}{k} \rfloor) \times O(f(\lfloor \frac{m}{k} \rfloor))$$

$$+(m - k \times \lfloor \frac{m}{k} \rfloor \times O(f(\lfloor \frac{m}{k} \rfloor) + 1) + O(h(m))$$

$O(g(m)) = O(m)$ and $O(h(m)) = O(m)$ are linear functions; thus, in the case of large collective, do not consider them:

$$CVPM(m, m_1, m_2, ...m_k) = (k - (m - k \times \lfloor \frac{m}{k} \rfloor) \times O(f(\lfloor \frac{m}{k} \rfloor))$$

$$+(m - k \times \lfloor \frac{m}{k} \rfloor \times O(f(\lfloor \frac{m}{k} \rfloor) + 1) \tag{2}$$

5. Efficiency of the VPM

The efficiency of the VPM is measured by comparing its computational complexity with that of the BCM. Denoting $p = \lfloor \frac{m}{k} \rfloor$, we have $m = kp + r$ $(0 \leq r < k)$ where r is the remainder in the division of m by k. Thus, $X_1, X_2, \ldots, X_k$ include:

- $(k - r)$ parts have p columns;
- r parts have $(p + 1)$ columns.

We have

$$CVPM(m, m_1, m_2, ...m_k) = (k - r) \times O(f(p)) + r \times O(f(p + 1)) \tag{3}$$

Because $2 \leq k \leq \lfloor \frac{m}{2} \rfloor$ (from (1)) and $p = \lfloor \frac{m}{k} \rfloor$, we have

$$p \geq 2 \tag{4}$$

The BCM directly calculates consensus based on all knowledge states of X. By $CBCM(m)$ we denote the computational complexity of the BCM. We have

$$CBCM(m) = O(f(m)) \tag{5}$$

Theorem 1. *If the computational complexity of the BCM is $O(m^2)$, we have*

$$CBCM > 1.75 \times CVPM$$

Proof.

The algorithm determining consensus has quadratic computational complexity.

From (4), we have

$$CVPM = (k - r)p^2 + r(p + 1)^2$$

$$CVPM = kp^2 + 2pr + 1 \tag{6}$$

From (5), we get

$$CBCM = m^2 = (kp + r)^2$$

$$CBCM = k^2p^2 + 2kpr + r^2 \tag{7}$$

From (6) and (7), we have

$$\frac{CBCM}{CVPM} = \frac{k^2p^2 + 2kpr + r^2}{kp^2 + 2pr + 1} = \frac{k(kp^2 + 2pr + 1) - (k - r^2)}{kp^2 + 2pr + 1}$$

$$= \frac{k(kp^2 + 2pr + 1)}{kp^2 + 2pr + 1} - \frac{k - r^2}{kp^2 + 2pr + 1}$$

$$= k - \frac{k - r^2}{kp^2 + 2pr + 1} > k - \frac{k}{kp^2 + 2pr + 1} > k - \frac{k}{kp^2} = k - \frac{1}{p^2}$$

Thus

$$\frac{CBCM}{CVPM} > k - \frac{k}{p^2} \tag{8}$$

From (1) and (4), we have $k \geq 2$ and $p \geq 2$. From (8), we get

$$\frac{CBCM}{CVPM} > k - \frac{1}{p^2} \geq 2 - \frac{1}{2^2} = 1.75$$

Or

$$CBCM > 1.75 \times CPVM$$

From (8), we can see that $\frac{CBCM}{CVPM}$ increases quickly if k increases. It reaches $\frac{m}{2^{m-1}}$ when $k = \lfloor \frac{m}{2} \rfloor$.

Theorem 2. *If the computational complexity of the BCM is higher than $O(m^2)$, we have*

$$CBCM > 1.75 \times CVPM$$

Proof.

Let us consider the case that the computational complexity of the BCM is (m^3).

From (3), we have

$$CVPM = (k - r)p^3 + r(p + 1)^3 \tag{9}$$

From (5), we have

$$CBCM = m^3 \tag{10}$$

From Theorem 1, we have

$$1.75 \times ((k - r)p^2 + r(p + 1)^2) < m^2 \tag{11}$$

Multiply both sides of (11) by m, we get

$$1.75 \times ((k - r)p^2 + r(p + 1)^2)m < m^3$$

Or

$$1.75 \times ((k-r)p^2m + r(p+1)^2m) < m^3 \tag{12}$$

Let us consider the left-hand side of the inequality (12). Because $m = kp + r$ and $k > r \geq 0$, we have $m \geq kp$.

Thus

$$1.75 \times ((k-r)p^2m + r(p+1)^2m) \geq 1.75 \times ((k-r)p^2(kp) + r(p+1)^2(kp)) \tag{13}$$

Because $k \geq 2$ and $p \geq 2$ (from (1) and (4)), then $kp > p + 1$. We have

$$1.75 \times ((k-r)p^2(kp) + r(p+1)^2(kp)) = 1.75 \times (k(k-r)p^3 + r(p+1)^2(kp))$$

$$\gg 1.75 \times ((k-r)p^3 + r(p+1)^3) \tag{14}$$

From (13) and (14), we get

$$1.75 \times ((k-r)p^2 + r(p+1)^2)m \gg 1.75 \times ((k-r)p^3 + r(p+1)^3) \tag{15}$$

From (12) and (15), we have

$$1.75 \times ((k-r)p^3 + r(p+1)^3) \ll m^3$$

Or

$$1.75 \times CVPM \ll CBCM$$

We proved that Theorem 2 is true if the computational complexity of the BCM is $O(m^3)$.

Assume that $1.75 \times CVPM \ll CBCM$ with the complexity of the BCM is $O(m^t)$ for $t > 3$. We have

$$CBCM = m^t \tag{16}$$

$$CPVM = (k-r)p^t + r(p+1)^t \tag{17}$$

$$1.75 \times ((k-r)p^t + r(p+1)^t) < m^t \tag{18}$$

We need to prove $1.75 \times CVPM \ll CBCM$ with the complexity of the BCM is $O(m^{t+1})$. In other words, we need prove that

$$1.75 \times ((k-r)p^{t+1} + r(p+1)^{t+1})m < m^{t+1} \tag{19}$$

Multiply both sides of (18) by m, we get

$$1.75 \times ((k-r)p^t + r(p+1)^t)m < m^{t+1}$$

Or

$$1.75 \times ((k-r)p^t m + r(p+1)^t m) < m^{t+1} \tag{20}$$

Let us consider the left-hand side of the inequality (20). Because $m = kp + r$ and $k > r \geq 0$, we have $m \geq kp$. Thus

$$1.75 \times ((k-r)p^t m + r(p+1)^t m) \geq 1.75 \times ((k-r)p^t(kp) + r(p+1)^t(kp)) \quad (21)$$

Because $k \geq 2$ and $p \geq 2$ (from (1) and (4)), we have $kp > p + 1$. We have

$$1.75 \times ((k-r)p^t(kp) + r(p+1)^t(kp)) = 1.75 \times (k(k-r)p^{t+1}) + r(p+1)^t(kp)$$
$$\gg 1.75 \times ((k-r)p^{t+1} + r(p+1)^{t+1}) \quad (22)$$

From (21) and (22), we obtain

$$1.75 \times ((k-r)p^t m + r(p+1)^t m) \gg 1.75 \times ((k-r)p^{t+1} + r(p+1)^{t+1}) \quad (23)$$

From (20) and (23), we have

$$1.75 \times ((k-r)p^{t+1} + r(p+1)^{t+1}) \ll m^{t+1}$$

It means that (19) was proved.

Theorem 3. *The computational complexity of the BCM is* $O(m^t n^w)$. *If* $t \geq 2$, *for any* $w \geq 0$, *we have*

$$CBCM > 1.75 \times CVPM$$

Proof.

From (3), we have

$$CVPM = (k-r)p^t n^w + r(p+1)^t n^w \quad (24)$$

From (5), we have

$$CBCM = m^t n^w \quad (25)$$

Thus

$$\frac{CBCM}{CVPM} = \frac{m^t n^w}{(k-r)p^t n^w + r(p+1)^t n^w}$$

$$= \frac{m^t n^w}{n^w \times ((k-r)p^t + r(p+1)^t)}$$

$$= \frac{m^t}{(k-r)p^t + r(p+1)^t}$$

From Theorem (1) and Theorem (2), we get

$$= \frac{m^t}{((k-r)p^t + r(p+1)^t)} > 1.75$$

Or

$$CBCM > 1.75 \times CPVM$$

6. Application of the PVM

This section examines the efficiency of the VPM through a case study. Determining the consensus for a binary collective is an NP-hard problem; applying the VPM can efficiently deal with this situation.

Set U is described as $U = \{u_1, u_2, ..., u_q\}$ where each element is a binary vector of length m. The size of U is 2^m. Each set $X \in \prod(U)$ is a collective that is represented as

$$X = \{x_1, x_2, ..., x_n\}$$

where each element x_i is a binary vector for $1 \leq i \leq n$. Each element $x_i \in X$ is represented as

$$x_i = (x_i^1, x_i^2, \ldots, x_i^m), x_i^j = \{0, 1\}, 1 \leq j \leq m.$$

The brute-force algorithm is used to find the optimal consensus for collectives containing binary vectors. This algorithm is unfeasible because its computational complexity is $O(n2^m)$. In this study, the VPM using the brute-force algorithm with two and three parts is investigated.

6.1. Algorithms

TwP algorithm
In this algorithm, the collective X is vertically partitioned into two parts: X_1 and X_2.

- X_1 has n vectors, the length of vectors is $\lfloor \frac{m}{2} \rfloor$.
- X_2 has n vectors, the length of vectors is $m - \lfloor \frac{m}{2} \rfloor$.

The brute-force algorithm is used to determine the 2-Optimality consensus for X_1 and X_2. Then, the 2-Optimality consensus of the collective X is determined. The TwP algorithm is represented as follows.

Algorithm 1. TwP

Input: Collective $X = \{x_1, x_2, ..., x_n\}$
Output: 2-Optimality consensus x^* of the collective X

BEGIN
1. Vertically partition the collective X into two parts: X_1, X_2;
2. $C(X_1) = \text{brute} - \text{force}(X_1)$;
3. $C(X_2) = \text{brute} - \text{force}(X_2)$;
4. $x^* = concat(C(X_1), C(X_2))$;

END

ThP algorithm
In the ThP algorithm, the collective X is vertically partitioned into three parts: X_1, X_2, and X_3. Note that the difference between the lengths of any two smaller parts is equal to 0 or 1. The brute-force algorithm is used todetermine the 2-Optimality consensus for X_1, X_2, and X_3. Finally, the 2-Optimality consensus of the collective X is determined.

This algorithm is presented as the followings.

Algorithm 2. ThP

Input: Collective $X = \{x_1, x_2, ..., x_n\}$

Output: 2-Optimality consensus x^* of the collective X

BEGIN

1. Vertically partition the collective X into two parts: X_1, X_2, X_3 ;
2. $C(X_1) = \text{brute} - \text{force}(X_1)$;
3. $C(X_2) = \text{brute} - \text{force}(X_2)$;
4. $C(X_3) = \text{brute} - \text{force}(X_3)$;
5. $x^* = concat(C(X_1), C(X_2), C(X_3))$;

END

6.2. Experiments and Evaluation

The TwP and ThP algorithms are the VPM using the brute-force algorithm. This section estimates the ability of the TwP and ThP algorithms by experiments. The two algorithms are examined both running time and consensus quality. We compare these two algorithms to the basic heuristic and brute-force algorithms. The reason is that the basic heuristic algorithm is the most common algorithm to find consensus for binary collectives, and the brute-force algorithm is used to develop the TwP and ThP algorithms.

The significant level α is chosen as 0.05. Consensus quality of a heuristic algorithms is calculated as follows:

$$CQ = 1 - \frac{|d^2(x^*, X) - d^2(x_{opt}, X)|}{d^2(x_{opt}, X)}$$

where x^* is the 2-Optimality consensus found by the heuristic algorithm, and x_{opt} the optimal consensus found by the brute-force algorithm.

Consensus quality

The following experiment aims to evaluate the consensus quality of the algorithms TwP and ThP. A dataset with 26 collectives is created randomly. Each collective includes 650 elements, and the element length is 22.

We run the basic heuristic, TwP, and ThP algorithms on the dataset. It generates three consensus quality samples of the basic heuristic, TwP, and ThP algorithms. The samples are represented in Table 1. In Fig.5., red, green, and black columns describes consensus quality for the TwP, ThP, and basic heuristic algorithms, respectively.

The boxplots of these consensus quality samples are described in Fig.6. The medians of the TwP, ThP, and basic heuristic algorithms' consensus quality are 0.99925, 0.99780, and 0.96590, respectively. The consensus quality sample of the basic heuristic algorithm has the lowest level of closeness with each other.

We need to determine the distribution of these samples. The null hypothesis H_0 for this test is that the consensus quality sample is normally distributed. The Shapiro-Wilk test is applied to find distributions of these samples. The *p-value* of the TwP algorithm's consensus quality sample is 0.0002. Because *p-value* $< \alpha$, H_0 is rejected. It indicates that the consensus quality sample of the TwP algorithm is not normally distributed.

The similarity, *p-values* of the consensus quality samples of the algorithms ThP and basic heuristic are less than the significant level (*p-value*=0.03077 and *p-value*=0.000002 for the consensus quality sample of the ThP and basic heuristic algorithms, respectively).

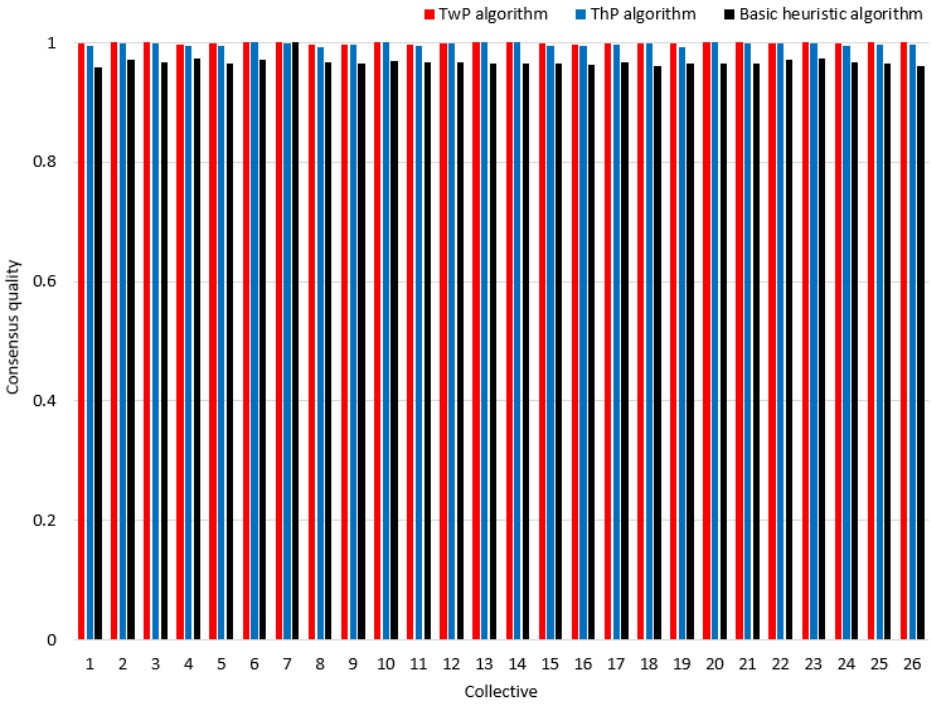


Fig. 2. Consensus quality of the algorithms TwP, ThP, and basic heuristic.

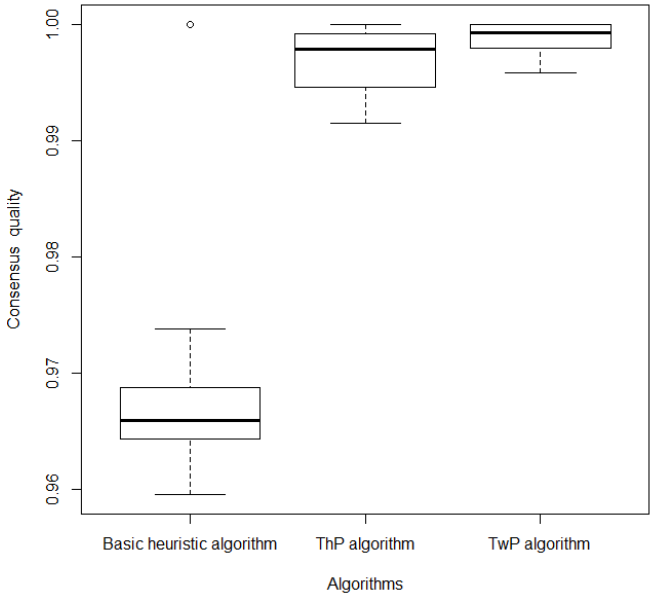


Fig. 3. The boxplots of consensus quality of the algorithms TwP, ThP, and basic heuristic.

Table 1. Consensus quality of the algorithms TwP, ThP, and basic heuristic.

Collective	TwP algorithm	ThP algorithm	Basic heuristic algorithm
1	0.9985	0.9945	0.9595
2	1.0000	0.9984	0.9718
3	1.0000	0.9981	0.9675
4	0.9968	0.9948	0.9738
5	0.9978	0.9936	0.9643
6	1.0000	1.0000	0.9716
7	1.0000	0.9993	1.0000
8	0.9958	0.9915	0.9672
9	0.9971	0.9966	0.9656
10	1.0000	1.0000	0.9687
11	0.9958	0.9935	0.9672
12	0.9996	0.9983	0.9674
13	1.0000	1.0000	0.9648
14	1.0000	1.0000	0.9643
15	0.9988	0.9946	0.9648
16	0.9960	0.9939	0.9624
17	0.9982	0.9975	0.9662
18	0.9989	0.9985	0.9616
19	0.9979	0.9926	0.9641
20	1.0000	1.0000	0.9652
21	1.0000	0.9992	0.9652
22	0.9985	0.9984	0.9715
23	1.0000	0.9986	0.9737
24	0.9981	0.9954	0.9666
25	1.0000	0.9958	0.9648
26	1.0000	0.9972	0.9611

It means that the consensus quality samples are not normally distributed. We compare these three consensus quality samples. The hypotheses are declared as follows:

- H_0: The medians of consensus quality of the algorithms TwP, ThP, and basic heuristic are equal.
- H_1: The medians of consensus quality of the algorithms TwP, ThP, and basic heuristic are not equal.

Because three samples do not come from the normal distribution, the Kruskal-Wallis test is applied to evaluate the hypotheses. We obtain *p-value*=2.7e-11. As *p-value*$<$0.05, H_0 is rejected. We can conclude that the medians of consensus quality of the TwP, ThP, and basic heuristic algorithms are not equal.

From the output of the Kruskal-Wallis test, we realize that there is a significant difference between samples. However, we do not know which pairs of samples are different. The Pairwise Wilcoxon test is used to calculate pairwise comparisons between samples with corrections for multiple testing. The *p-values* are shown for each pair in the output as follows:

– The *p-value* for the basic heuristic and ThP algorithms is 2.6e-08.

– The *p-value* for the basic heuristic and TwP algorithms is 1.4e-08.

– The *p-value* for the TwP and ThP algorithms is 0.024.

Since three *p-values* are less than 0.05, we can conclude that the difference in consensus quality between the basic heuristic algorithm and the ThP algorithm, between the

basic heuristic algorithm and the TwP algorithm, between the TwP algorithm and the ThP is statistically significant.

The consensus quality of the TwP algorithm is 0.1% higher than that of the ThP algorithm and 3.4% higher than that of the basic heuristic algorithm. The consensus quality of the TwP algorithm is 3.3% higher than that of the basic heuristic algorithm.

Running time

The brute-force algorithm determines consensus based on the knowledge states of all members in the collective. The brute-force is the BCM, and the algorithms TwP and ThP are the VPM. They are developed based on the brute-force algorithm. The following experiment aims to evaluate the running time of VPM by comparing the running time of the brute-force, TwP, and ThP.

A dataset containing 15 collectives is randomly created. The vector length is 22 and collective sizes are 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, and 1000. We perform the ThP, TwP, and brute-force algorithms on this dataset. Three running time samples of the three algorithms are generated. They are represented in Table 2.

Table 2. Running time of the algorithms brute-force, TwP, and ThP (seconds).

Collective size	Brute-force algorithm	TwP algorithm	ThP algorithm
300	96.810	0.111	0.029
350	107.038	0.129	0.034
400	123.702	0.148	0.038
450	138.886	0.165	0.044
500	154.490	0.181	0.046
550	170.055	0.201	0.048
600	184.172	0.221	0.054
650	199.352	0.234	0.062
700	218.713	0.254	0.069
750	233.024	0.271	0.073
800	248.737	0.297	0.078
850	264.513	0.311	0.081
900	282.291	0.325	0.086
950	294.624	0.344	0.093
1000	307.256	0.361	0.104

The Shapiro-Wilk test is applied to specify the distribution of the samples. Their *p-values* larger than α (*p-value*=0.601, *p-value* =0.7, *p-value*=0.739 for the running time sample of the algorithms brute-force, TwP, ThP, respectively). It means that these samples come from the normal distribution. The hypotheses to compare the running time of these algorithms are declared as follows:

- H_0: The means of running time of the algorithms brute-force, TwP, ThP are equal.
- H_1: The means of running time of the algorithms brute-force, TwP, ThP are not equal.

As the samples come from the normal distribution, we use the one-way ANOVA to evaluate the hypotheses. We get *p-value*=2e-16, it means that the means of running time of the brute-force, TwP, ThP algorithms are not equal.

This result indicates that some of the sample means are different. However, we do not know which pairs of samples are different. We use the Tukey HSD test for performing multiple pairwise-comparison between the means of samples. The *p-values* are shown for each pair in the output as follows:

– The *p-value* for the ThP algorithm and brute-force algorithms is 1e-12.

– The *p-value* for the TwP algorithm and brute-force algorithms is 1e-12.

– The *p-value* for the TwP algorithm and ThP algorithms is 0.99.

The difference in running time between the TwP algorithm and the ThP algorithm is not statistically significant. The difference in running time between the brute-force algorithm and others is statistically significant. The running time of the TwP, ThP algorithms are equal to 0.01%, 0.003% that of the brute-algorithm, respectively.

7. Discussion

The basic heuristic algorithm is popular to find consensus for collectives in the literature. The consensus quality of the algorithms TwP and ThP are 3.4% and 3.3% higher than that of the basic heuristic algorithm, respectively. Besides, the VPM proved its effectiveness in running time by experiments. The TwP and ThP algorithms' running time is hugely less than that of the brute-force algorithm if the collective is only partitioned into two and three parts. The running time continuously reduces if the number of smaller parts increases, satisfying (1). The VPM is an efficient tool to deal with large collectives.

8. Conclusions

In this study, we introduced the VPM to determine large collectives. We developed a general mathematical model for the VPM. The computational complexity of the VPM is computed as a function of the collective sizes of the smaller parts. We proved that the computational complexity of the VPM is lower than 57.1% that of the BCM. This ratio reduces quickly if the number of smaller parts reduces. Besides, The efficiency of the VPM was measured experimentally through experiments.

In the future, we will investigate combining the VPM and parallel processing to increase the efficiency of the VPM.

Acknowledgments. This work was supported by the 2021 Yeungnam University Research Grant.

References

1. Nguyen N.T., Szczerbicki E., Trawiński B., Nguyen V.D.: Collective Intelligence in Information Systems. Journal of Intelligent and Fuzzy Systems 37, No. 6, 7113–7115. (2019), https://doi.org/10.3233/JIFS-179324
2. Oxley A.: Security Risks in Social Media Technologies. Elsevier (2013).
3. Hansen D.L., Shneiderman B. et al.: Analyzing Social Media Networks with NodeXL. Elsevier Inc. (2020).
4. Amin F., Choi G.S.: Hotspots Analysis Using Cyber-physical-social System for a Smart City. IEEE Access, Vol. 8, 122197-122209. (2020), https://doi.org/10.1109/ACCESS.2020.3003030

5. Asghari P., Rahmani A.M., Javadi S.: Internet of Things Applications: A Systematic Review. Computer Networks, Vol. 148, 241–261. (2019).
6. Farooq M.S., Riaz S.et al.: A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. IEEE Access, Vol. 7, 156237–156271 (2019).
7. Verma P., Sood S.K.: Fog assisted-IoT Enabled Patient Health Monitoring in Smart Homes. IEEE Internet of Things Journal, Vol. 5, No. 3, 1789–1796 (2018).
8. Hassija V., Chamola V. et al.: A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. IEEE Access, Vol. 7, 82721–82743 (2019).
9. Sunhare P., Chowdhary R.R, Chattopadhyay M.K.: Internet of Things and Data Mining: An Application Oriented Survey. Journal of King Saud University - Computer and Information Sciences. (2020), https://doi.org/10.1016/j.jksuci.2020.07.002
10. Maleszka M., Nguyen N.T.: Integration Computing and Collective Intelligence. Expert Systems with Applications, Vol. 42, No. 1, 332–340. (2015), https://doi.org/10.1016/j.eswa.2014.07.036
11. Stephens Z.D., Lee S.Y. et al.: Big data: Astronomical or Genomical?. PLoS Biology, Vol. 13, No. 7, 1–11. (2015), https://doi.org/10.1371/journal.pbio.1002195
12. Yin Z., Lan H.: Computing Platforms for Big Biological Data Analytics: Perspectives and Challenges," Computational and Structural Biotechnology Journal, Vol. 15, 403–411. (2017).
13. Jansson J., Rajaby R., Shen C., Sung W.K.: Algorithms for the Majority Rule (+) Consensus Tree and the Frequency Difference Consensus Tree. IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 15, No. 1, 15–26. (2018).
14. Ali A., Meilă M.: Experiments with Kemeny ranking: What Works When?. Mathematical Social Sciences, Vol. 64, No. 1, 28–40, 2012, https://doi.org/10.1016/j.mathsocsci.2011.08.008
15. Dang D.T., Nguyen N.T., Hwang D.: Multi-Step Consensus: An Effective Approach for Determining Consensus in Large Collectives. Cybernetics and Systems, Vol. 50, No. 2, 208–229. (2019), https://doi.org/10.1080/01969722.2019.1565117
16. Badal P.S., Das A.: Efficient Algorithms Using Subiterative Convergence for Kemeny Ranking Problem. Computers and Operations Research, vol. 98, 198–210. (2018).
17. Nguyen N.T.: Processing Inconsistency of Knowledge in Determining Knowledge of a Collective. Cybernetics and Systems, Vol. 40, No.8, 670–688., (2009), https://doi.org/10.1080/01969720903294593
18. Nguyen N.T: Advanced Methods for Inconsistent Knowledge Management. London: Springer London. (2008).
19. D'Ambrosio A., Mazzeo G., Iorio C., Siciliano R.: A Differential Evolution Algorithm for Finding the Median Ranking Under the Kemeny Axiomatic Approach. Computers and Operations Research, Vol. 82, 126–138 (2017), https://doi.org/10.1016/j.cor.2017.01.017
20. Danilowicz C., Nguyen N.T.: Consensus-based Partitions in the Space of Ordered Partitions. Pattern Recognition, Vol. 21, No. 3, 269–273. (1988), https://doi.org/10.1016/0031-3203(88)90061-1
21. Dang D.T., Mazur Z., Hwang D. (2020) A New Approach to Determine 2-Optimality Consensus for Collectives. In: Fujita H., Fournier-Viger P., Ali M., Sasaki J. (eds) Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices. IEA/AIE 2020. Lecture Notes in Computer Science, Vol. 12144, 570-581. (2020), https://doi.org/10.1007/978-3-030-55789-8_49
22. Xiaohui C.: A study of Collective Intelligence in Multiagent Systems. University of Louisville, Kentucky, USA. (2004).
23. Meng ., Zhang H.T, Wang Z., Chen G.: Event-Triggered Control for Semiglobal Robust Consensus of a Class of Nonlinear Uncertain Multiagent Systems. IEEE Transactions on Automatic Control, Vol. 65, No. 4, 1683–1690. (2020), https://doi.org/10.1109/TAC.2019.2932752
24. Lynch N.A.: Distributed Algorithms. Morgan Kaufmann. (1996).
25. Sliwko L, Nguyen N.T.: Using Multi-agent Systems and Consensus Methods for Information Retrieval in Internet. International Journal of Intelligent Information and Database Systems, Vol. 1, No 2, 181-198. (2007), https://doi.org/10.1504/IJIIDS.2007.014949

26. Qin J., Ma Q., Shi Y., Wang L.: Recent Advances in Consensus of Multi-agent Systems: A Brief Survey. IEEE Transactions on Industrial Electronics, Vol. 64, No. 6, 4972–4983. (2017), https://doi.org/10.1109/TIE.2016.2636810
27. Li S., Oikonomou G. et al.: A Distributed Consensus Algorithm for Decision Making in Service-Oriented Internet of Things. IEEE Transactions on Industrial Informatics, Vol. 10, No. 2, 1461–1468. (2014), https://doi.org/10.1109/TII.2014.2306331
28. Arrow K.J.: Social Choice and Individual Values. Wiley, New York, 1963.
29. Nguyen N.T: Processing Inconsistency of Knowledge on Semantic Level. Journal of Universal Computer Science, Vol. 11, No. 2, 285-302. (2005), https://doi.org/10.3217/jucs-011-02-0285
30. Dang D.T., Nguyen N.T., Hwang D.: A Quick Algorithm to Determine 2-Optimality Consensus for Collectives. IEEE Access, Vol. 8, 221794–221807. (2020), https://doi.org/10.1109/ACCESS.2020.3043371
31. Nguyen N.T: A Method for Ontology Conflict Resolution and Integration on Relation Level. Cybernetics and Systems, Vol. 38, No. 8, 781-797. (2007), https://doi.org/10.1080/01969720701601098
32. Pietranik M., Nguyen N.T.: A Multi-atrribute based Framework for Ontology Aligning. Neurocomputing, Vol. 146, 276-290. (2014), https://doi.org/10.1016/j.neucom.2014.03.067
33. Amodio S., Ambrosio A.D, Siciliano R.: Accurate Algorithms for Identifying the Median Ranking When Dealing with Weak and Partial Rankings under the Kemeny Axiomatic Approach. European Journal of Operational Research, Vol. 249, No. 2, 667–676. (2016), https://doi.org/10.1016/j.ejor.2015.08.048
34. Yang B.: Bioinformatics Analysis and Consensus Ranking for Biological High throughput Data. Ph.D. Dissertation, University of Paris 11. (2015).
35. Dang D.T., Phan H.T., Nguyen N.T., Hwang D. (2021) Determining 2-Optimality Consensus for DNA Structure. In: Fujita H., Selamat A., Lin J.CW., Ali M. (eds) Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices. IEA/AIE 2021. Lecture Notes in Computer Science, vol 12798, 427-438. (2021), https://doi.org/10.1007/978-3-030-79457-6_36
36. Ilinkin I., Ye J., Janardan R.: Multiple Structure Alignment and Consensus Identification for Proteins. BMC Bioinform., Vol. 11, No. 1, 71-80. (2010).
37. Dong Y., ChenX., Herrera F.: Minimizing Adjusted Simple Terms in The Consensus Reaching Process With Hesitant Linguistic Assessments in Group Decision Making. Information Sciences, Vol. 297, 95–117. (2015), https://doi.org/10.1016/j.ins.2014.11.011
38. Wu Z. Xu J.: Managing Consistency and Consensus in Group Decision Making with Hesitant Fuzzy Linguistic Preference Relations. Omega, Vol. 65, 28–40. (2016), https://doi.org/10.1016/j.omega.2015.12.005
39. Wu Z., Xu J.: Possibility Distribution-Based Approach for MAGDM With Hesitant Fuzzy Linguistic Information. IEEE Transactions on Cybernetics, Vol. 46, No. 3, 694–705. (2016).
40. Duong T.H, Nguyen N.T. et al.: A Collaborative Algorithm for Semantic Video Annotation Using a Consensus-based Social Network Analysis. Expert Systems With Applications, Vol. 42, No. 1, 246-258. (2015), https://doi.org/10.1016/j.eswa.2017.01.012
41. Radojičić, D., Radojičić, N., Kredatus, S.: A Multicriteria Optimization Approach for the Stock Market Feature Selection. Computer Science and Information Systems, Vol. 18, No. 3, 749–769. (2021), https://doi.org/doi.org/10.2298/CSIS200326044R
42. Sobieska-karpinska J., Hernes M.: Consensus Determining Algorithm in Multiagent Decision Support System with Taking into Consideration Improving Agent's Knowledge. Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), 1035–1040. (2012).

Dai Tho Dang received the M.S degree in computer science from the University of Nice Sophia Antipolis, Nice, France, and the Ph.D. degree in computer science from

Yeungnam University, the Republic of Korea. He is currently working as a lecturer at Vietnam-Korea University of Infomation and Communication Technology, The University of Danang. His research interests include collective intelligence, algorithm, consensus theory, inconsistent knowledge processing. He is a reviewer for journals *IEEE Transactions on Cybernetics*, *Artificial Intelligence Review*, *Applied Intelligence*.

Thanh Ngo Nguyen received the M.Sc. degree in computer science from Le Quy Don Technical University, Hanoi, Vietnam, in 2013. He is currently pursuing the Ph.D. degree with the Faculty of Information and Communication Technology, Wroclaw University of Science and Technology, Poland. He is currently a Research and Teaching Assistant with the Faculty of Information and Communication Technology, Wroclaw University of Science and Technology. His research interests include association rules and pattern mining.

Dosam Hwang received the Ph.D. degree in Kyoto University, Kyoto, Japan. He is a full professor of the Department of Computer Engineering at Yeungnam University in Korea, whose research interests mainly include Natural Language Processing, Ontology, Knowledge Engineering, Information Retrieval and Machine translation. He has also held a position as a principal researcher at Korea Institute of Science and Technology (KIST) and has also been a visiting professor at Korea Advanced Institute of Science and Technology (KAIST). He has so far been not only a co-chair of several international conferences but also a steering committee member of ICCCI and ACIIDS, and MISSI international conferences. He has been honored as a Distinguished Researcher of KIST in 1988 by Korea's Ministry of Science and Technology (MoST) and awarded a prize for Good Conduct from Kyunghee High School in 1973. He had more than 50 publications.

Received: March 14, 2021; Accepted: August 31, 2021.

Computer Science and Information Systems 19(1):455–493 https://doi.org/10.2298/CSIS210423066B

Automatic Derivation of the Initial Conceptual Database Model from a Set of Business Process Models*

Drazen Brdjanin[1], Aleksandar Vukotic[2], Danijela Banjac[1], Goran Banjac[1], and Slavko Maric[1]

[1] University of Banja Luka, Faculty of Electrical Engineering, Patre 5
78000 Banja Luka, Bosnia and Herzegovina
{drazen.brdjanin,danijela.banjac,goran.banjac,slavko.maric}@etf.unibl.org
[2] Automovens Ltd, Jovana Ducica 23a
78000 Banja Luka, Bosnia and Herzegovina
aleksandar.vukotic@rt-rk.com

Abstract. The article presents an approach aimed at automatically deriving the initial conceptual database model from a set of business process models. The approach proposes the incremental synthesis of the target model by iteratively composing the partial conceptual database models that are derived from the models contained in the source set. The approach is implemented by the AMADEOS tool, which is the first online web-based tool enabling the automatic derivation of the conceptual database model from a set of business process models. The experimental evaluation proves that the implemented approach enables effective automatic derivation of the initial conceptual database model.

Keywords: AMADEOS, BPMN, Business Process Model, Class Diagram, Conceptual Database Model, Model Composition, UML.

1. Introduction

With the development of the model-driven paradigm, *business process models* (BPMs) are playing an increasingly important role in the field of information systems and software engineering, serving as a basis for generation of the target system models. In our research we focus on using BPMs as a basis for the *model-driven synthesis of data models* (MDSDM). Furthermore, some recent experiments [7,20] imply that well-formed data-centric BPMs enable effective and efficient automatic synthesis of *conceptual database models* (CDMs).

Motivation. Although the idea of MDSDM, taking BPMs as a starting base, dates back in the early 1990s, in the existing literature there is only a small number of papers presenting the implemented automatic model-driven generator of the target data model with the corresponding evaluation results, while the great majority of papers only present modest achievements in (semi)automated, or even manual, data model synthesis. The surveys [17,69,14] show that the existing approaches are characterized by the *direct*

* This article constitutes an extended version of the conference paper entitled "*Automatic Derivation of Conceptual Database Model from a Set of Business Process Models*" presented at *INISTA – 2020 (International Conference on Innovations in Intelligent SysTems and Applications)*, August 24-26, 2020, Novi Sad, Serbia.

synthesis of the target model based on BPMs represented by a single concrete notation such as BPMN [58] or UML [59] activity diagram (UML AD). Furthermore, the existing tools are mainly implemented as plug-ins or transformation programs within development platforms (typically Eclipse-based), and also able to process only a single BPM (represented by a single diagram), although a real source model contains a finite set of models (diagrams). A more detailed overview of the related work is provided later in the article.

The surveys [17,69,14] show that a fully automated MDSDM approach is still the subject of extensive research, and the aforementioned problems of limited functionality, portability, effectiveness and efficiency of the existing MDSDM tools remain a significant research goal. In order to contribute the solving these problems, we have started a long-term research project focused on the automatic CDM synthesis based on a collection of models representing business processes of an enterprise, with the main objective to develop an online platform-independent tool for the automatic CDM derivation from a collection of BPMs that may be represented by different notations, and differently serialized, as well. The first very important milestone in our progress was the indirect two-phase CDM synthesis [21], which enables and facilitates derivation of the target data model from differently represented BPMs. By applying the two-phase CDM synthesis approach, we have launched the AMADEOS[3] system [19,36,14], which was the first online web-based system enabling automatic CDM derivation from BPMs that may be represented by two different notations and differently serialized. However, AMADEOS was able to automatically generate a CDM based on a single BPM, as the large majority of the existing MDSDM tools. A more detailed overview of the project progress is provided later in the article.

Objectives. The last-mentioned limitation of the AMADEOS system, i.e. its ability to automatically derive an initial CDM only from a single BPM, directly motivated our research presented in this particular article, with the following main objectives:

(1) *find an appropriate way to achieve automatic CDM derivation from a collection of BPMs, and extend the existing functionality of the AMADEOS system by applying the given aproach;*
(2) *evaluate to what extent is the improved AMADEOS system able to automatically generate the target initial CDM by applying this approach.*

Contributions. In order to achieve the first research objective, we propose the incremental synthesis of the target model by iteratively merging the partial CDMs that are derived from the BPMs contained in the source set. Such an approach enables the automatic synthesis of the target CDM by retaining and exploiting all existing capabilities of MDSDM tools (AMADEOS, in this case). This constitutes the first main contribution of this particular research. By applying this approach, we obtained an online web-based tool that publicly provides the MDSDM functionality by enabling the automatic CDM synthesis based on the set of BPMs represented by two concrete notations: BPMN and UML AD. This constitutes the second main contribution of this research. In order to achieve the second research objective, we performed very extensive case study-based and experimental evaluation focused on the approach effectiveness, through the assessment of correctness

[3] Available at: *http://m-lab.etf.unibl.org:8080/amadeos/*

and completeness of the CDMs automatically derived from the sets of BPMs. The experimental results prove that the implemented approach enables effective automatic derivation of the initial CDM (particularly for classes).

Some of these research results have been already presented at the *INISTA-2020 Conference*. This particular article constitutes an extended version of the corresponding conference paper [26], which is extended by: (i) a detailed overview of the related work, (ii) a detailed presentation of the proposed approach, and (iii) the results of the experimental evaluation of the implemented approach.

Article organization. The article is structured as follows. After this introductory section, Section 2 presents the related work. Section 3 presents the approach, while Section 4 presents the implemented tool. Section 5 provides an illustrative example of automatic CDM derivation from a set of BPMN models. Section 6 presents the evaluation of the implemented approach. Finally, Section 7 concludes the article and gives directions for future work.

2. Related Work

In this section we firstly provide an overview of the existing MDSDM approaches[4], then we position our approach and present the advancements in comparison with our previous work and other related approaches.

Overview of the existing approaches. The existing MDSDM approaches, regarding the primary focus of the source notation, can be classified into four main categories [17]: *process-oriented*, *goal-oriented*, *function-oriented*, and *communication-oriented*.

Figure 1 presents a chronological overview of the existing MDSDM approaches, grouped by the source notation. Different marks are used to differentiate: (i) *completeness of the source model* – a source model can be *complete* or *partial* (partial model contains a single diagram, although a real model contains a finite set of diagrams), and (ii) *automatization level of the approach*, which can be *manual* (not supported by any software tool), *semiautomatic* (supported by a tool, but designer's assistance is still required), or *automatic* (without designer's assistance). The arrows depict improvements in the same approach, presented in different papers.

POM-based approaches. Our approach belongs to the most dominant category of the MDSDM approaches that take *process-oriented models* (POMs) as a basis for the MDSDM. Although the first POM-based approach [75] was proposed back in 1990, the boom of the POM-based MDSDM approaches was highly influenced in the last 15 years by the development of metamodel-based notations (particularly UML AD and BPMN) and specialized model-to-model transformation languages (ATL[5] and QVT[6]). Apart from BPMN and UML AD, which are dominantly used, the existing POM-based approaches also take source models represented by: Petri Net, RAD (Role Activity Diagram), EPC (Event-driven Process Chain), TCD (Task Communication Diagram), and A-graph.

[4] For a more detailed survey we refer the readers to [17].

[5] ATLAS Transformation Language [44]

[6] Query/View/Transformation [57]

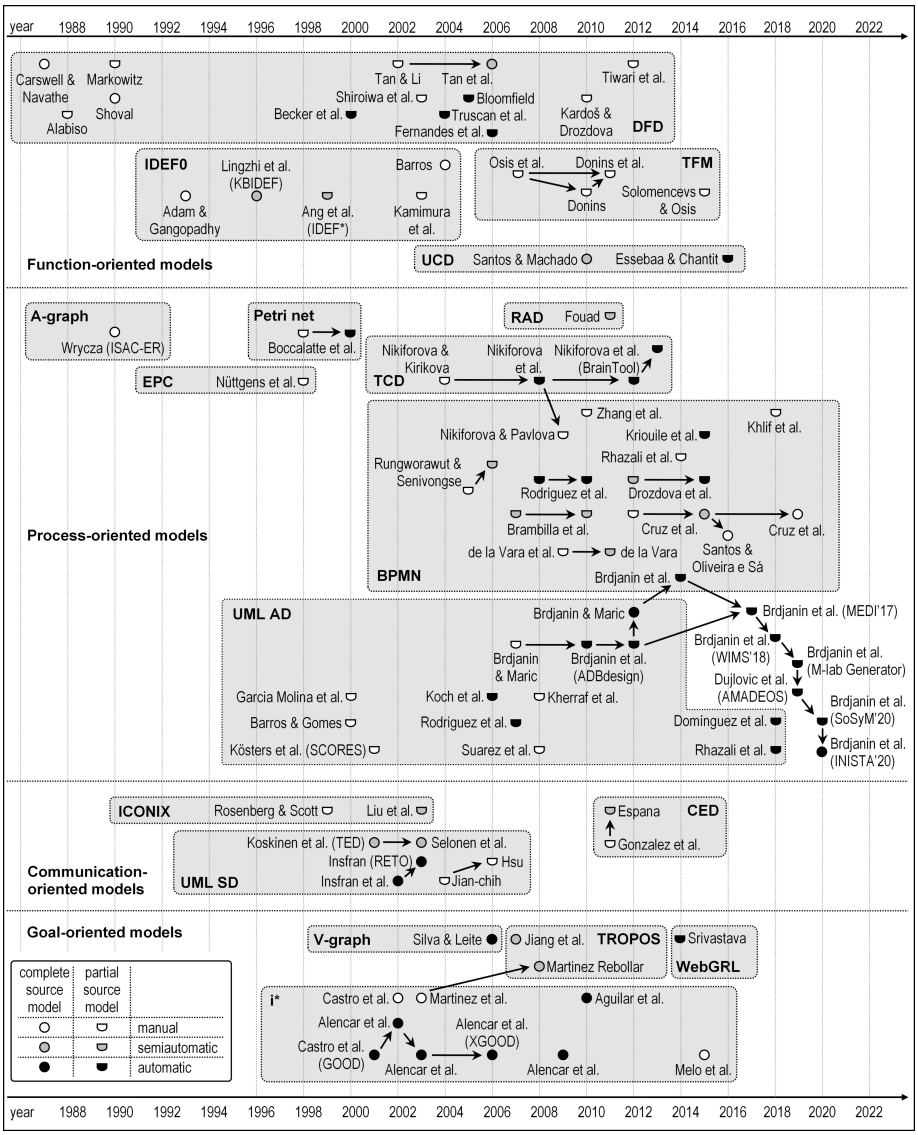


Fig. 1. Chronological overview of MDSDM approaches (based on [17,14])

Although more than 40 POM-based papers have been published in the last 30 years, there is still no approach and the corresponding tool that enable automatic synthesis of the complete target model, whereby the most approaches allow for (semi)automated synthesis with modest correctness and completeness. Moreover, many of them are based on guidelines and informal rules that do not allow automatic synthesis at all. Finally, almost all approaches consider a single process model as a basis for MDSDM, although a real model constitutes a collection of process models. Only three papers [75,25,29] consider the complete source model, but only [25] presents the implemented tool (*ADBdesign*).

Among about twenty papers that consider the BPMN-based MDSDM, only a few papers [64,63,49,34] present some tool (mainly ATL- and QVT-based transformation programs) enabling the automatic MDSDM, but with very low effectiveness. The semi-automatic BPMN-based MDSDM is presented in [65,12,13,32,35], while the other proposals [76,56,35,45,31] are not implemented at all. Regarding the formalism level of the existing BPMN-based approaches, the formal rules are presented in [23,66], and partially in [30,29,31], while the others give only the informal guidelines. Regarding the source model completeness, only three papers [29,31,66] consider a collection of the source models, but none of the proposed approaches have been implemented.

Among more than ten papers that consider UML AD as a basis for MDSDM, only [25] considers a collection of the source models and presents the implemented automatic tool (*ADBdesign*), while the majority [46,47,62,18,15,24,16,33,61] present the automatic data model derivation from incomplete source models, but with very low effectiveness.

There are also several related papers proposing the usage of TCD notation as a starting point for MDSDM, initially through an intermediate model, while [54] presents the BrainTool generator, which generates the data model directly from TCD. However, like the majority, they do not consider the complete source model. Among the other POM-based approaches, only two papers [11,40] present tools for the (semi)automatic data model synthesis based on the partial source model.

Other MDSDM approaches. Since our approach belongs to the POM-based approaches, here we provide only a short overview of other MDSDM approaches.[7]

The *function-oriented models* (FOMs), used as a basis for MDSDM, are represented by four different notations: DFD (Data Flow Diagram), IDEF0, TFM (Topological Functioning Model), and UML UCD (Use Case Diagram). Although the first ideas about the FOM-based MDSDM appeared back in the late 1980s, the survey shows that the semantic capacity of FOMs has not been sufficiently identified to enable automatic synthesis of the complete target data model. The large majority of the approaches are based on guidelines and informal rules, and take an incomplete source model as the basis. The automatic data model generation is presented in [9,73,39,10,38], while the semi-automatic generation is presented in [72,51,6,67].

The *goal-oriented models* (GOMs), used as a basis for MDSDM, are represented by the i* notation and some i*-originated notations like TROPOS, V-graph, and WebGRL. The automatic (to some extent) GOM-based MDSDM, based on the complete source model, is presented in [27,4,3,5,70,43,53,2,1,71].

The *communication-oriented models* (COMs), used as a basis for MDSDM, are represented by three different notations: ICONIX (Robustness Diagram), CED (Communicative Event Diagram) and UML SD (Sequence Diagram). The automatic data model synthesis, based on the complete source model, is presented in [41,42], while the semi-automatic synthesis is presented in [48,68,52,37].

Evaluation of the existing approaches. The large majority of all existing MDSDM approaches are not evaluated at all. Most of the papers reporting evaluation results mainly focus on approach usability, but not on the qualitative and/or quantitative assessment of the implemented tools or generated data models.

The GOM-based approaches are not evaluated.

[7] For more detailed overviews, we refer the readers to [17,20].

Only one COM-based approach [37] is evaluated based on lab demos and an experiment with students (reported model completeness is ∼70%).

Only [72] presents evaluation results of a FOM-based approach, but the authors do not focus on the assessment of the method effectiveness and efficiency.

Regarding the POM-based MDSDM approaches, the case-study based evaluation results are reported in [23,25,55], while the results of controlled experiments are reported in [32,40,7,22,20]. The most complete evaluation results, which are based on the experiment conducted with a significant number of practitioners, are presented in [22,20] (average completeness of the generated models is over 80%).

Comparison with our previous work and other related approaches. In this article we present the most recent developments in a long-term research project that is aimed at automatic BPM-driven CDM synthesis. As depicted in Fig. 1, the initial ideas for the manual data model derivation were presented more than ten years ago, while the first implementation (*ADBdesign*) was presented back in 2010 [18]. After the transformation rules were upgraded and formalized [16], the automatic MDSDM from a collection of the UML ADs was presented in 2012 [25]. Based on the experimentally confirmed [7,20] semantic capacity of data-centric BPMs, a two-phase BPM-driven approach to the CDM synthesis was proposed [21], and the first online service-oriented BPM-driven CDM generator (*M-lab Generator*) was implemented [19]. Finally, the AMADEOS system [36,14] was launched, as the first online web-based system for BPM-driven CDM synthesis. AMADEOS implements the CDM synthesis process through an orchestration of the *M-lab Generator* services, and enables the CDM synthesis based on BPMs that may be represented by BPMN or UML AD. Since the given approach enables the CDM synthesis based on differently represented source BPMs, all the developments after [21] are depicted outside of any POM region in Fig. 1.

The pre-existing AMADEOS release [14] was able to automatically derive the target model from a single BPM. In this article we present the most recent achievement in the entire project and development of the AMADEOS system – expanding its functionality to enable the automatic CDM derivation from the set of BPMs. In this way, we obtained the first online tool that publicly provides the MDSDM functionality based on a set of BPMs that may be represented by two different notations (BPMN or UML AD) and differently serialized (XMI[8] or XSD[9]).

Unlike the other existing MDSDM tools, AMADEOS is not dependent on any particular modeling platform and enables automatic CDM generation in a web browser, without any installation of tools or plug-ins. It enables users to upload a collection of BPMs, to generate the CDM, as well as to export the generated CDM in the XMI format for further processing in some other database design tool. In this way, AMADEOS may be very beneficial both for industrial and academic purposes – database designers are provided with a tool for BPM-based MDSDM, software engineers are provided with online services that may be invoked from their own tools, while researchers are able to compare their tools against AMADEOS. Currently, there are no other publicly available online tools that enable CDM generation based on a collection of BPMs, only some indications of plug-ins and transformation programs that are not publicly available.

[8] XML Metadata Interchange
[9] XML Schema Definition

3. Approach to Automatic CDM Derivation

This section presents the proposed approach to automatic CDM derivation from a set of BPMs. Firstly, we provide an overview of the whole process and the corresponding high-level algorithm for the incremental CDM synthesis. Then we present the major functional building blocks of the entire approach: (1) semantic capacity of the BPMs for the automatic CDM synthesis, (2) two-phase synthesis of the partial CDMs, and (3) composition of the partial CDMs. Finally, we present applied techniques to overcome some inconsistencies in the source set of BPMs.

3.1. Incremental CDM Synthesis

The starting point for the automatic CDM derivation is a set BM that contains a finite number of BPMs, i.e. $BM = \{bpm_1, \ldots, bpm_i, \ldots, bpm_n\}$.

The approach proposes the incremental synthesis of the target CDM (denoted by cdm) by iteratively composing the partial CDMs that are derived from the models contained in the source set BM. The entire process is formally specified by the high-level algorithm presented in Fig. 2, and illustrated in Fig. 3. The target model cdm is initially empty. For each model bpm_i from the source set BM, we obtain the corresponding partial CDM (cdm_i) by applying the appropriate transformation $\mathcal{G}$. This partial CDM and the result (cdm) of all previous iterations are composed ($\oplus$ operator) into the resulting model cdm. After processing all models from the source set BM, the resulting model represents the target CDM. The presented approach to the incremental synthesis of the target CDM corresponds to the well-known *binary ladder integration strategy* [8] of partial (conceptual) schemas.

The aforementioned high-level transformation $\mathcal{G}$ represents the entire process of the CDM derivation from a single source BPM. It is described in the following two subsections (3.2 and 3.3), while subsection 3.4 presents the model composition.

1: $cdm \leftarrow \emptyset$
2: **for all** $bpm_i \in BM$ **do**
3: $\quad cdm_i \leftarrow \mathcal{G}(bpm_i)$
4: $\quad cdm \leftarrow cdm \oplus cdm_i$
5: **end for**

Fig. 2. High-level algorithm for incremental CDM synthesis

3.2. BPM as Basis for CDM Derivation

BPMs contain some typical concepts and represent some typical facts that are inherent to business processes but may be differently represented by different modeling notations. Regardless of the applied notation, those facts and concepts possess a certain *semantic capacity* that allows the automatic CDM synthesis. Here we provide a short overview of the identified semantic capacity of BPMs for the automatic CDM synthesis (Fig. 4).[10]

[10] For the complete formal specification of the transformation rules for the direct BPM-driven CDM synthesis we refer the readers to [16,20]. Without loss of generality, source BPM concepts are represented by BPMN.

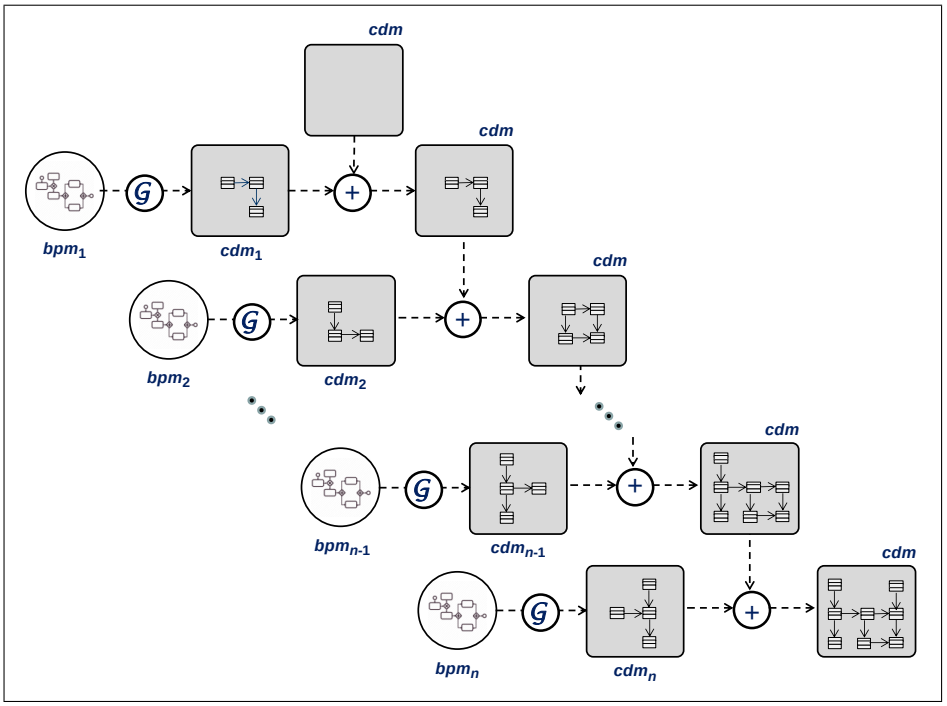


Fig. 3. Incremental CDM synthesis (binary ladder integration strategy)

The *entity types* in the target CDM can be derived from the following BPM concepts: *participants*, *roles*, *message flows*, *objects*, and *activations of existing objects*. The mapping of *participants* and their *roles* into the corresponding classes is specified by the T_1 rule (Fig. 4), while the mapping of different types of *objects* and *message flows* is specified by the T_2 rule. The T_3 rule specifies the mapping of the *activated existing objects*, i.e. objects that are created in other processes but *activated*[11] in the given process.

The *relationships* between classes in the target CDM can be automatically derived from several typical BPM patterns. The T_4 rule specifies the generation of the *participant-participant* associations between the class representing a *participant* and the classes representing its *roles*. The automatic generation of the *participant-object* associations is specified by the following rules: T_5 – *creation and subsequent usage of the generated objects*, T_6 – *exchange of messages*, and T_7 – *activation and subsequent usage of the activated objects*. The automatic generation of the *object-object* associations is specified by two transformation rules: T_8 – *association between the existing object and the corresponding activation class*, and T_9 – *tasks having input and output objects of different types*.

[11] An *activation* represents the fact that an existing object constitutes input in a task that changes its state. After activation, the *activated* object may be further used in some subsequent tasks, in the same way as generated objects are used.

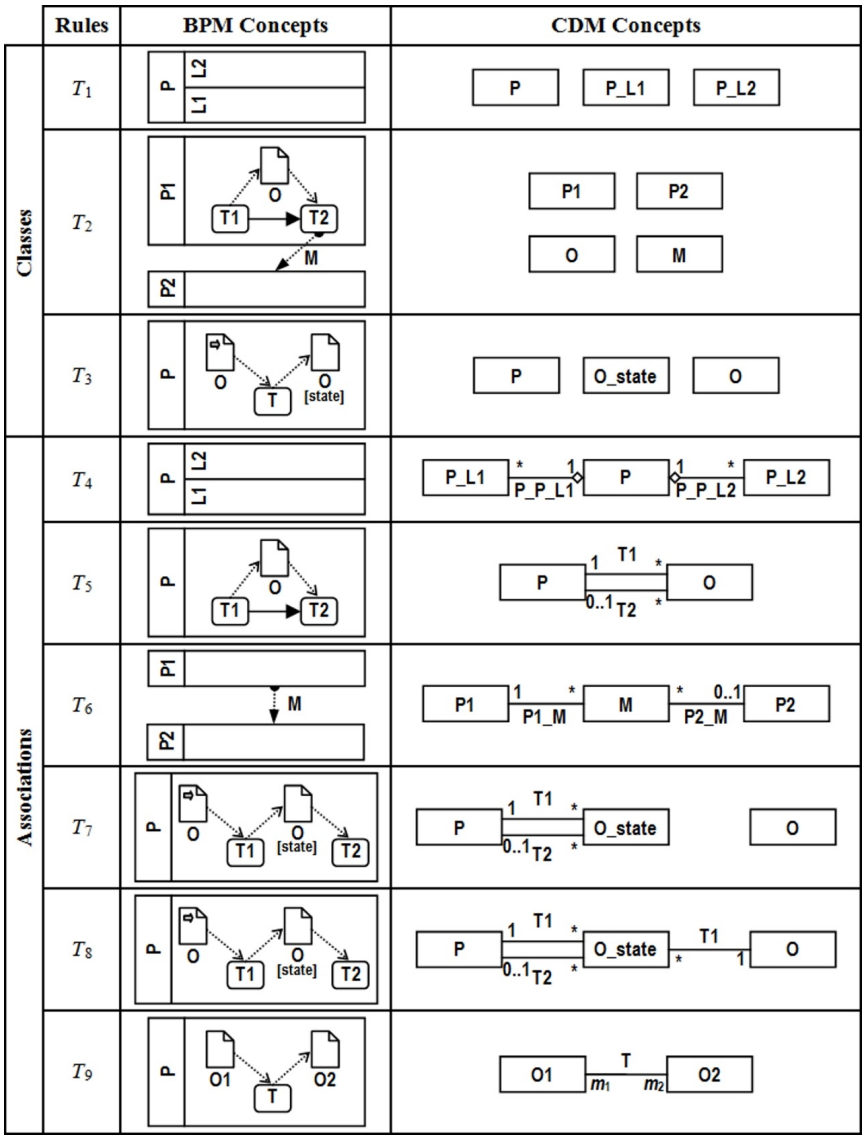


Fig. 4. Mapping of BPM concepts into CDM concepts [14]

3.3. Two-phase BPM-driven Synthesis of Partial CDMs

Each model transformation includes two types of actions: (1) *extraction* of specific elements from the source model(s), and (2) *generation* of the corresponding elements in the target model. In the existing MDSDM approaches, these two types of actions are strongly coupled and implemented by a single transformation program that takes the source BPM (represented by some concrete notation such as BPMN) and generates the target model (represented by another concrete notation, typically class diagram).

In order to overcome disadvantages of the *direct synthesis*, AMADEOS implements the *two-phase CDM synthesis* [21,19], which means that the extracting and generating actions are decoupled and separated into two consecutive activities (phases). In the first phase, appropriate *extractors* extract specific concepts from the source BPM and represent them by BMRL[12]. In the second phase, the *generator* generates the target CDM (UML class diagram) based on the BMRL-represented extracted concepts. Figure 5 illustrates the technical perspective of the approach, while Fig. 6 illustrates the two-phase CDM synthesis based on a simple BPMN model.[13] As illustrated in Fig. 5, the two-phase approach enables simple extensibility and support for other process-oriented notations (which are not necessarily metamodel-based) by implementing additional extractors.

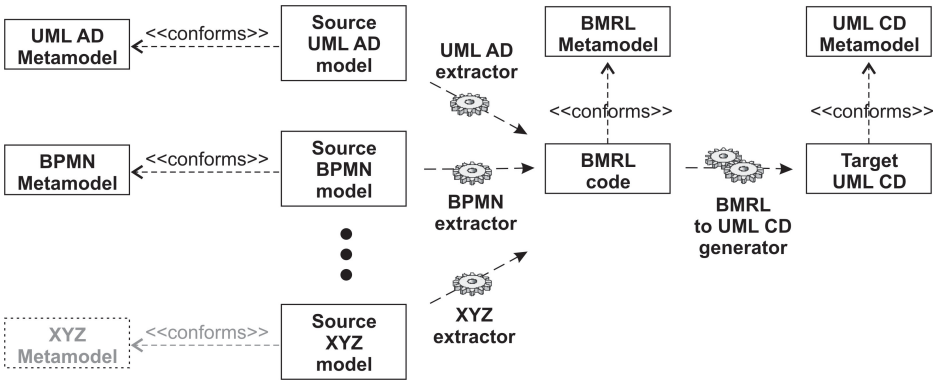


Fig. 5. Technical perspective of two-phase BPM-driven CDM synthesis [19]

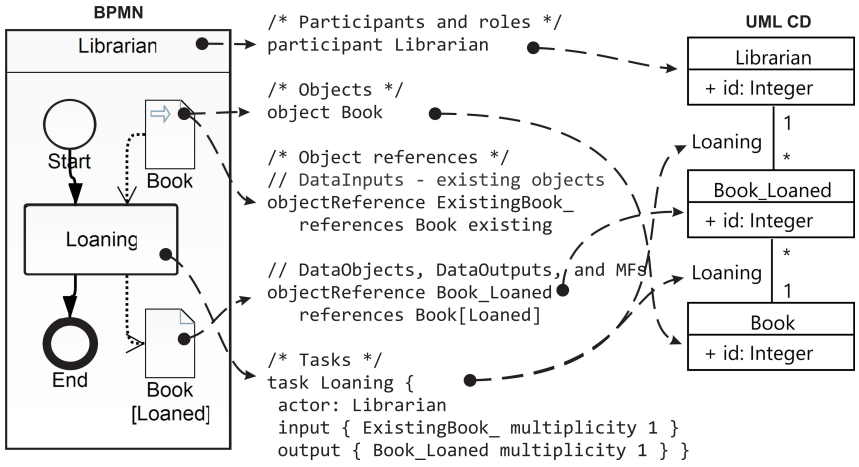


Fig. 6. From BPM (BPMN) through BMRL to CDM (UML class diagram)

[12] BMRL (*Business Model Representation Language*) is a simple domain-specific language aimed at representing the BPM concepts having the semantic capacity for automatic CDM synthesis. For the complete specification of BMRL, we refer the readers to [21,19].

[13] For the complete formal specification of the rules for the both phases we refer the readers to [21,19].

3.4. Partial CDM Composition

The whole CDM synthesis process is performed iteratively. In each iteration, the target CDM is incrementally built by composing two partial CDMs – a CDM derived from one of the source BPMs, and a CDM obtained for already processed source BPMs. In other words, in each iteration two UML class diagrams are to be composed into a single model. The model composition is performed by applying two sets of rules (Fig. 7), whereby each group deals with specific elements.

The first group (R1) of the rules, which considers the classes and their properties, consists of the following rules:

Rule R1.1: Each class is identified with its name[14]. If both source CDMs contain the same-named classes, we conclude that they both represent the same class, and the composition result (resulting CDM) is to contain only one corresponding class of the same name. If a class contained in one of the source CDMs does not yet exist in the resulting CDM, the corresponding same-named class is to be created in the resulting CDM. If the same-named class already exists in the resulting CDM, a new class will not be created in the resulting CDM.

Rule R1.2: Each class property is identified with its name. When a new class is added to the resulting CDM, all properties are copied from the source class to the target class in the resulting CDM. If the same-named class already exists in the resulting CDM, only missing properties are to be added to the target class in the resulting CDM.

The second group (R2) of the composing rules, which deals with the classes relationships, consists of the following rules:

Rule R2.1: The association, aggregation, and composition relationships are identified by their names and the corresponding classes among which the relationships exist. If the same-named relationship exists between different classes, they are to be treated as different relationships during the composition.

Rule R2.2: The association, aggregation, and composition relationships represent the structural relationships between objects, but have different weights in the semantics that are to be considered during the model composition, i.e. $association \prec aggregation \prec composition$ (association is the weakest relationship type). If two the same-named, but differently strong, relationships exist between the same-named classes in both source CDMs, then the resulting CDM is to contain the weaker same-named relationship between the corresponding classes.

Rule R2.3: The relationship ends have the multiplicities represented by the lower (*lbv*) and upper (*ubv*) bound values ($[lbv..]ubv$, $lbv = m$, $ubv = n \vee *$, $m \in \mathbb{N}_0$, $n \in \mathbb{N}$). During the composition, lower and upper bound values of multiplicities are compared for the conflicting relationships (the same-named relationships between the same-named classes), and the most flexible bound values ($min(lbv)..max(ubv)$) are to be set for the corresponding relationship ends in the resulting CDM.

Rule R2.4: The generalization relationships do not have names. If the resulting CDM does not already contain the given generalization relationship that exists in the source CDM, the corresponding generalization relationship is to be created between the corresponding classes in the target model.

[14] Strictly speaking, each serialized model element is uniquely identified by the corresponding *id* attribute, but in the context of the model composition, the characteristic model elements are identified by their names.

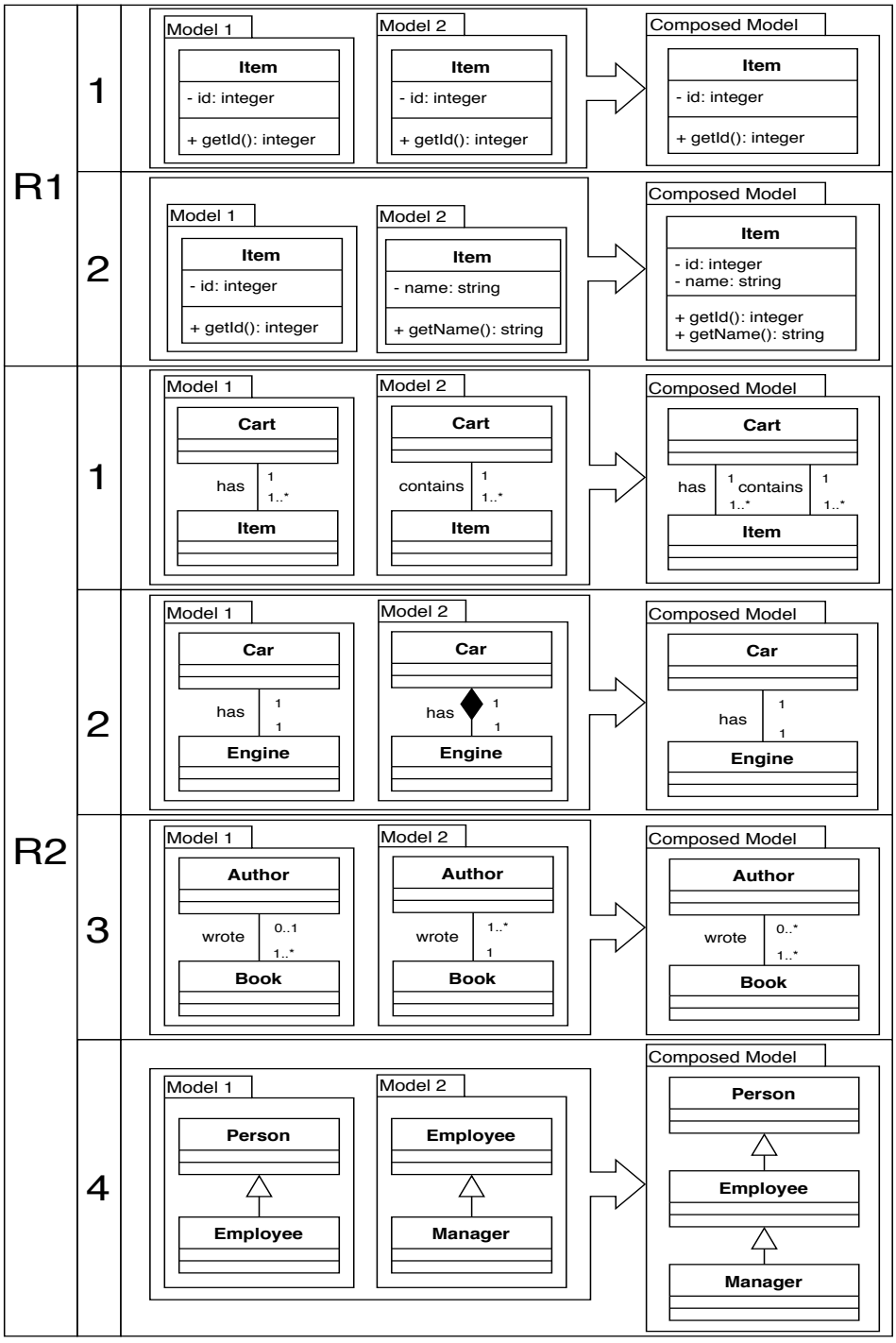


Fig. 7. Rules for composing partial CDMs

3.5. Dealing with Source Model Inconsistencies

The aforementioned model composition rules are applicable to an ideal set of the source models. However, a number of modelers are usually involved in business process modeling, and if strict modeling guidelines are not set before starting the modeling process, there is a high probability that some inconsistencies occur in the created models. These inconsistencies can be seen in the usage of different naming notations, synonyms, etc.[15]

To overcome some inconsistencies, AMADEOS provides its users a possibility to use advanced composition approach. This means that the aforementioned set of composition rules is applied, but AMADEOS additionally tries to overcome different naming notations problem and accidental typing errors. These goals are achieved by combining the following techniques: (1) case sensitivity and (2) Levenshtein distance.

Case sensitivity defines whether lowercase and uppercase letters in text are treated as distinct (case-sensitive) or equivalent (case-insensitive). Advanced composition is using case-insensitive comparison of strings during composition, which helps to overcome the problem of different naming notations.

The Levenshtein distance (LD) [50] is used[16] to overcome accidental typing errors. During advanced model composition, the measured distance between names of the model elements (i.e. the minimum number of single character insertions, deletions or substitutions required to transform one word to the other) is checked against the threshold value that is specified in the configuration file of the composer service (at the moment, threshold value is set to 1). Specified threshold value represents the number of single-character edits that are allowed. If measured distance is lower or equal to threshold value, words will be treated as equivalent.

4. Implemented Tool

The presented approach is implemented in the AMADEOS system. Figure 8 shows the architecture of the improved system. In comparison with the pre-existing version, the improved system ia able to automatically generate the target CDM based on a set of source BPMs. AMADEOS currently supports two different source notations (BPMN and UML AD), but all models in the entire source set must be represented by the same notation. Source UML AD models can be XMI-serialized, while BPMN models can be XMI- or XSD-serialized (all models in the entire set must be serialized in the same way).

The server-side is implemented as a set of web services. The *Orchestrator* service orchestrates the entire process, while each particular activity is implemented by the corresponding web service. The first phase of the CDM synthesis process is implemented by the *BPMN extractor* and *UML AD extractor* services. These two services implement the extraction of the characteristic concepts from the source BPM, and generation of the corresponding BMRL code. The second phase of the CDM synthesis process is implemented by the *UML CD generator* service, while the model composition is performed by the *UML CD composer* service. Finally, the *UML CD layouter* service is aimed at automatic layout of the corresponding UML class diagram.

[15] For the most comprehensive taxonomy of merging conflicts we refer the readers to [60].

[16] In order to eliminate inconsistencies in the source BPMs, various techniques can be applied. AMADEOS currently applies a common technique based on LD. Some other algorithms could be used as well, like Jaaro-Winkler [74], which could be a part of future work.

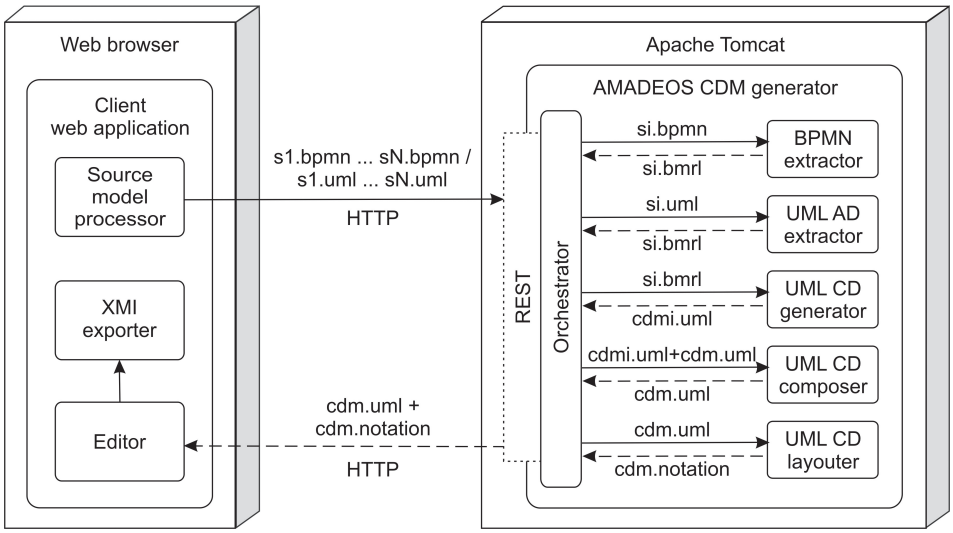


Fig. 8. Architecture of AMADEOS system

In a positive usage scenario[17] (Fig. 9), the orchestrator service receives a set of the source BPMs (*s1.bpmn...sN.bpmn/s1.uml...sN.uml*), and returns the corresponding automatically generated CDM (*cdm.uml+cdm.notation*). For each source BPM, the orchestrator orchestrates the two-phase synthesis process that results with the corresponding partial CDM, and incrementally builds the target CDM. More concretely, for each source model (*si.bpmn/si.uml*), the orchestrator firstly sends it to the corresponding extractor service, which generates and returns the corresponding BMRL code (*si.bmrl*). The orchestrator further sends the BMRL code to the generator service that generates and returns the corresponding partial CDM (*cdmi.uml*). This partial CDM and the CDM obtained for already processed source BPMs, are further sent (*cdmi.uml+cdm.uml*) to the composer service that performs model composition and returns the composed CDM (*cdm.uml*). After processing all source BPMs, the orchestrator forwards the resulting CDM (*cdm.uml*) to the layouter service, which automatically generates and returns the layout of the corresponding diagram (*cdm.notation*). Finally, the model and the diagram are merged by the orchestrator into a single JSON[18] object (*cdm.uml+cdm.notation*), and returned to the client.

The *client web application* (Fig. 10) allows users to upload a set of source BPMs to the orchestrator service. When the entire synthesis process is finished, the client application receives the JSON response and visualizes[19] the class diagram in the browser (*Editor* component). The visualized diagram is editable so users can additionally improve it. It is also possible to export the model in the XMI format (*XMI exporter*), and further use it in some other modeling tool.

[17] This assumes that a user has selected appropriate options in the user interface and uploaded an appropriate collection of BPMs.

[18] JavaScript Object Notation

[19] The implementation is based on the jsUML2 library (*http://www.jrromero.net/tools/jsUML2*).

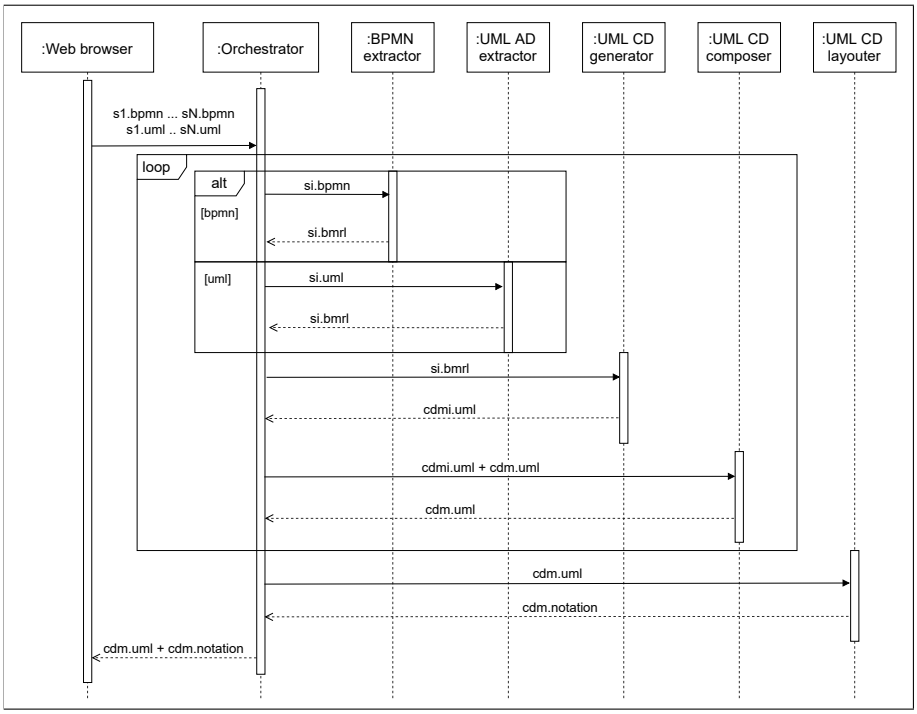


Fig. 9. Sequence diagram representing positive usage scenario of AMADEOS system

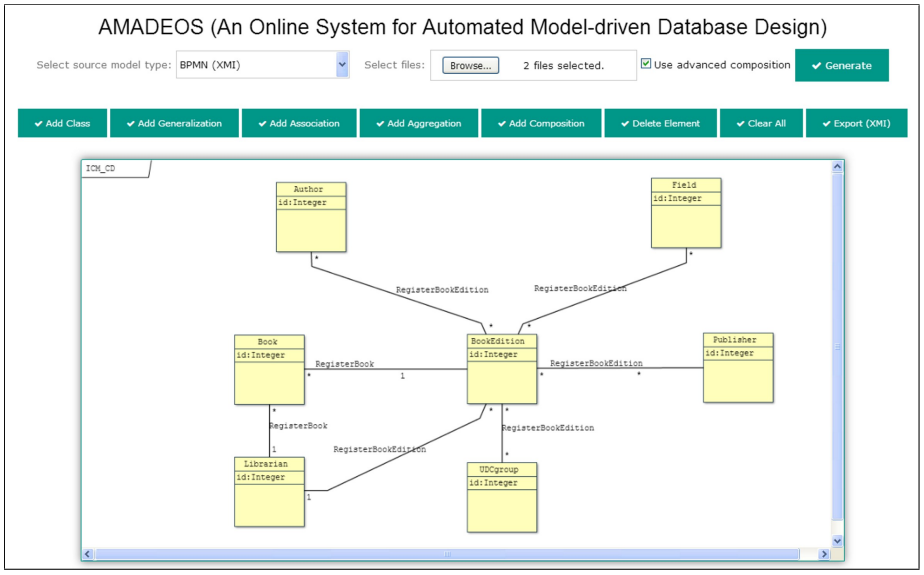


Fig. 10. Screenshot of AMADEOS client web application

5. Illustrative Example

In this section we provide an example that illustrates the implemented approach. This example is provided here for the purpose of the approach illustration, while the next section presents the results of the experimental evaluation of the entire approach.

The sample source set of BPMs (Fig. 11), which is used in this example, represents three main processes in the Faculty Library: (1) Book Edition Registration, (2) Book Registration, and (3) Book Write Off. These three simple, mutually related models belong to the set of BPMs of the Faculty Library, which is used for evaluation (next section) of the implemented approach.

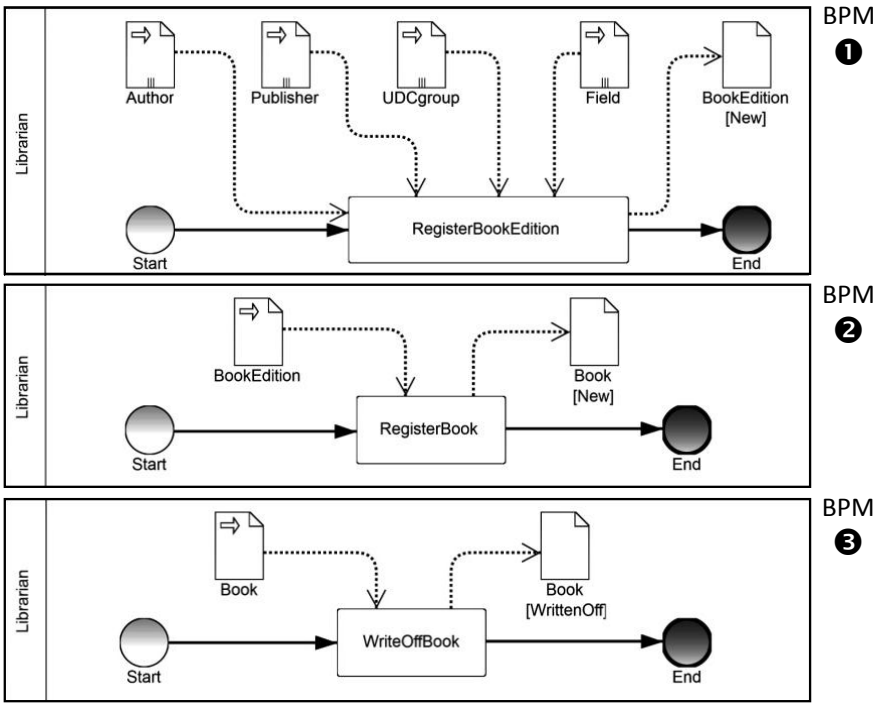


Fig. 11. Sample source set of BPMN models

Figure 12 illustrates the process of the incremental CDM composition based on the sample source set of BPMs. Firstly, CDM Generator generates the corresponding CDM based on the first BPM (Book Edition Registration). After that, CDM Generator derives the corresponding CDM from the second BPM (Book Registration), and then UML CD Composer composes these two partial CDMs into the resulting CDM. Finally, UML CD Composer combines this resulting CDM with the CDM derived from the third BPM (Book Write Off), and generates the CDM that corresponds to the sample source set.[20]

[20] Figure 12 contains class diagrams that correspond to the automatically generated CDMs in the AMADEOS system, after some manual improvements of the automatically generated layout.

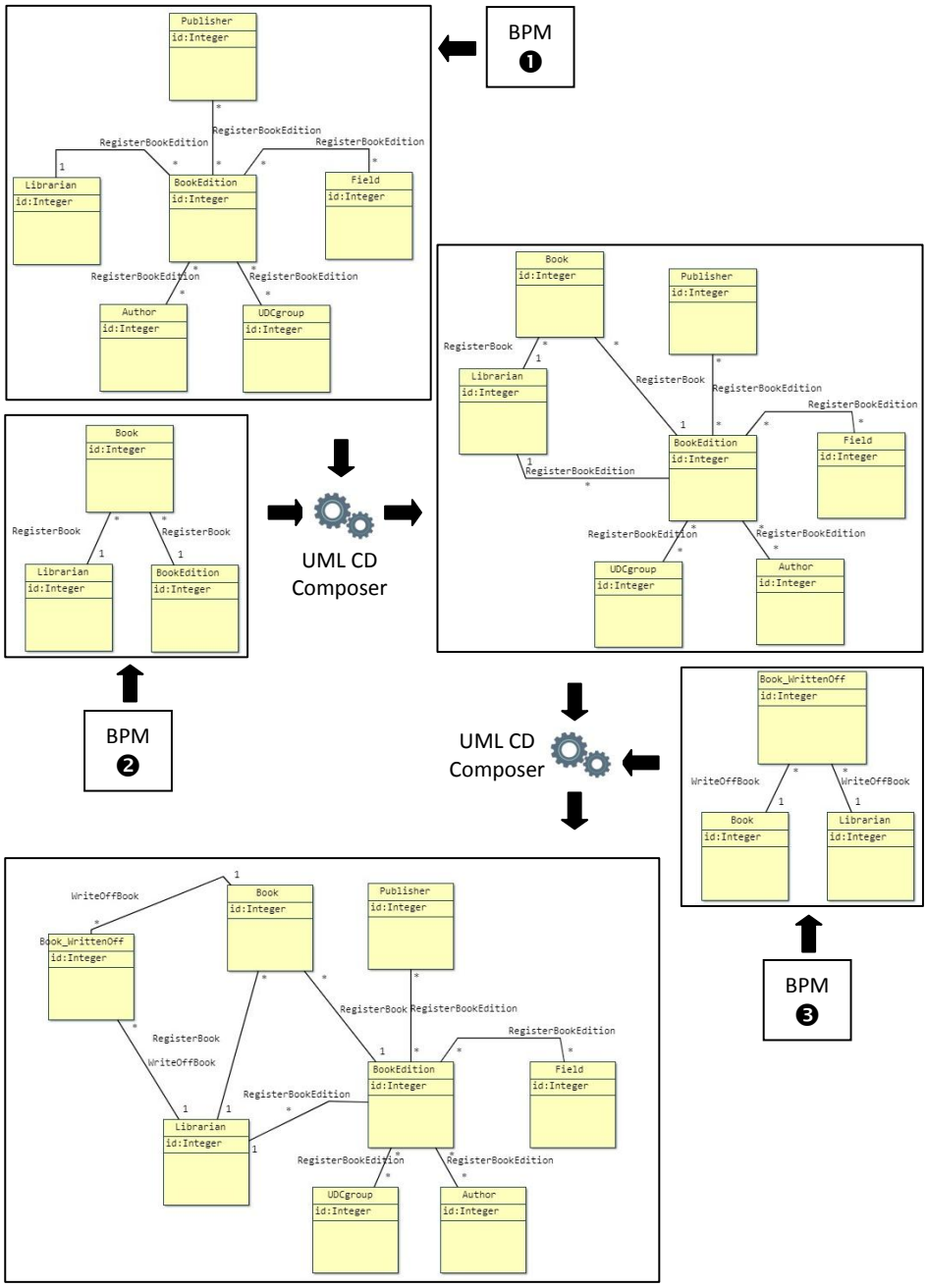


Fig. 12. Illustration of incremental CDM composition based on sample source set of BPMN models

6. Evaluation

In order to evaluate the implemented approach, we conducted a very extensive evaluation focused on the approach effectiveness, through the assessment of correctness and completeness of the CDMs automatically derived from the sets of BPMs. Firstly, we performed a case study-based evaluation. After that, we performed several experiments in order to verify the obtained case study-based results. This section presents these evaluation activities and obtained results – firstly the case study, followed by the experiments. Finally, we consider threats to validity of the experiments and derived conclusions, as well as lessons we learned.

6.1. Case Study

The implemented approach has been firstly evaluated through a case study of the Faculty Library Information System, through the assessment of the CDM, which is automatically derived from a set of BPMs of the Faculty Library, with a CDM that corresponds to the Library database.

Reference models. Figure 13 depicts a class diagram representing the CDM that corresponds to the existing (current) Library database within the Faculty Library Information System. The given class diagram is manually designed based on the corresponding relational database schema.[21] In the rest of the article, we use the *reference CDM* term for this model, since it is used later as a reference in the experiments.

In order to evaluate the approach and implemented tool, we used a set of BPMs of the Faculty Library. This collection contains 14 models[22], some of which are already shown in the example illustrating the incremental composition process in the previous section. The labels and names of the corresponding business processes are shown in Table 1. The used collection contains seven models of *main* (operational/transactional) *processes* (R-01...R-07) and seven models of *auxiliary processes* (R-08...R-14). The main processes allow the creation of new data (e.g. R-01 represents the process of registering a new book edition, R-02 represents the process of registering a new book, etc.). The auxiliary processes enable the maintenance of the existing registers and the modification of the existing data (e.g. R-08 represents the process of modifying authors' data, etc.). This collection of BPMN models was also (partly) used in the experiments and is hereinafter referred to as the *reference set of BPMs*.

Evaluation results. Based on the reference set of BPMs, CDMs were automatically generated using the AMADEOS tool, and then the generated CDMs were compared with the reference CDM of the Faculty Library Information System (Fig. 13). We were particularly interested in the contribution of the *models of the main processes* (*main BPMs* in the rest of the article) and the contribution of the *models of the auxiliary processes* (*auxiliary BPMs* in the rest of the article). Therefore, we generated CDMs based on each reference BPMN model, as well as based on the partial reference set containing only the main BPMs, and the complete reference set of BPMs.[23]

[21] Note that class attributes are left out because at the moment AMADEOS is not able to generate class attributes, except a single attribute for each class, which represents its primary key.

[22] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/CS/Ref-BPMs*

[23] The generated models are available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/CS/CDMs*

Table 1. Reference BPMN models of main (left) and auxiliary processes (right)

Main processes		Auxiliary processes	
ID	Process name	ID	Process name
R-01	Book Edition Registration	R-08	Author Update
R-02	Book Registration	R-09	Book Edition Update
R-03	Member Enrollment	R-10	Publisher Update
R-04	Member Seceding	R-11	Book Update
R-05	Book Borrowing	R-12	Field Update
R-06	Book Returning	R-13	UDC Group Update
R-07	Book Write Off	R-14	Member Record Update

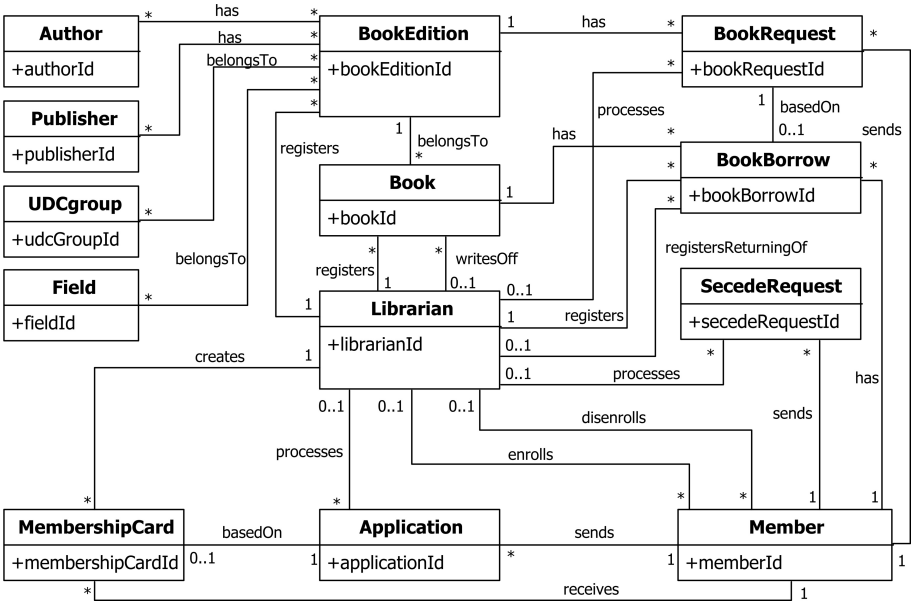


Fig. 13. Reference CDM of Faculty Library Information System

We use the following metrics for the quantitative evaluation of the generated CDM in comparison with the reference CDM:

- N_g – total number of generated concepts in the generated CDM,
- N_c – number of correctly generated concepts in the generated CDM, i.e. number of concepts that are identical to the concepts contained in the reference CDM,
- N_w – number of incorrectly generated concepts in the generated CDM, i.e. number of concepts that are not present in the reference CDM, and
- N_m – number of missing concepts in the generated CDM in comparison with the reference CDM.

We use *recall*, *precision*, and *F-score* as measures for the evaluation of the automatically generated CDM.

Recall (R) constitutes the measure of *completeness* of the generated CDM in comparison to the reference CDM. It may be defined as follows:

$$R = \frac{N_c}{N_c + N_m}. \tag{1}$$

Precision (P) constitutes the measure of *correctness* of the generated CDM. It may be defined as follows:

$$P = \frac{N_c}{N_c + N_w}. \tag{2}$$

F-score (F) constitutes the *effectiveness* measure. It represents the harmonic mean of precision (P) and recall (R), and it may be defined as follows:

$$F = \frac{2PR}{P + R}. \tag{3}$$

The results of the comparison of automatically generated CDMs with the reference CDM are given in Table 2. The first 14 rows (rows labeled from "01" to "14", where the label corresponds to the source model ID) contain results for the CDMs derived from the corresponding single BPMN model contained in the reference set of BPMs. The row labeled with "01-07" contains results for the CDM derived from the partial set of the main BPMs, while the row labeled with "01-14" contains results for the CDM derived from the complete reference set of BPMs. Figure 14 shows a comparison of R, P, and F for the generated classes (top) and associations (bottom) based on the reference BPMN models in comparison with the reference CDM.

Results discussion. The comparison results of the generated CDMs with the reference CDM show that none of the individual source BPMN models from the reference set has the semantic capacity to enable the generation of neither all classes nor all associations that exist in the reference CDM. As expected, a set of BPMs enables the generation of a more complete CDM with respect to the individual BPMs – no single BPM has enabled the generation of more than 46% of classes neither more than 20% of associations contained in the target CDM, while the set of BPMs enabled the generation of 100% of classes and 72% of associations contained in the target CDM.

Regarding the generation of classes, the results show that AMADEOS generated all classes that exist in the reference CDM, when the source reference set of BPMs was used, and also show that only main BPMs were sufficient to generate all classes (R=1 for the CDM derived from the set containing models R-01...R-07). Based on the auxiliary BPMs, additional classes were generated in comparison to the reference CDM, which constitute a surplus in this particular case.[24] These excessive classes, which were generated based on the auxiliary BPMs, do not increase the recall but reduce the precision because they represent a surplus in this particular case. The achieved precision in generating the classes is P=0.65 (based on the set of the main BPMs), and P=0.48 (based on the complete reference set). Since recall in both cases is R=1, the *F-score* for the class generation process is F=0.79 (based on the set of the main BPMs), and F=0.65 (based on the complete reference set).

[24] The analysis of excessive classes shows that these classes represent activation classes, which are not important in this case, because the observed system does not keep a history of changes in the state of individual objects, but only records their current states/values.

Table 2. Assessment of automatically generated CDMs (based on reference BPMN models) in comparison with reference CDM

Source model(s)	Assessment of classes							Assessment of associations						
	N_g	N_c	N_m	N_w	R	P	F	N_g	N_c	N_m	N_w	R	P	F
01	6	6	7	0	0.46	1.00	0.63	5	5	20	0	0.20	1.00	0.33
02	3	3	10	0	0.23	1.00	0.38	2	2	23	0	0.08	1.00	0.15
03	6	4	9	2	0.31	0.67	0.42	13	4	21	9	0.16	0.31	0.21
04	7	3	10	4	0.23	0.43	0.30	16	2	23	14	0.08	0.13	0.10
05	7	5	8	2	0.38	0.71	0.50	15	5	20	10	0.20	0.33	0.25
06	3	1	12	2	0.08	0.33	0.13	2	0	25	2	0.00	0.00	N/A
07	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
08	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
09	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
10	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
11	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
12	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
13	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
14	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
01-07	**20**	**13**	**0**	**7**	**1.00**	**0.65**	**0.79**	**55**	**18**	**7**	**37**	**0.72**	**0.33**	**0.45**
01-14	27	13	0	14	1.00	0.48	0.65	69	18	7	51	0.72	0.26	0.38

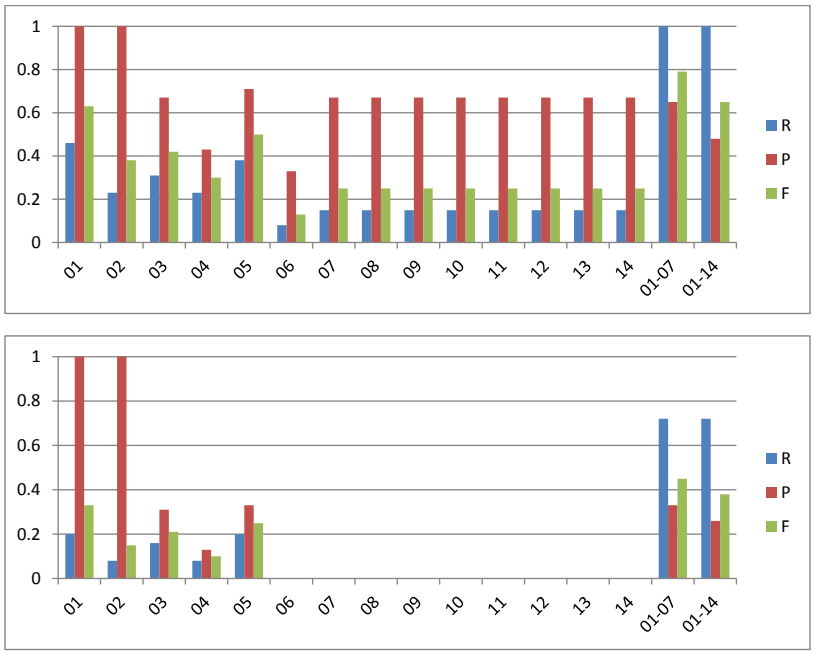


Fig. 14. Comparison of *R*, *P*, and *F* for generated classes (top) and associations (bottom) based on reference BPMN models in comparison with reference CDM

Regarding the generation of associations, the results show that AMADEOS generated 72% of associations that exist in the reference CDM, when the source reference set of BPMs was used, and also show that this recall can be achieved by using only main BPMs (R-01...R-07) as a basis for the generation. In this case, the auxiliary BPMs did not contribute to the recall increase, as well.

Additional analysis of the obtained results and reference set of BPMs showed that it would be possible to achieve a slightly higher recall than R=0.72, if some BPMs were further improved, but these additional improvements would not allow the generation of all associations between classes compared to the reference CDM. In other words, additional improvements to the reference BPMs cannot result in achieving R=1.00 in the associations generation process, because the tool generates associations between inappropriate classes in some situations – this refers to existing objects that are activated in one process and further used in another process. This problem does not exist when generating a CDM based on a single BPM, but it does occur when integrating partial CDMs that are generated based on a set of BPMs. This problem will be the subject of our future research, in order to increase the completeness of the automatically generated CDM.

The results show that a significant number of generated associations is excessive compared to the reference CDM, so the precision is relatively low (P=0.33 for the CDM derived from the set of the main BPMs, and P=0.26 for the CDM derived from the complete reference set). Given the relatively low precision, the effectiveness of the association generation process is F=0.45 (based on the main BPMs) and F=0.38 (based on the complete reference set). The analysis of excessively generated associations shows that the vast majority of these associations are not incorrectly generated (they have correct end multiplicities), but they do not exist in the reference CDM and therefore represent a surplus in this particular case. In other words, the given BPMs contain some process patterns having the semantic capacity for automatic generation of associations (as shown in the previous research), but these associations represent a surplus in this particular case. The problem of excessive associations (although correctly generated) has been identified in the previous research [20]. This is not a major problem when only one BPM is taken as a starting point, but with larger sets of BPMs the total number of excessively generated associations is significant, which may constitute a problem.

Since the case study showed that the main BPMs are sufficient to generate the target CDM, i.e. the auxiliary BPMs do not contribute to the increase of recall and precision, only main BPMs were further used in the experiments (described in the following subsections).

6.2. Experiment #1 (E-1)

Experiment design. According to Conway's law [28], the independent work of different designers will result in different models. Therefore, this experiment aimed to assess the effectiveness of the implemented approach in generating the CDM based on different real sets of BPMs, i.e. how real models affect the effectiveness of the generation process.

A total of 98 undergraduate students participated in E-1.[25] The experiment was conducted as a part of a mandatory course that includes business process modelling. All

[25] All participants in E-1 were students of the third year at the Software Engineering Department, at the Faculty of Electrical Engineering, University of Banja Luka.

participants firstly underwent the usual training (10 hours) for the BPMN-based business process modelling using the appropriate modelling tool (Eclipse/BPMN Modeler). Upon completion of the training, the task was to create a set of BPMN models based on the given textual specification of the main business processes in the Faculty Library. Given the scope of work and available time, all participants performed the task in pairs, which resulted in a total of 49 sets of the main BPMs of the Faculty Library. Given the previous education, as well as level of knowledge and skills in the BPMN-based business process modelling, all participants can be considered novice modelers.

After the analysis and evaluation of the created sets of the main BPMs[26], we have singled out 27 sets that were graded with a minimum of 70/100. These sets[27] were used as a starting point for the automatic CDM generation in the AMADEOS system. Based on each set, two CDMs were generated: (1) with applied "Advanced Composition" option[28] (case insensitivity and *LD*=1), and (2) without the "Advanced Composition" option[29] (case sensitivity and *LD*=0). The generated CDMs were then compared with the reference CDM. When evaluating the generated CDMs against the reference CDM, we used the same metrics and measures as in the case study.

Results. The comparison results of the automatically generated CDMs ("Advanced Composition" applied), with the reference CDM, are shown in Table 3. Each table row contains results for the CDM generated based on the corresponding set of the BPMN models (the first column contains the ID of the source set of the BPMN models, i.e. ID of the corresponding automatically generated CDM), while the other columns contain the corresponding metrics and measures for automatically generated classes and automatically generated associations, respectively. Figure 15 shows a comparison of R, P, and F for the generated classes (top) and associations (bottom) based on the students' sets of the BPMN models in comparison with the reference CDM.

Discussion. The experimental results largely confirm the results obtained in the case study.

Regarding the generation of classes, the obtained results confirm that a set of the main BPMs enables the generation of a very complete CDM because a very high average recall was achieved ($\overline{R}$=0.94) when using 27 real sets of the BPMN models ($\overline{R}$=1.00 in the case study). The achieved average precision for the class generation is $\overline{P}$=0.59 ($\overline{P}$=0.65 in the case study), and the average effectiveness of the class generation process is $\overline{F}$=0.72 ($\overline{F}$=0.79 in the case study).

Regarding the generation of associations, the average completeness is $\overline{R}$=0.57 ($\overline{R}$=0.72 in the case study) with the average precision $\overline{P}$=0.29 ($\overline{P}$=0.33 in the case study). This means that the average effectiveness of generating associations is $\overline{F}$=0.38 ($\overline{F}$=0.45 in the case study). The high matching of the achieved precision in the experiment ($\overline{P}$=0.29) and the case study ($\overline{P}$=0.33) confirms that the generator generates a significant number of excessive associations (relative to the reference CDM).

[26] The analysis and evaluation were conducted by the teachers of the course, who are also some of the coauthors of this article.

[27] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-1/BPMs*

[28] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-1/CDMs-AC*

[29] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-1/CDMs-NoAC*

Table 3. Assessment of automatically generated CDMs ("Advanced Composition" applied), based on students' sets of BPMN models, in comparison with reference CDM

Source set	Assessment of classes							Assessment of associations						
	N_g	N_c	N_m	N_w	R	P	F	N_g	N_c	N_m	N_w	R	P	F
01	16	12	1	4	0.92	0.75	0.83	49	14	11	35	0.56	0.29	0.38
02	20	12	1	8	0.92	0.60	0.73	53	11	14	42	0.44	0.21	0.28
03	18	13	0	5	1.00	0.72	0.84	43	17	8	26	0.68	0.40	0.50
04	16	12	1	4	0.92	0.75	0.83	47	16	9	31	0.64	0.34	0.44
05	19	13	0	6	1.00	0.68	0.81	41	17	8	24	0.68	0.41	0.52
06	20	12	1	8	0.92	0.60	0.73	50	14	11	37	0.56	0.27	0.37
07	19	12	1	7	0.92	0.63	0.75	40	14	11	26	0.56	0.35	0.43
08	20	11	2	9	0.85	0.55	0.67	57	15	10	42	0.60	0.26	0.37
09	24	13	0	11	1.00	0.54	0.70	67	18	7	49	0.72	0.27	0.39
10	27	13	0	14	1.00	0.48	0.65	62	12	13	50	0.48	0.19	0.28
11	13	13	0	10	1.00	0.57	0.72	39	16	9	23	0.64	0.41	0.50
12	18	13	0	5	1.00	0.72	0.84	46	16	9	30	0.64	0.35	0.45
13	31	13	0	18	1.00	0.42	0.59	69	14	11	55	0.56	0.20	0.30
14	18	12	1	6	0.92	0.67	0.77	54	17	8	37	0.68	0.31	0.43
15	20	12	1	8	0.92	0.60	0.73	36	14	11	22	0.56	0.39	0.46
16	21	13	0	8	1.00	0.62	0.76	52	16	9	36	0.64	0.31	0.41
17	27	13	0	14	1.00	0.48	0.65	60	15	10	45	0.60	0.25	0.35
18	24	13	0	11	1.00	0.54	0.70	69	13	12	56	0.52	0.19	0.28
19	23	12	1	11	0.92	0.52	0.67	55	15	10	40	0.60	0.27	0.38
20	28	12	1	16	0.92	0.43	0.59	76	12	13	64	0.48	0.16	0.24
21	21	11	2	10	0.85	0.52	0.65	50	13	12	37	0.52	0.26	0.35
22	19	12	1	7	0.92	0.63	0.75	34	11	14	23	0.44	0.32	0.37
23	24	12	1	12	0.92	0.50	0.65	50	13	12	37	0.52	0.26	0.35
24	23	11	2	12	0.85	0.48	0.61	58	12	13	46	0.48	0.21	0.29
25	20	12	1	8	0.92	0.60	0.73	42	14	11	28	0.56	0.33	0.42
26	21	12	1	9	0.92	0.57	0.71	50	13	12	37	0.52	0.26	0.35
27	19	12	1	7	0.92	0.63	0.75	37	14	11	23	0.56	0.38	0.45
Mean	**21.4**	**12.3**	**0.7**	**9.1**	**0.94**	**0.59**	**0.72**	**51.4**	**14.3**	**10.7**	**37.1**	**0.57**	**0.29**	**0.38**

Based on the results, we can conclude that the sets of the main BPMs, although they were created by novice modelers, represent a very good starting point for generating the target CDM, because the average completeness of the generated CDM is 94% for classes and 57% for associations. This further implies that we can expect that "each" real set of the main BPMs, created by experienced modelers, will enable the automatic generation of a more complete CDM in comparison to the recall achieved in E-1.

Impact of accidental errors on the process effectiveness. Regarding the impact of accidental errors in the source BPMs on the effectiveness of the CDM generation process, the analysis shows that the applied option "Advanced Composition" allows to reduce the total number of generated concepts – for all 27 sets the same or fewer concepts is

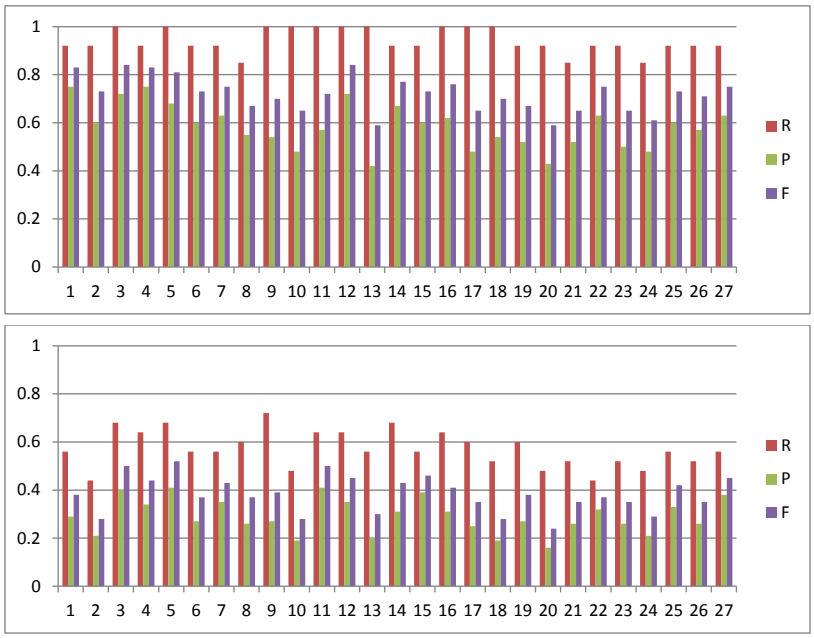


Fig. 15. Comparison of *R*, *P*, and *F* for generated classes (top) and associations (bottom) based on students' sets of BPMN models in comparison with reference CDM

generated than when the option "Advanced Composition" is not applied. The analysis of the generated CDMs shows that in 44% (12/27) cases a reduction in the number of generated concepts was achieved, while in 56% (15/27) cases identical models were obtained. The analysis shows that the differences in the generated models are mainly related to the generated classes – in all 12/27 cases, a smaller number of classes was generated, while only in 2/27 (7%) cases a smaller number of associations was generated.

The analysis shows that the reduced number of the generated concepts did not reduce the completeness of the generated CDM compared to the reference CDM in any case (average recall is the same regardless of whether the "Advanced Composition" option is applied). A smaller number of generated concepts reduces redundancy and increases precision (for classes: $\overline{P}_{AC}$=0.5857, $\overline{P}_{NoAC}$=0.5704; for associations: $\overline{P}_{AC}$=0.2910, $\overline{P}_{NoAC}$=0.2906). Increasing the precision with the same recall resulted in higher effectiveness (for classes: $\overline{F}_{AC}$=0.7183, $\overline{F}_{NoAC}$=0.7068; for associations: $\overline{F}_{AC}$=0.3822, $\overline{F}_{NoAC}$=0.3818). For the purpose of illustration, Fig. 16 shows differences between *P* values for the generated classes, based on the students' sets of the BPMN models in comparison with the reference CDM, in case that "Advanced Composition" is applied (AC) or not applied (NoAC).

To conclude, applied "Advanced Composition" reduces impact of the accidental errors and inconsistencies in the source BPMs on the effectiveness of the CDM generation process.

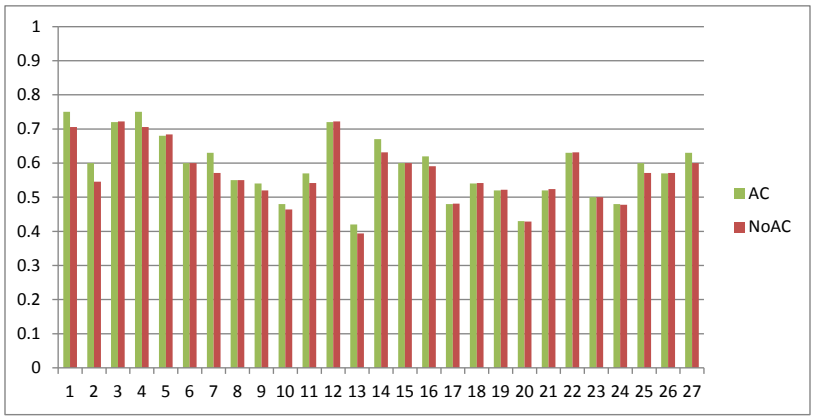


Fig. 16. Comparison of *P* values for generated classes: (AC) – "Advanced Composition" applied, (NoAC) – "Advanced Composition" not applied

6.3. Experiment #2 (E-2)

Experiment design. Considering Conway's law [28], E-2 was conducted in order to assess how much the CDM, which is automatically generated based on the reference set of the main BPMs, matches real CDMs that are manually designed for the same business system, based on the same textual specifications of business processes.

A total of 18 graduate students participated in E-2.[30] The experiment was conducted as a part of an elective course that includes advanced database design techniques. All participants had the necessary knowledge and skills for manual database design, which they acquired as undergraduate students, so they did not have additional training. The task was to manually design the CDM based on the given textual specification of the main business processes in the Faculty Library (the same specification was given to the participants in E-1 for creating the BPMN models). The participants solved the task in pairs. In the end we had a total of nine manually designed CDMs for the Faculty Library.[31] All manually designed CDMs were analyzed by the teachers and evaluated as adequate.[32] Each manually designed CDM was then compared with the CDM automatically generated based on the reference set of the main BPMs, and the same metrics and measures were used in the evaluation as in the case study and in E-1.

Results. The results of comparing the CDM, which was automatically generated based on the reference set of the main BPMs, with the manually designed CDMs are shown in Table 4. Each table row contains results for the corresponding manually designed CDM – the first column contains the ID of the manually designed CDM, while the other columns contain the corresponding metrics and measures for automatically

[30] All participants in E-2 were master students at the Software Engineering Department, at the Faculty of Electrical Engineering, University of Banja Luka.

[31] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-2/CDMs*

[32] The analysis and assessment were conducted by the teachers of the course, who are also some of the coauthors of this article.

generated classes and automatically generated associations, respectively. The *N*-labeled column contains a number of the corresponding concepts in the real CDM. Figure 17 shows a comparison of R, P, and F for the classes and associations in the automatically generated CDMs (based on the reference set of the main BPMs) in comparison with the manually designed CDMs.

Table 4. Assessment of automatically generated CDM (based on reference set of main BPMs) in comparison with manually designed CDMs

CDM ID	Assessment of classes								Assessment of associations							
	N	N_g	N_c	N_m	N_w	R	P	F	N	N_g	N_c	N_m	N_w	R	P	F
01	10	20	10	0	10	1.00	0.50	0.67	12	55	9	3	46	0.75	0.16	0.27
02	13	20	12	1	8	0.92	0.60	0.73	21	55	10	11	45	0.48	0.18	0.26
03	10	20	10	0	10	1.00	0.50	0.67	21	55	16	5	39	0.76	0.29	0.42
04	15	20	13	2	7	0.87	0.65	0.74	20	55	12	8	44	0.60	0.22	0.32
05	16	20	15	1	5	0.94	0.75	0.83	20	55	13	7	42	0.65	0.24	0.35
06	16	20	15	1	5	0.94	0.75	0.83	24	55	12	12	43	0.50	0.22	0.30
07	9	20	8	1	12	0.89	0.40	0.55	8	55	5	3	50	0.63	0.09	0.16
08	10	20	10	0	10	1.00	0.50	0.67	9	55	7	2	48	0.78	0.13	0.22
09	18	20	18	0	2	1.00	0.90	0.95	35	55	23	12	32	0.66	0.42	0.51
Mean	**13**	**20**	**12.3**	**0.7**	**7.7**	**0.95**	**0.62**	**0.74**	**19**	**55**	**11.9**	**7.1**	**43.1**	**0.64**	**0.22**	**0.31**

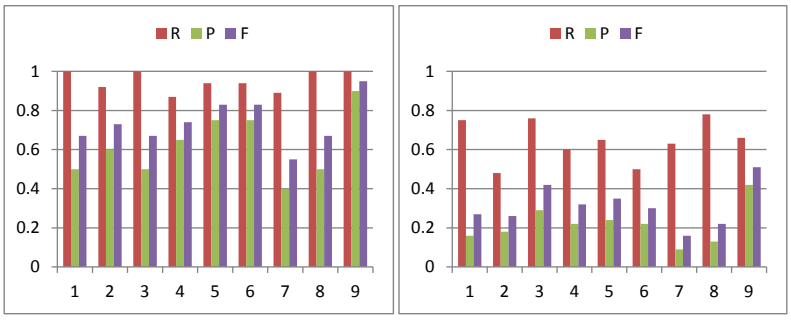


Fig. 17. Comparison of *R*, *P*, *F* for generated classes (left) and associations (right) in automatically generated CDMs (based on reference set of main BPMs) in comparison with manually designed CDMs

Discussion. Both the experimental results achieved in E-2, as well as in E-1, largely confirm the results obtained in the case study.

Regarding the generation of classes, the obtained results confirm that the set of the main BPMs enables the generation of a very complete CDM, because compared to nine real CDMs designed based on the same text specification, a very high average recall ($\overline{R}$=0.95) was achieved ($\overline{R}$=1.00 in the case study). The achieved average precision for the generation of classes is $\overline{P}$=0.62 ($\overline{P}$=0.65 in the case study), and the average effectiveness of the class generation process is $\overline{F}$=0.74 ($\overline{F}$=0.79 in the case study).

Regarding the generation of associations, the average completeness of the generated CDM comparing to the manually designed CDMs is $\overline{R}$=0.64 ($\overline{R}$=0.72 in the case study), with an average precision $\overline{P}$=0.22 ($\overline{P}$=0.33 in the case study), so the average effectiveness of the association generation process is $\overline{F}$=0.31 ($\overline{F}$=0.45 in the case study). The relatively low precision ($\overline{P}$=0.22) confirms that the generator generates a significant number of redundant associations (compared to the manually designed CDMs).

Based on the results, it can be concluded that the reference set of the main BPMs is a very good starting point for generating the target CDM, because the average completeness of the generated CDM is 95% for classes and 64% for associations, compared to the manually designed CDMs.

Finally, given the results obtained in E-1 and E-2, we can conclude that "each" real set of the main BPMs, created by experienced modelers, will constitute a very good starting base for the automatic generation of the target CDM.

6.4. Experiment #3 (E-3)

Experiment design. An earlier experiment [20] with database practitioners as participants, showed that an automatically generated CDM provides a solid basis for designing the target CDM, because it speeds up the design process compared to designing from scratch. Therefore, E-3 was conducted, which aimed to assess how much the CDM, which is automatically generated based on the reference set of the main BPMs, is a good starting point for designing the target CDM, instead of designing the target CDM from scratch.

The participants in E-3 were mostly the students who also participated in E-2. The task in E-3 was to use the CDM, which was generated based on the reference set of the main BPMs of the Faculty Library, as a starting point for designing the target CDM of the Library. The participants were able to discard excessive concepts, correct incorrectly generated concepts and/or add missing concepts in the automatically generated CDM. Since all participants already had the appropriate domain knowledge acquired in E-2 and had already designed a CDM from scratch, in E-3 each participant individually completed the task, which resulted in 12 CDMs[33] of the Library that are designed based on the CDM that was automatically generated based on the reference set of the main BPMs. All resulting CDMs were analyzed by teachers and graded as adequate.[34] Afterwards, each manually designed CDM was compared with the initial automatically generated CDM, and the same metrics and measures were used in the evaluation as in the other experiments.

Results. The results of comparing the CDMs, which were manually designed based on the automatically generated CDM, with the CDM that was automatically generated based on the reference set of the main BPMs, are shown in Table 5. Each table row contains results for the corresponding manually designed CDM – the first column contains the ID of the manually designed CDM, while the other columns contain the corresponding metrics and measures for automatically generated classes and automatically generated associations, respectively. The *N*-labeled column contains a number of the corresponding

[33] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-3/CDMs*

[34] Analysis and evaluation were conducted by the same teachers as in E-2.

concepts in the manually designed CDM. Figure 18 shows a comparison of R, P, and F for the classes and associations in the automatically generated CDMs (based on the reference set of the main BPMs) in comparison with the manually designed CDMs (based on the automatically generated CDM).

Table 5. Assessment of automatically generated CDM (based on reference set of main BPMs) in comparison with manually designed CDMs (based on automatically generated CDM)

CDM ID	Assessment of classes								Assessment of associations							
	N	N_g	N_c	N_m	N_w	R	P	F	N	N_g	N_c	N_m	N_w	R	P	F
01	15	20	13	2	7	0.87	0.65	0.74	22	55	14	8	41	0.64	0.25	0.36
02	10	20	10	0	10	1.00	0.50	0.67	14	55	9	5	46	0.64	0.16	0.26
03	13	20	13	0	7	1.00	0.65	0.79	34	55	31	3	24	0.91	0.56	0.70
04	17	20	17	0	3	1.00	0.85	0.92	25	55	22	3	33	0.88	0.40	0.55
05	15	20	15	0	5	1.00	0.75	0.86	36	55	36	0	19	1.00	0.65	0.79
06	18	20	18	0	2	1.00	0.90	0.95	36	55	34	2	21	0.94	0.62	0.75
07	17	20	17	0	3	1.00	0.85	0.92	24	55	22	2	33	0.92	0.40	0.56
08	12	20	10	2	10	0.83	0.50	0.63	12	55	8	4	47	0.67	0.15	0.24
09	9	20	9	0	11	1.00	0.45	0.62	11	55	11	0	44	1.00	0.20	0.33
10	18	20	18	0	2	1.00	0.90	0.95	26	55	17	9	38	0.65	0.31	0.42
11	18	20	18	0	2	1.00	0.90	0.95	33	55	24	9	31	0.73	0.44	0.55
12	10	20	10	0	10	1.00	0.50	0.67	12	55	8	4	47	0.67	0.15	0.24
Mean	**14.3**	**20**	**14**	**0.3**	**6**	**0.98**	**0.70**	**0.80**	**23.8**	**55**	**19.7**	**4.1**	**34.9**	**0.80**	**0.36**	**0.48**

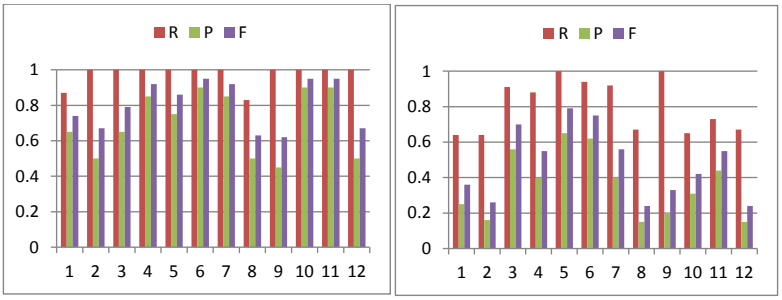


Fig. 18. Comparison of *R*, *P*, *F* for generated classes (left) and associations (right) in automatically generated CDMs (based on reference set of main BPMs) in comparison with manually designed CDMs (based on automatically generated CDM)

Discussion. The results obtained in E-3 confirm the previously obtained results in the experiment [20] with database professionals as participants. The results show that the CDM, which is automatically generated based on the set of the main BPMs, provides a good basis for manually designing the target CDM, rather than designing the target CDM from scratch.

Regarding the generation of classes, the obtained results confirm that the CDM, which is automatically generated based on the set of the main BPMs, enables the generation of a very complete CDM, because with 12 CDMs designed based on the automatically generated CDM, very high average recall ($\overline{R}$=0.98) is achieved ($\overline{R}$=1.00 in the case study). High average recall and high average precision $\overline{P}$=0.70 ($\overline{P}$=0.65 in the case study), resulted in high effectiveness ($\overline{F}$=0.80) of the class generation process ($\overline{F}$=0.79 in the case study).

Regarding the generation of associations, the average recall for the generated CDM in comparison to the manually designed CDMs is $\overline{R}$=0.80 ($\overline{R}$=0.72 in the case study), while the average precision is $\overline{P}$=0.36 ($\overline{P}$=0.33 in the case study). This means that the average effectiveness of generating associations is $\overline{F}$=0.48 ($\overline{F}$=0.45 in the case study). The results in E-3 also confirm that the generator generates a significant number of redundant associations (average precision for generating associations is $\overline{P}$=0.36).

The results obtained in E-3 are better than the results obtained in both the case study and in previous experiments, which was expected – the participants in E-3 had the automatically generated CDM as a starting point, so design was reduced mainly to discarding surplus from the initial CDM. Also, the designed models contain more concepts than CDMs that were manually designed from scratch – the average number of classes in CDMs that were manually designed from scratch in E-2 is 13, and in CDMs that were manually designed based on the automatically generated CDM is 14.3; while the average number of associations is 19 and 23.8, respectively.

6.5. Summative Evaluation Results

In order to achieve a better insight into the obtained results, we aggregated and compared the main results achieved in the case study and performed experiments. The summative evaluation results are presented in Table 6, where each row contains the corresponding R, P, and F values for the automatically generated classes and associations in a particular evaluation round. For the case study-based evaluation, the corresponding row contains values achieved by comparing the CDM derived from the reference set of the main BPMs with the reference CDM. For the experiments, each row contains average values achieved in the corresponding experiment. Finally, the bottom rows contain average values and the corresponding standard deviation (σ). The summative results are also shown in Fig. 19.

The summative overview of the results shows that the experiments largely confirmed the results obtained in the case study.

Regarding the generation of classes, the obtained results show that the implemented approach enables the generation of complete or almost complete CDM ($\overline{R}$=0.97, σ=0.03) with average precision $\overline{P}$=0.64 (σ=0.05), which gives the effectiveness of the process of generating classes $\overline{F}$=0.76 (σ=0.04).

Regarding the generation of associations, the obtained results show that the implemented approach enables the generation of an average of 68% of associations ($\overline{R}$=0.68, σ=0.10) with average precision $\overline{P}$=0.30 (σ=0.06), which gives the effectiveness of the process of generating associations $\overline{F}$=0.41 (σ=0.08).

The summative results show a relatively low precision ($\overline{P}$=0.30, σ=0.06) in generating associations, i.e. the tool generates a significant number of surplus associations in comparison to the target CDM.

Table 6. Summative evaluation results

Evaluation round	Classes			Associations		
	R	P	F	R	P	F
CS (CDM derived from reference set of main BPMs ↔ reference CDM)	1.00	0.65	0.79	0.72	0.33	0.45
E-1 (CDMs derived from students' sets of main BPMs ↔ reference CDM)	0.94	0.59	0.72	0.57	0.29	0.38
E-2 (CDM derived from reference set of main BPMs ↔ manually designed CDMs from scratch)	0.95	0.62	0.74	0.64	0.22	0.31
E-3 (CDM derived from reference set of main BPMs ↔ manually designed CDMs based on generated CDM)	0.98	0.70	0.80	0.80	0.36	0.48
Mean	**0.97**	**0.64**	**0.76**	**0.68**	**0.30**	**0.41**
Standard Deviation (σ)	**0.03**	**0.05**	**0.04**	**0.10**	**0.06**	**0.08**

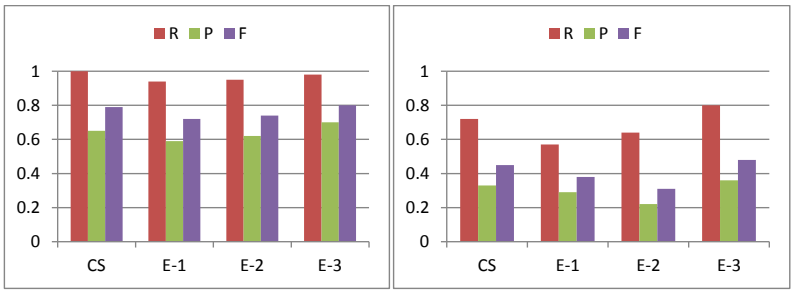


Fig. 19. Comparison of R, P, F for generated classes (left) and associations (right) achieved in case study-based evaluation and performed experiments

Comparison with results that could be obtained by applying other POM-based tools. For the same reference source set of BPMN models, the pre-existing version of the AMADEOS system is able to generate up to 46% of all classes, and up to 20% of all associations of the target reference CDM, since it is able to derive the CDM from only one single BPM from the source set.

Regarding the other tools taking BPMN models as a starting base for the CDM generation, we are only able to estimate[35] the completeness that could be achieved by their application to the same set of models. Since all these tools are able to derive the CDM from only one single BPM, like pre-existing AMADEOS, we conclude that they could not achieve higher completeness than pre-existing AMADEOS, since they implement a less complete set of transformation rules.

[35] As noted in Sect. 2, other tools are not publicly available, and we are not able to evaluate them.

6.6. Threats to Validity

In this section we consider possible threats to the validity of the experiments and derived conclusions, which may lie in the used models and the way the experiments were conducted.

Concerning the threats related to the source models, we have conducted E-1 in order to assess the approach effectiveness based on different real sets of BPMs, and not to draw conclusions based on a single set used in the case study. That is why we conducted the experiment with a large number of participants, which resulted in a relatively large number of sets for the same problem domain. In this way, we eliminated the threat that the initial (reference) set is predisposed to achieve maximum effectiveness.

Someone might consider that work in pairs of the participants in E-1 enabled better source BPMs. Since it is about novice modelers, it can be considered that we have just got relevant sets to be prepared by more experienced modelers and that the work of novice modelers in pairs is not a real threat. Working in pairs has additionally enabled us to analyze the appearance of accidental errors in the source sets, as well as possibilities for their elimination.

As for E-1, the only real threat may be that the source sets of BPMs were created in uncontrolled conditions to some extent (the participants created models as homework), so all sets might not be original enough. The performed analysis of the source sets found some plagiarisms and such duplicates were eliminated and not further used in the experiment. If one takes a look at the results achieved in E-1, it can be concluded that there are no identical results, which means there are no identical sets.

Regarding the threats related to the target models, we have conducted E-2 in order to assess to what extent the automatically generated CDM matches real CDMs manually designed for the same business system, and not to draw conclusions based on the one (reference) CDM used in the case study. Therefore, we conducted the experiment with a relatively large number of participants, which resulted in a larger number of target models. In this way, we eliminated the threat that the initial source set is predisposed to achieve maximum effectiveness in generating the CDM in comparison with one concrete or typical CDM in the given problem domain.

In addition to the manually designed CDMs from scratch, in E-3 we also used the CDMs that were manually designed on the basis of the automatically generated CDM. Although it was expected that more complete target models would be obtained at the time, some of the models with a very high recall for associations (R=1.00 for two models) can be suspected. The main reason that could have led to such high completeness is the relatively complex automatically generated CDM which was relatively difficult to edit in a web browser. This is confirmed by the fact that some participants in E-3 first exported the automatically generated CDM, then imported and edited it in some other tool. [36].

Although we used realistic sets of BPMs in the evaluation, it is nevertheless about relatively small sets that were created by small teams (pairs). Therefore, it is possible that, on the sets containing a bigger number of BPMs and created by more numerous teams, there could be a higher number of merging conflicts (e.g. synonyms) that could not be eliminated by simple techniques for composing partial CDMs. In such situations, we would have an additional decrease in precision and effectiveness.

[36] Note: See the CDMs designed in E-3.

The very important issue is related to the comparison of the automatically generated CDMs with the reference CDM, since the evaluation results are dependent on the comparison outcomes. Here we would like to emphasize that all the comparisons were performed manually by three evaluators, and all outcomes were achieved by their consensus.

6.7. Lessons Learned

This section presents some lessons we learned from the entire evaluation.

The results show that AMADEOS generates a significant number of surplus associations. This reduces the precision and effectiveness of the generation process and creates a feeling of dissatisfaction with designers (such an opinion was expressed by the majority of the participants in E-3) because they should manually eliminate surplus associations. Desirable tool improvement would be to enable the user/designer to influence the generation process by choosing which types of associations (s)he wants or does not want in the target model. Additional interviews with the E-3 participants showed that the tool does not provide appropriate comfort in editing complex models, so these AMADEOS' functionalities should be additionally improved in order to make the work easier for users.

The sets of BPMs in E-1 were prepared by pairs. The results show that inconsistencies happen in 44% of cases (12/27 sets were characterized by some kind of inconsistency, which resulted in redundant concepts in the automatically generated CDMs if the "Advanced Composition" option was applied during the generation). Potentially, the level of inconsistencies of the source sets could be higher in the case of larger project teams and larger sets of BPMs. Since AMADEOS currently allows the elimination of single typographical errors (LD=1), the possible AMADEOS' improvement could be increasing the level of user influence on elimination of inconsistencies so that the user would be able to set the value for LD.

6.8. Approach and Tool Limitations

In this section we discuss some limitations of the approach and implemented tool.

Although the approach enables the generation of a CDM structure with a fairly high level of completeness, the given set of transformation rules are not sufficient to enable automatic generation of the complete CDM. Some associations are still missing, as well as generalization relationships. The approach further enables just to automatically generate a CDM structure but does not enable automatic generation of attributes in the classes (except simple primary keys). These particular approach deficiencies were not in the primary focus of this article, but they constitute our main challenges and long-term research goals.

Although AMADEOS supports different source notations (BPMN and UML AD), and different serialization formats (XMI and XSD), currently all models in the entire source set must be represented by the same notation, and also serialized in the same format.

Although this particular research and conducted experiments did not face problems in merging partial CDMs, AMADEOS currently deploys very simple techniques that enable just to overcome inconsistencies related to different naming notations (case sensitivity) and accidental typing errors (Levenshtein distance) and therefore is not able to solve other types of merging conflicts which could appear.

Although the approach enables automatic generation of some kinds of associations that may be useful in some cases, the performed experiments showed that the tool generates a significant number of redundant associations that make it difficult to deal with the automatically generated CDM.

Although AMADEOS enables users to automatically generate the initial CDM, the forward database engineering based on the generated CDM is not possible, but users can only export the generated CDM and further use it in some other database design tool.

7. Conclusions

In this article we presented an approach to automatic CDM derivation from a set of BPMs, and the corresponding tool that implements the proposed approach. The approach proposes the incremental synthesis of the target model by iteratively composing the partial CDMs that are derived from the models contained in the source set. The implemented AMADEOS tool is the first online web-based tool that publicly enables automatic CDM derivation from a set of BPMs that may be represented by two different notations (BPMN or UML AD).

The approach effectiveness was evaluated in a case study and a series of experiments. As expected, the evaluation results confirmed that a set of BPMs enables the generation of a more complete CDM relative to the individual BPMs contained in the given set. The results show that the main BPMs constitute a sufficient basis to derive the target CDM since the auxiliary BPMs do not increase the recall achieved by deriving the CDM from the set of the main BPMs.

The summative results show that the implemented approach allows the generation of complete or almost complete CDM when it comes to classes (average completeness of generated models is 97%), which with an average precision of 64% gives average effectiveness of 76% of the class generation process. Regarding the generation of associations, the obtained results show that the implemented approach enables the generation of an average of 68% of the total number of associations, which with an average precision of 30%, gives average effectiveness of 41%. Analyzes show that the precision of generating associations is relatively low because the tool generates surplus associations (which are generally not incorrectly generated).

Future work will focus on further improving the approach and AMADEOS system, in line with the long-term research goals, the lessons learned in the experiments conducted, and the stated limitations. Future work will include: further identification of the semantic capacity of BPMs to increase the effectiveness of the CDM synthesis process, further tool improvements to provide users with greater comfort in working with complex models, and adding functionalities for the forward database engineering based on the generated CDM. Our intention is also to evaluate the approach with more complex real collections of BPMs.

Acknowledgments. The research is partially supported by Ministry for Scientific and Technological Development, Higher Education and Information Society of the Republic of Srpska, through Project no. 19.032/961-114/19: "*AMADEOS: Automatic derivation of conceptual database models from a collection of business process models*".

References

1. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: An MDA approach for goal-oriented requirement analysis in web engineering. Journal of Universal Computer Science 16(17), 2475–2494 (2010)
2. Alencar, F., Marín, B., Giachetti, G., Pastor, O., Pimentel, J.H.: From i* Requirements Models to Conceptual Models of a Model Driven Development Process. In: POEM 2009, LNBIP, vol. 39, pp. 99–114. Springer (2009)
3. Alencar, F., Pedroza, F., Castro, J., Amorim, R.: New mechanisms for the integration of organizational requirements and object oriented modeling. In: Proc. of WER 2003. pp. 109–123 (2003)
4. Alencar, F.M.R., Filho, G.A.C., Castro, J.F.: Support for Structuring Mechanism in the Integration of Organizational Requirements and Object Oriented Modeling. In: Proc. of WER 2002. pp. 147–161 (2002)
5. Alencar, F.M.R, Pedroza, F.P., Castro, J., Silva, C.T.L., Ramos, R.A.: XGOOD: A tool to automatize the mapping rules between i* framework and UML. In: Proc. of CIbSE 2006. pp. 125–138 (2006)
6. Ang, C.L., Khoo, L.P., Gay, R.K.L.: IDEF*: a comprehensive modelling methodology for the development of manufacturing enterprise systems. Int. Journal of Production Research 37(17), 3839–3858 (1999)
7. Banjac, D., Brdjanin, D., Banjac, G., Maric, S.: Evaluation of Automatically Generated Conceptual Database Model Based on Collaborative Business Process Model: Controlled Experiment. In: ICT Innovations 2016, AISC, vol. 665, pp. 134–145. Springer (2016)
8. Batini, C., Lenzerini, M., Navathe, S.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. 18(4), 323–364 (1986)
9. Becker, L.B., Pereira, C.E., Dias, O.P., Teixeira, I.M., Teixeira, J.P.: MOSYS: A methodology for automatic object identification from system specification. In: Proc. of ISORC 2000. pp. 198–201. IEEE Computer Society (2000)
10. Bloomfield, T.: MDA, meta-modelling and model transformation: Introducing new technology into the defence industry. In: ECMDA-FA 2005, LNCS, vol. 3748, pp. 9–18. Springer (2005)
11. Boccalatte, A., Giglio, D., Paolucci, M.: ISYDES: the project of a tool aimed at information system development. In: Proc. of AIWORC 2000. pp. 293–298. IEEE (2000)
12. Brambilla, M., Cabot, J., Comai, S.: Automatic Generation of Workflow-Extended Domain Models. In: MoDELS 2007, LNCS, vol. 4735, pp. 375–389. Springer (2007)
13. Brambilla, M., Cabot, J., Comai, S.: Extending Conceptual Schemas with Business Process Information. Advances in Software Engineering, vol. 2010, Article ID 525121 (2010)
14. Brdjanin, D., Ilic, S., Banjac, G., Banjac, D., Maric, S.: Automatic derivation of conceptual database models from differently serialized business process models. Software and Systems Modeling 20(1), 89–115 (2021)
15. Brdjanin, D., Maric, S.: Towards the initial conceptual database model through the UML metamodel transformations. In: Proc. of Eurocon 2011. pp. 1–4. IEEE (2011)
16. Brdjanin, D., Maric, S.: An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram. Computer Science and Information Systems 9(1), 249–283 (2012)
17. Brdjanin, D., Maric, S.: Model-driven Techniques for Data Model Synthesis. Electronics 17(2), 130–136 (2013)
18. Brdjanin, D., Maric, S., Gunjic, D.: ADBdesign: An approach to automated initial conceptual database design based on business activity diagrams. In: ADBIS 2010, LNCS, vol. 6295, pp. 117–131. Springer (2010)
19. Brdjanin, D., Banjac, D., Banjac, G., Maric, S.: Automated two-phase business model-driven synthesis of conceptual database models. Computer Science and Information Systems 16(2), 657–688 (2019)

20. Brdjanin, D., Banjac, G., Banjac, D., Maric, S.: An experiment in model-driven conceptual database design. Software & Systems Modeling 18(3), 1859–1883 (Jun 2019)
21. Brdjanin, D., Banjac, D., Banjac, G., Maric, S.: An Approach to Automated Two-phase Business Model-driven Synthesis of Data Models. In: Model and Data Engineering, LNCS, vol. 10563, pp. 57–70. Springer (2017)
22. Brdjanin, D., Banjac, G., Banjac, D., Maric, S.: Controlled Experiment in Business Model-driven Conceptual Database Design. In: Enterprise, Business-Process and Information Systems Modeling, LNBIP, vol. 287, pp. 289–304. Springer (2017)
23. Brdjanin, D., Banjac, G., Maric, S.: Automated Synthesis of Initial Conceptual Database Model Based on Collaborative Business Process Model. In: ICT Innovations 2014: World of Data, AISC, vol. 311, pp. 145–156. Springer (2015)
24. Brdjanin, D., Maric, S.: On Automated Generation of Associations in Conceptual Database Model. In: ER Workshops 2011, LNCS, vol. 6999, pp. 292–301. Springer (2011)
25. Brdjanin, D., Maric, S.: Towards the Automated Business Model-Driven Conceptual Database Design. In: Advances in Databases and Information Systems, AISC, vol. 186, pp. 31–43. Springer (2012)
26. Brdjanin, D., Vukotic, A., Banjac, G., Banjac, D., Maric, S.: Automatic Derivation of Conceptual Database Model from a Set of Business Process Models. In: 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). pp. 1–8. IEEE (2020)
27. Castro, J.F., Alencar, F.M.R., Filho, G.A.C., Mylopoulos, J.: Integrating organizational requirements and object oriented modeling. In: Proc. of ISRE 2001. pp. 146–153. IEEE (2001)
28. Conway, M.: How do committees invent? Datamation 14(4), 28–31 (1968)
29. Cruz, E., Machado, R., Santos, M.: Deriving a Data Model from a Set of Interrelated Business Process Models. In: Proc. of ICEIS 2015. pp. 49–59 (2015)
30. Cruz, E.F., Machado, R.J., Santos, M.Y.: From Business Process Modeling to Data Model: A systematic approach. In: Proc. of QUATIC 2012. pp. 205–210. IEEE (2012)
31. Cruz, E.F., Machado, R.J., Santos, M.Y.: On the Rim Between Business Processes and Software Systems. In: Cruz, A.M.R., Cruz, M.E.F. (eds.) New Perspectives on Information Systems Modeling and Design, pp. 170–196 (2019)
32. de la Vara, J.L.: Business process-based requirements specification and object-oriented conceptual modelling of information systems. PhD Thesis, Valencia Polytechnic Uni. (2011)
33. Dominguez, E., Pérez, B., Rubio, A., Zapata, M.A., Allué, A., López, A.: Generating persistence structures for the integration of data and control aspects in business process monitoring. In: Proc. of the 20th Int. Conf. on Enterprise Information Systems – Vol. 2: ICEIS. pp. 320–327. SciTePress (2018)
34. Drozdova, M., Kardos, M., Kurillova, Z., Bucko, B.: Transformation in Model Driven Architecture. In: Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part I. pp. 193–203. Springer, Cham (2016)
35. Drozdová, M., Mokryš, M., Kardoš, M., Kurillová, Z., Papán, J.: Change of Paradigm for Development of Software Support for eLearning. In: Proc. of ICETA 2012. pp. 81–84. IEEE (2012)
36. Dujlovic, I., Obradovic, N., Kelec, A., Brdjanin, D., Banjac, G., Banjac, D.: An Approach to Web-based Visualization of Automatically Generated Data Models. In: IEEE EUROCON 2019 – 18th International Conference on Smart Technologies. pp. 1–6. IEEE (2019)
37. España, S.: Methodological integration of communication analysis into a model-driven software development framework. PhD Thesis, Valencia Polytechnic Uni. (2011)
38. Essebaa, I., Chantit, S.: Toward an automatic approach to get PIM level from CIM level using QVT rules. In: 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA). pp. 1–6. Mohammedia (2016)
39. Fernandes, J.M., Lilius, J., Truscan, D.: Integration of DFDs into a UML-based model-driven engineering approach. Software and Systems Modeling 5(4), 403–428 (2006)

40. Fouad, A.: Embedding Requirements within the Model Driven Architecture. PhD Thesis, Bournemouth Uni. (2011)
41. Insfran, E., Pastor, O., Wieringa, R.: Requirements Engineering-Based Conceptual Modelling. Requirements Engineering 7(2), 61–72 (2002)
42. Insfran, E.: Requirements engineering approach for object-oriented conceptual modeling. PhD Thesis, Valencia Polytechnic Uni. (2003)
43. Jiang, L., Topaloglou, T., Borgida, A., Mylopoulos, J.: Goal-oriented conceptual database design. In: Proc. of RE '07. pp. 195–204. IEEE, Los Alamitos, USA (2007)
44. Jouault, F., Allilaire, F., Bezivin, J., Kurtev, I.: ATL: A model transformation tool. Science of Computer Programming 72(1-2), 31–39 (2008)
45. Khlif, W., Elleuch, N., Alotabi, E., Ben-Abdallah, H.: Designing BP-IS Aligned Models: An MDA-based Transformation Methodology. In: Proc. of the 13th Int. Conf. on Evaluation of Novel Approaches to Software Engineering – ENASE 2018. pp. 258–266 (2018)
46. Koch, N.: Transformation Techniques in the Model-Driven Development Process of UWE. In: Proc. of the Workshops at ICWE'06, Art. No. 3. ACM (2006)
47. Koch, N., Zhang, G., Escalona, M.J.: Model Transformations from Requirements to Web System Design. In: Proc. of ICWE'06. pp. 281–288. ACM (2006)
48. Koskinen, J., Peltonen, J., Selonen, P., Systa, T., Koskimies, K.: Model processing tools in UML. In: Proc. of ICSE 2001. pp. 819–820. IEEE Computer Society (2001)
49. Kriouile, A., Addamssiri, N., Gadi, T.: An MDA Method for Automatic Transformation of Models from CIM to PIM. American Journal of Software Engineering and Applications 4(1), 1–14 (2015)
50. Levenshtein, I.V.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
51. Lingzhi, L., Ang, C.L., Gay, R.K.L.: Integration of Information Model (IDEF1) with Function Model (IDEF0) for CIM Information System Design. Expert Systems with Applications 10(3/4), 373–380 (1996)
52. Liu, D., Subramaniam, K., Far, B., Eberlein, A.: Automating Transition from Use-cases to Class Model. In: Proc. of CCECE 2003. pp. 831–834. IEEE (2003)
53. Martinez Rebollar, A.: Conceptual Schemas Generation from Organizational Models in an Automatic Software Production Process. PhD Thesis, Valencia Polytechnic Uni. (2008)
54. Nikiforova, O., Gusarovs, K., Gorbiks, O., Pavlova, N.: BrainTool: A tool for generation of the UML class diagrams. In: Proc. of ICSEA 2012. pp. 60–69. IARIA (2012)
55. Nikiforova, O., Gusarovs, K., Gorbiks, O., Pavlova, N.: Improvement of the Two-Hemisphere Model-Driven Approach for Generation of the UML Class Diagram. Applied Computer Systems 14(1), 19–30 (2013)
56. Nikiforova, O., Pavlova, N.: Application of BPMN instead of GRAPES for two-hemisphere model driven approach. In: ADBIS 2009 Workshops, LNCS, vol. 5968, pp. 185–192. Springer (2010)
57. OMG: MOF 2.0 Query/View/Transformation Specification, v1.0. OMG (2008)
58. OMG: Business Process Model and Notation (BPMN), v2.0. OMG (2011)
59. OMG: Unified Modeling Language (OMG UML), v2.5. OMG (2015)
60. Pottinger, R.A., Bernstein, P.A.: Merging models based on given correspondences. In: J.C. Freytag et al. (ed.) Procs. 2003 VLDB Conference, pp. 862–873. Morgan Kaufmann (2003)
61. Rhazali, Y., Hadi, Y., Chana, I., Lahmer, M., Rhattoy, A.: A Model Transformation in Model Driven Architecture from Business Model to Web Model. IAENG International Journal of Computer Science 45(1), 214–227 (2018)
62. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: Analysis-Level Classes from Secure Business Processes Through Model Transformations. In: TrustBus 2007, LNCS, vol. 4657, pp. 104–114. Springer (2007)

63. Rodriguez, A., Garcia-Rodriguez de Guzman, I., Fernandez-Medina, E., Piattini, M.: Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach. Information and Software Technology 52(9), 945–971 (2010)
64. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: Towards Obtaining Analysis-Level Class and Use Case Diagrams from Business Process Models. In: ER Workshops 2008, LNCS, vol. 5232, pp. 103–112. Springer (2008)
65. Rungworawut, W., Senivongse, T.: Using Ontology Search in the Design of Class Diagram from Business Process Model. PWASET 12, 165–170 (2006)
66. Santos, M.Y., Oliveira e Sá, J.: A Data Warehouse Model for Business Processes Data Analytics. Springer, Cham (2016)
67. Santos, M.Y., Machado, R.J.: On the Derivation of Class Diagrams from Use Cases and Logical Software Architectures. In: Proc. of ICSEA '10. pp. 107–113. IEEE (2010)
68. Selonen, P., Koskimies, K., Sakkinen, M.: Transformations Between UML Diagrams. Journal of Database Management 14(3), 37–55 (2003)
69. Sepúlveda, C., Cravero, A., Cares, C.: From Business Process to Data Model: A Systematic Mapping Study. IEEE Latin America Transactions 15(4), 729–736 (2017)
70. Silva, L.F., Leite, J.C.S.P.: Generating requirements views: A transformation-driven approach. Electronic Communications of the EASST 3, 1–14 (2006)
71. Srivastava, S.: Model Transformation Approach for a Goal Oriented Requirements Engineering based WebGRL to Design Models. International Journal of Soft Computing and Engineering (IJSCE) 3(6), 66–75 (2014)
72. Tan, H.B.K., Yang, Y., Blan, L.: Systematic Transformation of functional analysis model in Object Oriented design and Implementation. IEEE Transaction on Software Engineering 32(2), 111–135 (2006)
73. Truscan, D., Fernandes, J.M., Lilius, J.: Tool support for DFD-UML based transformation. In: Proc. of ECBS '04. pp. 378–387. IEEE (2004)
74. Winkler, W.E.: String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In: Proc. of the Section on Survey Research Methods. pp. 354–359. American Statistical Association (1990)
75. Wrycza, S.: The ISAC-driven transition between requirements analysis and ER conceptual modelling. Information Systems 15(6), 603–614 (1990)
76. Zhang, J., Feng, P., Wu, Z., Yu, D., Chen, K.: Activity based CIM modeling and transformation for business process systems. International Journal of Software Engineering and Knowledge Engineering 20(3), 289–309 (2010)

Drazen Brdjanin is an Associate Professor at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina), where he heads the M-lab Research Group. His research interests focus on information systems and software engineering. He has participated in several national and international R&D projects, and also authored a number of research papers and articles in the field of model-driven development.

Aleksandar Vukotic is a Master's Student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). He is a Senior Software Developer at RT-RK Auto and a member of the M-lab Research Group. His research interests include model-driven software development, databases, and UML. He has published a couple of research papers.

Danijela Banjac is a Senior Teaching Assistant and PhD student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). She is a member of the M-lab Research Group. Her research interests include model-driven software development, business process modeling, object-oriented information systems, and UML. She has published several research papers and articles.

Goran Banjac is a Senior Teaching Assistant and PhD student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). He is a member of the M-lab Research Group. His research interests include model-driven software development, business process modeling, databases, and UML. He has published several research papers and articles.

Slavko Maric is a Full Professor at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). His current research interests include: information systems modeling, design and development, databases, eGovernment systems, service oriented architecture and parallel processing. He has published over 50 research papers and articles, and participated in a number of research and development projects.

Received: April 23, 2021; Accepted: December 14, 2021.

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

004

COMPUTER Science and Information Systems : the International journal /
Editor-in-Chief Mirjana Ivanović. – Vol. 19, No 1 (2022) - . – Novi Sad (Trg D. Obradovića 3): ComSIS Consortium, 2022 - (Belgrade : Sigra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 (Print) 2406-1018 (Online) = Computer Science and Information Systems
COBISS.SR-ID 112261644

Cover design: V. Štavljanin
Printed by: Sigra star, Belgrade