ComSIS Computer Science and Information Systems

Volume 18, Number 1, January 2021

Contents

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

Computer Science and Information Systems

Vol 18, No 1, January 2021

ComSIS

Computer Science and Information Systems

Published by ComSIS Consortium

Volume 18, Number 1
January 2021

ComSIS is an international journal published by the ComSIS Consortium

ComSIS Consortium:

University of Belgrade:
Faculty of Organizational Science, Belgrade, Serbia
Faculty of Mathematics, Belgrade, Serbia
School of Electrical Engineering, Belgrade, Serbia
Serbian Academy of Science and Art:
Mathematical Institute, Belgrade, Serbia
Union University:
School of Computing, Belgrade, Serbia

University of Novi Sad:
Faculty of Sciences, Novi Sad, Serbia
Faculty of Technical Sciences, Novi Sad, Serbia
Technical Faculty “Mihajlo Pupin”, Zrenjanin, Serbia
University of Niš:
Faculty of Electronic Engineering, Niš, Serbia
University of Montenegro:
Faculty of Economics, Podgorica, Montenegro

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 18, Number 1, 2021
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 2019 two-year impact factor 0.927,
- 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

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

Computer Science and Information Systems

Volume 18, Number 1, January 2021

CONTENTS

Papers

Special section: Invited papers of Distinguished Top Cited ComSIS authors

Computer Science and Information Systems 18(1): i–iv https://doi.org/10.2298/CSIS210100iI

Editorial

Mirjana Ivanović[1] and Miloš Radovanović[1]

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

At the start of 2021, this first issue of Volume 18 of Computer Science and Information Systems contains 13 regular papers and 4 articles in the Special Section: Invited Papers of Distinguished Top Cited ComSIS Authors in last 10 years. We invited 10 authors of the most cited papers in the last 10 years to prepare new articles for our journal, and we are happy that four of them accepted our invitation. Accordingly, we are very thankful to those most cited authors who were willing to accept our invitation and prepare new papers for our journal, and we hope that their new articles will also be highly cited in the future. Last but not least, acknowledge the diligence and hard work of all our authors and reviewers, without whom the current issue, and journal publication in general, would not be possible.

The regular paper section starts with “Throughput Prediction based on ExtraTree for Stream Processing Tasks” by Zheng Chu et al, where the problem of large volumes of streaming data is tackled by proposing a volatility detection algorithm, a selection algorithm, and a throughput prediction method based on the ExtraTree ensemble learning algorithm. Experimental results demonstrate good accuracy and efficiency of the proposed approach.

The second article, “Multi-Objective Optimization of Container-Based Microservice Scheduling in Edge Computing” by Guisheng Fan el al. formulates container-based microservice scheduling as a multi-objective optimization problem, and proposes a latency, reliability and load balancing aware scheduling (LRLBAS) algorithm to determine the container-based microservice deployment in edge computing, based on particle swarm optimization. Simulation experiments showcase the effectiveness and efficiency of the proposed algorithm.

“PureEdgeSim: A Simulation Framework for Performance Evaluation of Cloud, Edge and Mist Computing Environments” by Charafeddine Mechalikh et al. presents PureEdgeSim, a simulation toolkit that enables the simulation of cloud, edge, and mist computing environments and the evaluation of the adopted resources management strategies, in terms of delays, energy consumption, resources utilization, and tasks success rate. Evaluation on the introduced case study demonstrates the effectiveness of the proposed framework modeling complex and dynamic environments.

In the article entitled “DroidClone: Attack of the Android Malware Clones - A Step Towards Stopping Them,” Shahid Alam and Ibrahim Sogukpinar propose DroidClone, an approach for detection of code clones (segments of code that are similar) in Android applications to help detect malware. DroidClone uses control flow patterns for reducing the effect of obfuscations and detecting clones that are syntactically different but semantically similar enough, and is independent of the underlying programming language. Evaluation incorporating real malware demonstrated good accuracy, as well as a reasonable degree of resistance to obfuscations.

Masoud Reyhani Hamedani et al., in “TrustRec: An Effective Approach to Exploit Implicit Trust and Distrust Relationships along with Explicit ones for Accurate Recommendations,” present TrustRec, an approach based on matrix factorization that provides a solution to three identified problems of existing trust-aware recommendation approaches, incorporating them all into a single matrix factorization model. Experimental results demonstrate that TrustRec outperforms existing approaches in terms of effectiveness and efficiency.

“A Dual Hybrid Recommender System based on SCoR and the Random Forest,” authored by Costas Panagiotakis et al. uses the synthetic coordinate recommendation system (SCoR) and the random forest machine learning model to construct a dual hybrid recommender system by proposing a dual training approach resulting in two recommender systems that are subsequently combined. Experimental results demonstrate the high performance of the proposed system on the Movielens datasets.

The article “A Method of Assessing Rework for Implementing Software Requirements Changes,” by Shalinka Jayatilleke and Richard Lai present a definition for rework and describe a method of assessing rework for implementing software requirements changes. The method consists of three stages: (1) change identification; (2) change analysis and (3) rework assessment. A running example is used to explain the concepts.

“Double-Layer Affective Visual Question Answering Network” by Zihan Guo et al. proposes a network architecture (DAVQAN) that divides the task of generating emotional answers in visual question answering into two simpler subtasks: the generation of non-emotional responses and the production of mood labels, with two independent network layers used to tackle these subtasks. The article also introduces a more advanced word embedding method and more fine-grained image feature extractor to further improve accuracy.

Muhammad Ahmad Rathore and JongWon Kim in their article “Spatio-temporal Summarized Visualization of SmartX Multi-View Visibility in Cloud-native Edge Boxes” explore a family of data summaries that take advantage of the multiple layers i.e. physical/virtual resources with temporal and spatial correlation among distributed edge boxes. The authors present the idea of maintaining summarized spatio-temporal data and verify it through visualization of gathered operational data.

In “A QPSO Algorithm Based on Hierarchical Weight and Its Application in Cloud Computing Task Scheduling,” Guolong Yu et al. propose a modification of the quantum behaved particle swarm optimization (QPSO) algorithm called hierarchical weight QPSO (HWQPSO) in which the higher the fitness value of a particle, the higher the level of the particle, and the greater the weight. The effectiveness of the approach is demonstrated on the task scheduling problem for cloud computing platforms, exhibiting faster convergence, shortest time consumption and the most balanced computing resource load.

The article “Convexity of Hesitant Fuzzy Sets Based on Aggregation Functions” by Pedro Huidobro et al. mathematically extends the notion of convexity for hesitant fuzzy sets in order to fulfill some necessary properties, namely being compatible with the intersection operation and fulfilling the cutworthy property.

“Spoken Notifications in Smart Environments Using Croatian Language” by Renato Šoić et al. proposes a model for natural language generation and speech synthesis in a smart environment using the Croatian language. Evaluation of user experience quality demonstrates that most users perceive grammatically correct spoken texts as being of the highest quality.

Concluding the regular paper section, “Students’ Preferences in Selection of Computer Science and Informatics Studies – A Comprehensive Empirical Case Study” by Miloš Savić et al. presents a survey-based empirical study with the goal of determining the main motivating factors directing students to select computer science, informatics or similar study programs. The survey was conducted on a sample of more than 1500 students from five well established faculties of computer science and informatics at three largest university cities in Serbia, showing that while the majority of students are primarily interested in that topic, there was also a significant number of students who wanted to study something else, but selected computer science and informatics due to more possibilities for employment and higher salaries.

The Special Section: Invited Papers of Distinguished Top Cited ComSIS Authors begins with “Hypothetical Tensor-based Multi-criteria Recommender System for New Users with Partial Preferences,” where Minsung Hong and Jason J. Jung propose a hypothetical tensor model (HTM) to leverage auxiliary data complemented through three intuitive rules dealing with user’s unfamiliarity with item domains. The approach has three phases: (1) four patterns of partial preferences are found that are caused by users’ unfamiliarity, (2) rules are defined by considering relationships between multi-criteria, and (3) complemented preferences are modeled by a tensor to maintain an inherent structure of and correlations between the multi-criteria. Experiments on a TripAdvisor dataset showed that the approach offers a considerable performance boost compared to the baseline methods.

The second article in the special section, “Metaphor Research in the 21st Century: A Bibliographic Analysis” by Dongyu Zhang et al. examines the advancements in metaphor research from 2000 to 2017 using data retrieved from Microsoft Academic Graph and Web of Science. The article presents a macro analysis of metaphor research and expounds the underlying patterns of its development, revealing the evolution of research topics and the inherent relationships among them and providing insights into the current state of the art of metaphor research as well as future trends in this field.

Next article, “Incorporating Privacy by Design in Body Sensor Networks for Medical Applications: A Privacy and Data Protection Framework” by Christos Kalloniatis et al. proposes a privacy and data protection framework that provides the appropriate steps to undertake proper technical, organizational and procedural measures in an eHealth/M-Health system. The framework supports the combination of privacy with the newly introduced General Data Protection Regulation (GDPR) requirements in order to create a strong elicitation process for deriving the set of the technical security and privacy requirements that should be addressed.

Finally, Sašo Sršen and Marjan Mernik, in “A JSSP Solution for Production Planning Optimization Combining Industrial Engineering and Evolutionary Algorithms” tackle the job shop scheduling problem (JSSP), where p processes and n jobs should be processed on m machines so that the total completion time is minimal. In this article, the production times are integrated into an evolutionary algorithm to solve real-world JSSP problems, proposing an Internet of Things (IoT) architecture as a possible solution.

We hope that this issue brings diverse and very interesting papers that cover a range of contemporary research topics and that scientific community and readers will enjoy reading them. Also, we believe that the presented research could be attractive and represent a good starting point and/or motivation for other authors to extend the presented scientific achievements and continue with similar research efforts.

Computer Science and Information Systems 18(1):1–22 https://doi.org/10.2298/CSIS200131031C

Throughput Prediction based on ExtraTree for Stream Processing Tasks

Zheng Chu[1], Jiong Yu[1], and Askar Hamdulla[1]

School of Information Science and Engineering, Xinjiang University, Urumqi 830046, PR China
chuzheng@stu.xju.edu.cn,{yujiong,askar}@xju.edu.cn

Abstract. In the era of big data, as the amount of streaming data continues to increase, stream processing tasks (SPTs) face serious challenges in real-time processing scenarios with low latency and high throughput. However, much of the current literature on the performance of SPTs pays attention to the reactive approach, which cannot well avoid the problem of system crashes due to the inherent performance volatility. In this paper, a novel throughput prediction method based on ExtraTree for SPTs is presented to address these challenges. A volatility detection algorithm was proposed to obtain the reasonable metric values after the performance volatility of SPTs was studied. Moreover, a selection algorithm of regression function was proposed to output the performance values of SPTs under a relative stead state. Furthermore, a ExtraTree-based algorithm was proposed to predict the throughput of SPTs. The experimental results from two open-source benchmarks running on Apache Flink, a popular stream processing system (SPS), indicated that the average of the accuracy and efficiency of the proposed method could achieve 90.535% and 0.835 s/10,000 samples, which proved the effectiveness of the proposed method on the task of predicting the throughput of SPTs.

Keywords: streaming data, stream processing tasks, performance prediction, ensemble learning, ExtraTree.

1. Introduction

The emergence of big data processing systems enables organizations to store and process high-dimension, diverse, and high-speed data [17]. Data processing approaches are usually divided into batch processing and stream processing. The former is generally used for static data, and the latter is used for streaming data. For dynamically changing data, most of the systems based on the Map-reduce [6] computing algorithm use the batch processing approach to process and analyze the data. Products in the ecosystem include HDFS [2], HBase [25], and Hive [27]. Accordingly, popular stream processing systems (SPSs) include Apache Flink [4], Twitter Heron [16], and Apache Storm [26], etc. These systems mainly use stream processing approach to process and analyze data.

With the rapid development of social media [8], news sources, and the Internet of Things (IoT) [14], large-scale streaming data from various sensor devices [20], mobile devices [15], and smart devices [13] generated and streamlined by the SPS in real time. Due to the streaming data with the characteristics of large scale, rapid change, and continuous generation, SPSs and SPTs must ensure low latency and high throughput as much as possible. To achieve the dual goals of low latency and high throughput, current research efforts are focusing on task scheduling [29], load balancing [18], elastic computing [11], etc. In these studies, all strategies are triggered after streaming data occurs a burst and the performance of SPT cannot meet the requirements of users, i.e., reactive approach. This approach may render the system unavailable for a certain period. If the streaming data continues to fluctuate or oscillate, the above-mentioned reactive approach will cause the system to enter a continuous adjustment process, which will cause the continuous adjustment time exceed the available time, and even cause the system to crash.

A reasonable approach is to predict the throughput of SPTs under different conditions, i.e., different streaming data rates. If the current throughput can be predicted in advance, system crashes can be better prevented. To avoid this situation that SPTs cannot cope with the rapid increase in the amount of streaming data due to its limited processing capacity, the study of this paper aims to predict the maximum throughput of SPTs with latency guarantees. This issue is also an important research in load management, query scheduling, permission control, schedule monitoring, system scale customization, etc., and these studies will not be described here.

To predict the maximum throughput of SPTs with latency guarantees, we analyzed the performance volatility of SPTs and proposed a detection algorithm for performance volatility. Moreover, a polynomial regression algorithm was applied to the performance volatility analysis to accurately estimate the throughput at a specific data rate. Furthermore, the throughput of SPTs in a relatively stable state was output using a volatility regression algorithm. Finally, an ExtraTree-based algorithm was used for predicting the throughput of SPTs.

The main contributions of this paper are as follows:

(1) A volatility algorithm was proposed to detection the performance volatility of SPTs after the volatility was studied.

(2) A polynomial regression algorithm was proposed to apply to the performance volatility to evaluate the performance of SPTs in a relatively stable state by configuring different regression items and selecting the appropriate regression function automatically.

(3) An ExtraTree-based algorithm was proposed to predict the throughput of SPTs. In particular, we first predict the maximum throughput of SPTs in this paper;

(4) The experimental results from two open-source benchmarks running on Apache Flink, a popular SPS, indicated that the average of the accuracy and efficiency of the proposed method could reach 90.535% and 0.835 s/10,000 samples, which proved the effectiveness of the proposed method. More importantly, the proposed method outperforms other ensemble learning algorithms in term of accuracy and efficiency.

The structure of this paper is organized as follows: Section 2 gives an overview of the related work. Section 3 describes the performance volatility phenomenon and volatility detection algorithm in detail. The performance evaluation method for SPTs are described in Section 4. Section 5 elaborates on the performance prediction algorithm. In Section 6, we evaluate the effect of the ExtraTree-based throughput prediction algorithm on SPTs through experiments and analysis. Section 7 concludes the paper with a summary and suggestions for future works.

2. Related Work

At present, related works have achieved good results in many fields, e.g., natural language processing [30], speech recognition [7], image processing [21], autopilot [1], etc., using machine learning algorithms, while there are relatively few works on traditional computer systems, especially SPSs. In this section, we will describe related works, mainly involving SPSs, ensemble learning, and performance prediction for SPTs.

Stream processing systems: A stream processing system is a kind of system that continuously processes, aggregates, and analyzes streaming data. Unlike Hive and HBase, it is a software system based on a stream computing framework. The processing latency of a SPS is measured in seconds or even milliseconds level. Such a system typically uses a Directed Acyclic Graph (DAG) computation algorithm to process streaming data on the nodes within the graph and pass the streaming data on the edges between the nodes. In [26], the authors proposed a stream processing system, named Apache Storm, which is a distributed, reliable, and fault-tolerant SPS. Study [4] proposed a SPS, named Apache Flink that is used for computing unbounded and bounded streaming data using a stateful computing framework and a distributed processing engine. In [16], the authors proposed a SPS, named Twitter Heron that is a real-time, distributed, and fault-tolerant stream processing engine. An SPT is a DAG task written by users and running in a specific SPS.

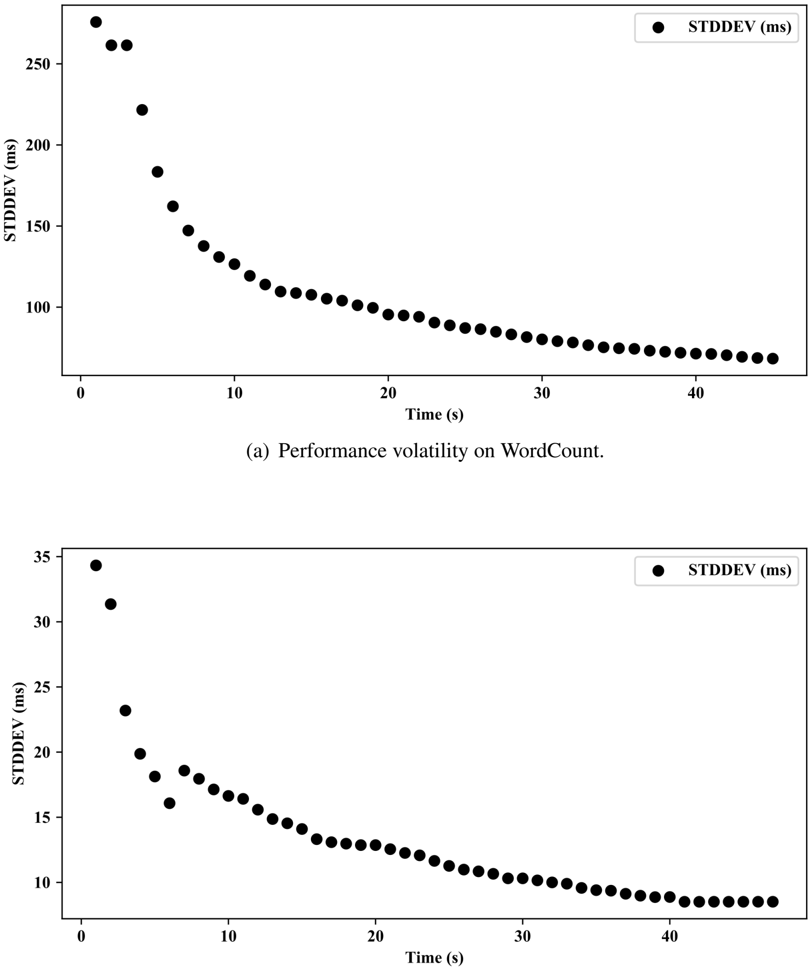
Ensemble learning: Study [5] first proposed the concept of ensemble learning. In [23], the authors used Boosting algorithm to combine multiple weak classifiers into a strong classifier. This algorithm makes ensemble learning to become an important research area. Study [9] proposed AdaBoost ensemble learning algorithm that is efficient and widely used in many fields. In [3], the authors proposed random forest algorithm that has achieved good results in many fields, so it is regarded as one of the best algorithms in machine learning. The authors proposed the integration of Gradient Boosting Decision Tree (GBDT) algorithm in [10]. GBDT is also a member of the Boosting family. In [12], the authors proposed the ExtraTree algorithm that can construct a completely randomized tree in extreme cases and its structure is independent of the output value of the learning sample.

Performance evaluation and prediction for SPSs: Most current performance evaluations for SPSs are based on experience methods [22] [24]. These methods first deployed SPTs in a specific SPS, and then collected performance metrics for task feedback, e.g., latency, throughput, etc. The performance prediction for SPSs is also like performance evaluation [28]. The above two types of performance evaluation and prediction research mainly focused on the impact of hardware resource on the performance.

In an actual production environment, once a SPT is deployed, changes in the task will affect the execution of SPT. The best way is to predict performance in advance and prevent it from happening this situation. The most direct impact on the performance is the data rate of the SPT. In this paper, with the latency guarantees, the performance volatility of SPTs was analyzed and the ExtraTree algorithm was used to predict the throughput of SPTs under different data rates to avoid the situation of unavailable services.

3. Performance Volatility

In this section, we mainly describe the performance volatility of SPTs during execution. To obtain the performance of the task under a relatively stable state, a volatility detection algorithm was proposed and analyzed.



(a) Performance volatility on WordCount.

(b) Performance volatility on Iteration.

Fig. 1. Performance volatility on WordCount and Iteration.

3.1. Phenomenon of Performance Volatility

Performance volatility is a common phenomenon in which an SPT exhibits unstable performance over time under normal operating conditions, as shown in Fig.1 (data sets and experimental environment are described in Section 6).

The horizontal axis of Fig 1(a) represents the running time of an SPT, and the vertical axis represents the standard deviation of the latency metric. It can be clearly seen from the Fig 1(a) that the standard deviation reaches more than 250 in the initial stage. After 40 s, standard deviation stays around 20 and keeps relatively stable.

The results of Fig 1(b) are similar to that of Fig 1(a) and shows the phenomenon that the latency metric fluctuates over time. After 40 s, the standard deviation of the latency metric is kept at about 5. The performance of SPTs has the following characteristics: (1) the performance of SPTs fluctuates over time; (2) performance volatility tend to decline over time and eventually tend to be relatively stable.

3.2. Volatility Detection

Generally, the metrics for volatility detection of samples have extreme values, variances, and standard deviations. These metrics are formulated as follows:

$$X_{range} = \max(X) - \min(X) \tag{1}$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{X}\right)^2 \tag{2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{X}\right)^2} \tag{3}$$

In Equation 1, the extreme value X_{range} is obtained by subtracting the minimum value from the maximum value in the sample set. In Equation 2, the variance metric σ^2 is the average $\bar{X}$ of the squared value of the difference between the average of each sample value x_i and the total sample size. In Equation 3, the standard deviation σ is the square root of the variance σ^2.

The extreme metric is very susceptible to noise from the samples. The variance metric is used to measure the volatility of a group of samples, that is, the deviation of a group of samples from the mean of samples. Similarly, the standard deviation can also reflect the degree of deviation among samples. However, the value of the variance is the square of the difference between the sample and the mean, which is greatly affected by the sample data. Therefore, it is more reasonable to use the standard deviation as a measure of the performance volatility for SPTs. The performance volatility detection algorithm uses the standard deviation to measure volatility.

By executing the volatility detection algorithm, the performance volatility values of SPTs are easily and efficiently calculated at a certain moment, but we do not know whether the volatility values are in a relatively stable state. In Section 4, we will evaluate stable states and output performance.

4. Performance Evaluation

In this section, we first perform a regression algorithm on the volatility values described in the previous section, and then elaborate on the choice of regression functions. Finally, the algorithm outputs the performance values in a relatively stable state, that is, evaluates the relative steady-state performance of SPTs.

4.1. Volatility Regression

To reduce the burden on humans to observe performance volatility, it is necessary to intelligently identify performance when an SPT is in a relatively stable state. Through the description of the performance volatility described in Section 3.1, the volatility will decrease and become relatively stable over time. To do this, we first perform a polynomial regression on the performance volatility, as shown in follows:

$$\hat{y}(w,\sigma) = w_0\sigma_0^0 + w_1\sigma_1^1 + \cdots + w_m\sigma_m^m + \xi(\sigma) \tag{4}$$

where w_i denotes the weight, and σ_i^i denotes the performance volatility value. If $\sigma_0 = 1$, Equation 4 is formulated as follows:

$$\hat{y}(w,\sigma) = \sigma \cdot W + \xi(\sigma) \tag{5}$$

where σ represents an $n \times (M+1)$ matrix, and W represents a $(M+1) \times 1$ matrix. In Equation 4 and 5, $\xi(\sigma)$ represents the error function and it is formulated as follows:

$$\xi(\sigma) = \min\left\{\|\sigma \cdot \mathrm{W} - y\|^2 + \alpha\|w\|^2\right\} \tag{6}$$

Regression task is performed by minimizing the sum of squared errors, and α is used for controlling the amount of expansion and contraction of the coefficients. Thus, the regression function of the performance volatility of SPTs is obtained by polynomial regression, and the derivative $\hat{y}'(w,\sigma)$ of the regression function was obtained. The problem of determining whether an SPT is in a relatively stable state is converted into a problem of calculating derivative value of regression function, i.e., $\hat{y}'(w,\sigma)$. However, this method will lead to another problem in selecting regression functions, because configuring different regression items will obtain different regression functions.

Some regression functions are capable of solving the volatility selection problem well, but others will bring unsatisfactory results. This phenomenon is shown in the experiment in Section 5. Ideally, $\hat{y}'(w,\sigma)$ close to 1 or -1 means that the performance of an SPT is more unstable, and close to 0 proves that the performance is stable.

4.2. Regression Selection

The R-squared value R^2, called the coefficient of determination, reflects the proportion of all variation of the dependent variable that can be interpreted by the independent variable through the regression relationship. The higher the value, the better the algorithm. The maximum value is 1, and R^2 is formulated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(\sigma_i - \hat{\sigma})^2}{\sum_{i=1}^{m}(\sigma_i - \bar{\sigma})^2} \tag{7}$$

By calculating the R^2 of the corresponding function of multiple regression terms, the function with the largest R^2 is selected. This function is the best choice among candidate functions, and the algorithm is briefly described in Algorithm 1.

Regarding the time complexity of Algorithm 1, the complexity of the loop in step 1 is $O(m)$, the loop in step 2 is $O(l)$, the loop in step 3 is $O(l)$, the loop in step 4 is $O(l)$.

```
Algorithm 1: RIS (Regression Item Selection)
Input: Performance volatility set $F = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$, regression item set
       $D = \{d_1, d_2, \ldots, d_l\}$.
Output: Regression function $\widehat{y}_{max}$.
begin
    (1) Calculate the mean of samples $\overline{\sigma}$ in $F$:
    for $i \leftarrow 0$ to m-1 do
        $\overline{\sigma} \leftarrow \overline{\sigma} + \sigma_i$;
    end
    $\overline{\sigma} \leftarrow \frac{\overline{\sigma}}{m}$;
    (2) Calculate regression set $Y = \{\widehat{y_1}, \widehat{y_2}, \ldots, \widehat{y_l}\}$ using regression item set
      $D = \{d_1, d_2, \ldots, d_l\}$:
    for $i \leftarrow 0$ to l-1 do
        $\widehat{y_i} \leftarrow Regression(d_i)$;
    end 3
    Calculate R-squared set $R^2 = \{r_1^2, r_2^2, \ldots, r_l^2\}$ using regression function set
      $Y = \{\widehat{y_1}, \widehat{y_2}, \ldots, \widehat{y_l}\}$ (refer Equation 7):
    for $i \leftarrow 0$ to l-1 do
        $R[i] \leftarrow 1 - \frac{\sum_{i=1}^{m}(\sigma_i - \hat{\sigma})^2}{\sum_{i=1}^{m}(\sigma_i - \bar{\sigma})^2}$;
    end 4
    Select the maximum $r_{max}^2$ in $R^2 = \{r_1^2, r_2^2, \ldots, r_l^2\}$:
    $r_{max}^2 \leftarrow 0$;
    for $i \leftarrow 0$ to l-1 do
        if $r_i^2 > r_{max}^2$ then
            $r_{max}^2 \leftarrow r_i^2$
        end
    end
    (5) Calculate the function $\widehat{y}_{max}$ in $Y$ using $r_{max}^2$:
    for $i \leftarrow 0$ to l-1 do
        if $r_i^2 = r_{max}^2$ then
            $\widehat{y}_{max} \leftarrow \widehat{y_i}$;
        end
    end
    return $\widehat{y}_{max}$;
end
```

In step 5, the loop is also $O(l)$. Therefore, the final time complexity of Algorithm 1 is $O(m + 4l)$.

Also, regarding the spatial complexity of Algorithm 1, the complexity of $\overline{\sigma}$ in step 1 is $O(1)$. The set Y in step 2 is $O(l)$. The loop R-square set in step 3 is $O(l)$, and r_{max}^2 in step 4 is $O(1)$. The $\widehat{y}_{maxl}$ in step 5 is $O(1)$. Therefore, the final spatial complexity of Algorithm 1 is $O(2l + 3)$, i.e., $O(l)$.

4.3. Performance Output under A Steady State

The optimal regression function is obtained through the regression term selection algorithm, i.e., Algorithm 1. When the value of the derivative function of the regression func-

Algorithm 2: POA (Performance Output Algorithm)

```
Input: Volatility regression function $\widehat{y}_{max}$, performance metric set
        $X = \{x_1, x_2, \ldots, x_n\}$.
Output: Performance metric $x_i$.
begin
    (1) Calculate the derivative $\widehat{y}^{'}_{max}$ using $\widehat{y}_{max}$;
    (2) Calculate each derivative $\widehat{y}^{'}_{max}(x_i)$ in $X = \{x_1, x_2, \ldots, x_n\}$:
    $Y[i] \leftarrow Null$;
    for $i \leftarrow 0$ to n-1 do
        $Y[i] \leftarrow \widehat{y}^{'}_{max}(x_i)$;
    end 3
    Output performance $x_i$ when the derivative value is equal to 0:
    for $i \leftarrow 0$ to l-1 do
        if $Y[i] == 0$ then
            return $x_i$;
        end
    end
end
```

tion is 0, an SPT enters a relatively stable state. At this time, the performance value is output through the performance output algorithm, i.e., Algorithm 2.

Algorithm 2 is relatively simple, so no specific analysis is performed here. The time complexity of the algorithm is $O(n)$, and the spatial complexity is $O(1)$.

5. Performance Prediction

In Section 4, the performance output algorithm outputs the throughput of an SPT in a relatively stable state, so the ExtraTree algorithm is used for predicting performance at different data rates. In this section, this algorithm is described.

5.1. ExtraTree Introduction

ExtraTree is a novel tree-based ensemble learning algorithm for supervising classification and regression problems. It mainly emphasizes on randomness and selection for segment point when splitting tree nodes. In extreme cases, it is constructed completely randomly. The structure of the tree is independent of the output values of the learning samples. Compared with the random forest algorithm, this algorithm has higher computational efficiency and higher accuracy. The ExtraTree algorithm is very similar to the random forest algorithm. Although they are composed of multiple decision trees, the ExtraTree and the random forest have two differences: (1) the random forest uses the Bagging algorithm, that is to say, the training samples for each weak learner are not all, but the ExtraTree uses all training samples to train every weak learner. In addition, ExtraTree adopts a random selection strategy to select features, so its results are better than random forests; (2) the random forest obtains the best bifurcation attribute in a random subset, but the ExtraTree obtains the bifurcation value completely and randomly to implement the bifurcation

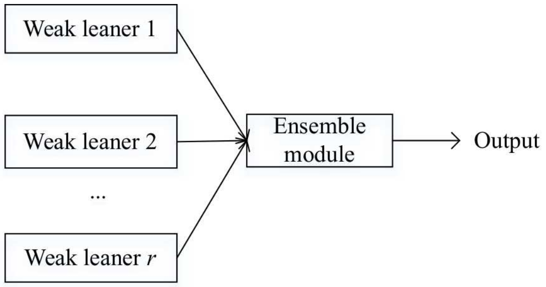


Fig. 2. The structure of ensemble learning.

of the decision tree. Ensemble learning forms a strong ensemble learning algorithm by constructing and combining multiple weak learners to complete specific learning tasks.

Fig 2 shows a general structure of ensemble learning that combines a group of weak learners through a specific strategy. Weak learners are usually trained by existing learning algorithms, such as C4.5 decision tree algorithm, BP neural network algorithm, etc. One of the most important advantages of ensemble learning is that the algorithm achieves superior excellent generalization than a single learner by combining multiple weak learners. In general, it combines non-optimal learners into one piece and gets the best learner. Therefore, the combination strategy for weak learners is particularly important. Assuming that ensemble learning includes T weak learners $h_1, h_1, \ldots, h_T$, where the output of h_i on x is $h_i(x)$. Average, and voting strategy are formulated as follows:

$$H(x) = \frac{1}{T}\sum_{i=1}^{T} h_i(x) \tag{8}$$

$$H(x) = \frac{1}{T}\sum_{i=1}^{T} w_i h_i(x) \tag{9}$$

where w_i is the weight of the weak learner h_i. To be noted, $w_i \geq 0$ and $\sum_{i=1}^{T} w_i = 1$ are required.

5.2. Algorithm Construction

Fig 3 shows a schematic diagram of the ExtraTree structure. The ExtraTree algorithm contains multiple decision trees, each of which contains a tree-like decision node sequence. Based on this sequence, the tree splits into various branches until it reaches the end of the tree (the leaf node). The prediction result of each decision tree is output through the leaf nodes, and the final outputs of the multiple decision trees are combined for prediction.

For the throughput prediction algorithm of SPTs, assuming that data set is $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where N denotes the sample size, x_i denotes the sample data, and y_N denotes the throughput of an SPT. When generating each decision

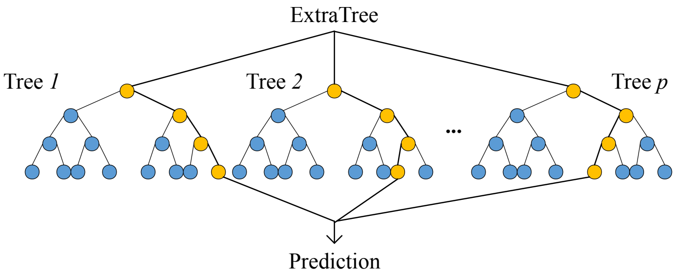


Fig. 3. The structure of ExtraTree.

Algorithm 3: EBA (ExtraTree Building Algorithm)

```
Input: Data set $D$, the number of trees $N_t$.
Output: ExtraTree $F_{tree}$.
begin
    for $i \leftarrow 0$ to $N_t - 1$ do
        (1) Calculate the optimal feature $j$ and the point $s$ to split current node:
        $\min_{j,s}[\sum_{x_i \in R_1(j,s)} (y_i - \hat{c}_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - \hat{c}_2)^2]$;
        (2) Calculate output value $\widehat{c_m}$ using $min(j, s)$ in the current node:
        $\widehat{c_m} \leftarrow \frac{1}{N_m} \sum_{x_i \in R_m(j,s)} y_i$,
        where $R_1(j,s) = \left\{x|x^{(j)} \le s\right\}$, $R_2(j,s) = \left\{x|x^{(j)} > s\right\}$;
        (3) Repeat (1) and (2) using $R_1(j,s)$ and $R_2(j,s)$;
        (4) Divide input space into $m$ nodes $R_1, R_2, \ldots, R_m$ and generate decision tree
         $f_i(x)$:
        $f_i(x) \leftarrow \sum_{m=1}^{M} \hat{c}_m I\left(x \in R_m\right)$.
        where $I = \begin{cases} 1\ if\ (x \in R_m) \\ 0\ if\ (x \notin R_m) \end{cases}$;
        (5) Add current decision tree $f_i(x)$ into $F_{tree}$:
        $F_{tree}[i] \leftarrow f_i(x)$;
    end
    return $F_{tree}$
end
```

tree, the algorithm calculates the best features j and output value s, as shown in follows:

$$\min_{j,s}\left[\sum_{x_i \in R_1(j,s)} \left(y_i - \hat{c}_1\right)^2 + \sum_{x_i \in R_2(j,s)} \left(y_i - \hat{c}_2\right)^2\right] \tag{10}$$

where $R_1(j,s) = \left\{x|x^{(j)} \le s\right\}$ and $R_2(j,s) = \left\{x|x^{(j)} > s\right\}$ are two regions divided by j and s. $\hat{c}_1$ and $\hat{c}_2$ are the throughput output values. In addition, $\left(y_i - \hat{c}_1\right)^2$ and $\left(y_i - \hat{c}_2\right)^2$ are the mean square error (MSE). The algorithm repeats the above steps until all features are segmented, and the construction process is shown in Algorithm 3.

For algorithm 3, it is assumed that the number of features is k, steps 1-2 need to be repeated k times. Moreover, a total of cycles is required N_t. Therefore, the time complexity of Algorithm 3 is $O(kN_t)$.

```
Algorithm 4: TPA (Throughput Prediction Algorithm)
Input: ExtraTree $F_{tree}$, Sample $x$.
Output: The predicted throughput $p$.
begin
    $p_{sum} \leftarrow 0$;
    for $i \leftarrow 0$ to $N_t - 1$ do
        Predict throughput $p$ and add it to $p_{sum}$:
        $p_{sum} \leftarrow p_{sum} + f_i(x)$;
    end
    Calculate the mean predicted throughput:
    $p \leftarrow \frac{p_{sum}}{N_t}$;
    return $p_{sum}$;
end
```

5.3. Throughput Prediction

The output of Algorithm 3 during the prediction phase is the mean of the output values of multiple decision trees and it is formulated as follows:

$$f(x) = \frac{1}{N_t} \sum_{i=1}^{N_t} f_i(x) \tag{11}$$

where N_t is the number of decision trees, $f_i(x)$ is the predicted throughput value, and the prediction process is shown in Algorithm 4.

Algorithm 4 is relatively simple, giving the time complexity of the algorithm is $O(N_t)$, and the space complexity is $O(1)$.

6. Experimental Evaluation

In this section, we describe the methodology, experimental environment, evaluation metrics, volatility regression, comparison of errors, comparison of accuracy and efficiency, and the impact of different sample ratio on errors in detail.

6.1. Methodology

In the experiments, the proposed methodology and the proposed prediction model are applied on an evolving data stream. The overall work principal is shown in Fig. 4.

As shown in Fig. 4, the experimental methodology consists of two components: (1) Online prediction, and (2) Offline learning. The first component firstly detects volatility (Section 3), and then evaluates performance (Section 4) in a real-time fashion. When the performance in steady state is evaluated, the output performance is used to predict the throughput in a real-time style. In addition, an copy of the output is used for training the proposed model in a offline style. During the offline learning phase, the model continuously optimizes itself.

Two open-source benchmarks, i.e., WordCount (WC), and Iteration (ITE) were used to evaluate the effectiveness of the proposed method. An external server was built outside

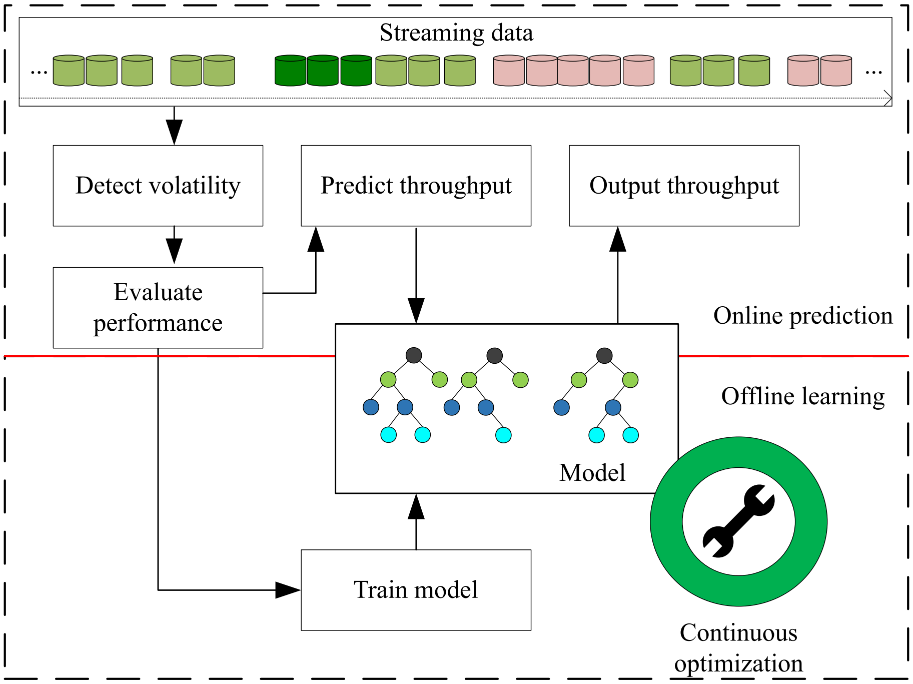


Fig. 4. The experimental methodology.

an SPS cluster that includes one JobManager and three TaskManagers. The external server undertaken the task of collecting throughout of SPTs in real time, and executed real-time throughput prediction for SPTs.

In these benchmarks, WC sent English sentences to an SPT by configuring different sending rates. The SPT first segmented the received English sentences, and then continuously counted the number of occurrences of each word. ITE continuously sent values to an SPT, and then the SPT iteratively calculated the values. Table 1 summarized all data sets from two benchmarks.

Table 1. The description of data sets.

Benchmarks	Total sample size	Training sample size	Predicting sample size
WC	99,980	79,984	19,996
ITE	100,002	80,002	20,000

During the performance prediction phase, three ensemble learning algorithms were used to compare the proposed algorithm.

AdaBoost (Adaptive Boosting), a typical Boosting algorithm, belongs to the Boosting algorithm family. The core of the algorithm is the process of promoting weak learner to a strong learner. The working mechanism is as follows: (1) a weak learner is trained from the initial training set; (2) the training sample distribution is adjusted according to the performance of the weak learner, so that the training samples of the previous weak learner's errors receive more attention on the subsequent training process; (3) train the next weak learner based on the adjusted sample distribution. Repeat these processes until the number of weak learners reaches T, and finally combine the weights of all weak learners. The AdaBoost algorithm is a linear combination based on the weak learners, and it is formulated as follows:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{12}$$

where α_t is the proportion of weak learners to a strong learner, which is different from the weighted average method.

GBDT is also a member of the Boosting family. When training a single weak learner, the algorithm considers the loss function of the previous weak learners. In addition, GBDT also uses an iterative approach through a forward-distributed algorithm. Note that weak learners in this algorithm can only use the CART regression tree algorithm.

Random Forest (RF), an extension of Bagging, is based on the ensemble learning of Bagging with decision tree learners, and adds the characteristics of random attribute selection. The Bagging randomly selects training data, and then constructs multiple weak learners. Finally, it combines multiple decision trees to improve the overall performance. In short, a random forest is obtained by constructing multiple decision trees and merging all the decision trees together to achieve accurate and stable prediction results. It has the advantages of simplicity, easy implementation, and low computational cost. Therefore, this algorithm is one of the comparison algorithms in this paper.

Additionally, to fairly evaluate the performance of different algorithms, the parameter values for each algorithm used in the experimental study is set to the same. Main parameter configurations are summarized in Table 2.

Table 2. Main parameter configurations of different algorithms.

Parameters	Values	Description
n_estimators	50	The number of trees in the forest.
max_depth	30	The maximum depth of the tree.
min_samples_split	2	The minimum number of samples required to split an internal node.
min_samples_leaf	1	The minimum number of samples required to be at a leaf node.

Other parameter configurations use the default values in scikit-learn packages [19].

6.2. Experimental Environment

There are many popular SPSs, such as Apache Flink, Twitter Heron, Apache Storm, Apache Spark, etc. Apache Spark simulates real-world stream processing using a micro-batch processing approach. Twitter Heron is an enhanced version of Apache Storm. Because Apache Flink has strong state support and high performance for streaming data, it was used as the carrier for all experiments.

Apache Flink is built in a local cluster consisting of four servers. One server is the JobManager (Master) and others are TaskManagers (Slaves). The JobManager server is mainly responsible for the distribution and coordination of tasks, and TaskManager servers are mainly responsible for executing specific SPTs, i.e., WC, and ITE. The four servers in the cluster have the same hardware configuration as the external server. CPU is ”Intel(R) Core(TM) i7-4790 CPU 3.60GHz”, memory is 8 GB, hard disk is 500 G, and operating system is CentOS-6.5.

6.3. Evaluation Metrics

To evaluate the performance of the proposed method for the throughput prediction of SPTs, six metrics were used to compare different algorithms.

(1) Explain variance (EV) is a measure of the ability of a regression equation to explain the degree of change in the dependent variable or the degree to which the equation fits a sample. The closer the EV is to 1, the better the algorithm, and the lower the value, the worse the algorithm. EV is formulated as follows:

$$EV = 1 - \frac{\operatorname{Var}(y_i - \hat{y})}{\operatorname{Var}(y_i)} \tag{13}$$

(2) The R-squared value (R^2) is the degree to which the regression equation characterizes the dependent variables. The R-squared of the best algorithm is 1, and the difference is smaller. This metric is formulated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{m} (y_i - \hat{y})^2}{\sum_{i=1}^{m} (y_i - \bar{y})^2} \tag{14}$$

(3) The mean absolute error (MAE) is the average difference between the predicted value and the true value, and it is formulated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i| \tag{15}$$

(4) The mean square error (MSE) is the expected value of the square of the difference between the predicted value and the true value. It is recorded as a convenient method to measure the average error. It was used to evaluate how much the data has changed. The smaller the value, the better the prediction algorithm will describe the data in the experiments. This metric formulated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (f_i - y_i)^2 \tag{16}$$

(5) The root mean square error ($RMSE$) is the arithmetic square root of the mean square error, and it is formulated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (f_i - y_i)^2} = \sqrt{MSE} \tag{17}$$

(6) The median absolute error ($MediaAE$) is formulated as follows:

$$MediaAE = \mathrm{media}\left(\||y_i - \hat{y}_i|, \ldots, |y_n - \hat{y}_n|\right) \tag{18}$$

From Equation 13 to Equation 18, f_i is the predicted throughput value, y_i is the real throughput value, $e_i = |f_i - y_i|$ is the absolute error. n is the sample size, $\overline{y}$ is the mean of throughput.

6.4. Volatility Regression

In Section 3, polynomial regression was introduced and performed on performance volatility values. In this experiment, by configuring different regression terms $D = \{d_1, d_2, \cdots, d_l\}$, Equation 3 was used to calculate the corresponding regression function set. Regression items were set to 2, 3, 4, and 5, respectively. The experimental results were shown in 5.

In Fig 5, there was a clear trend of increasing prediction performance as terms increased. For example, the regression function and the performance volatility values were very different when the regression term was set to 2. In contrast, when the regression term was set to 5, the function could well regress the performance volatility values. The results of Fig 5 showed that the tangent of regression function on WC was -1 when the time was about 30 s, but the corresponding performance volatility scatter plot did not reach a relative stable state, which explained the necessity of regression term selection, i.e., Algorithm 1, in Section 4. The results from other benchmarks, i.e., ITE, showed the similar trend as shown in Fig 5. If the algorithm output the throughput of an SPT at this time, meaning that it was not in a relatively stable performance. When the regression term became larger, e.g., 5, the output of performance was more representative of the throughput in a relatively stable state, as shown in Fig 5(d), and Fig 5(h). To evaluate the performance of regression item selection algorithm in more detail, the R-squared values R^2 of different regression items were shown in Table 3.

Table 3. R^2 of different regression items.

Benchmarks	Item-2	Item-3	Item-4	Item-5
WC	0.87	0.95	0.98	0.98
ITE	0.84	0.90	0.93	0.96

In Table 3, R^2 gradually approached 1 as the regression term on WC, and ITE increased, which also indicated that the selection of regression terms was necessary in the performance volatility regression process. Therefore, the regression algorithm based on excellent performance was more accurate to judge a relatively stable state, and the output was more reasonable.

6.5. Comparison of errors

The purpose of this experiment was to compare the prediction errors of different ensemble learning algorithms under a steady state at different data rates. The experimental results were shown in Fig 6.

Table 4. The errors of different ensemble learning algorithms.

Benchmarks	MAE				MSE				MediaAE			
	A	E	G	R	A	E	G	R	A	E	G	R
WC	0.21	**0.15**	0.19	0.23	0.15	**0.04**	0.14	0.08	0.09	**0.07**	0.12	0.13
ITE	0.12	**0.05**	0.10	0.09	0.09	**0.03**	0.10	0.12	0.05	**0.02**	0.13	0.10
AVG	0.165	**0.10**	0.145	0.16	0.12	**0.035**	0.12	0.10	0.07	**0.045**	0.125	0.115

Note: (A) AdaBoost; (E) ExtraTree; (G) GBDT; (R) Random Forest; (AVG) Average.

As shown in Fig 6, there were differences between the four ensemble learning algorithms, and ExtraTree had lower errors compared with other algorithms, i.e., AdaBoost, GBDT, and Random Forest. Table 4 summarized the detailed results.

As shown in Table 4, MAE, MSE, and $MediaAE$ of ExtraTree on all benchmarks were lower than that of other algorithms. For instance, MAE of ExtraTree, AdaBoost, GBDT, and Random Forest on the benchmark WC were 0.15, 0.21, 0.19, and 0.23, respectively, which showed that the ExtraTree had the lowest error. On the benchmark ITE, MSE and $MediaAE$ of all algorithms showed the same results. Moreover, the average of MAE, MSE, and $MediaAE$ of the ExtraTree were 0.10, 0.035, 0.045, respectively, which indicated the ExtraTree had the lowest errors compared with other algorithms.

6.6. Comparison of Accuracy and Efficiency

The purpose of this experiment was to compare the accuracy and efficiency of different ensemble learning algorithms for the throughput prediction of SPTs. Accuracy and execution time are formulated as follows:

$$Accuracy = \left(100 - \frac{1}{n}\sum_{i=1}^{n}\left|\frac{f_i - y_i}{y_i}\right|\right) \times 100\% \qquad (19)$$

where, n is the number of samples, f_i is the predicted throughput value, y_i is the actual throughput value.

$$ET = \frac{10000}{n}\sum_{i=1}^{n} t_i \qquad (20)$$

where, n is also the number of samples, and t_i is the prediction execution time for one sample. Since the execution time to predict one sample is short and not good for comparison, the constant coefficient 10,000 in Equation 20 is used to estimate the prediction time for per 10,000 samples. Table 5 summarized the experimental results from the benchmark WC, and ITE.

From Table 5, it was apparent that ExtraTree on all the benchmarks resulted in the highest values of accuracy and efficiency. For example, the accuracy of ExtraTree on all the benchmarks had the highest rates with 91.13%, and 89.94%. The ExtraTree had the lowest execution time with 0.82, and 0.85 s/10,000 samples. These results indicated that the ExtraTree had the highest accuracy and the highest efficiency compared with other algorithms on all benchmarks.

Table 5. Accuracy and efficiency of different algorithms.

Benchmarks	Accuracy (%)				ET (s/10,000 samples)			
	A	E	G	R	A	E	G	R
WC	68.36	**91.13**	75.59	67.84	0.96	**0.82**	0.85	0.82
ITE	70.61	**89.94**	77.55	69.71	0.95	**0.85**	0.86	0.86
AVG	69.485	**90.535**	76.57	68.775	0.955	**0.835**	0.855	0.84

Note: (A) AdaBoost; (E) ExtraTree; (G) GBDT; (R) Random Forest; (AVG) Average.

6.7. The Impact of Different Sample Ratio on Errors

To verify the generalization ability of the proposed method, the prediction errors of different training sample ratios were used. The experimental results were shown in Fig 7.

As shown in Fig 7, as the proportion of training samples increased, EV of all ensemble learning algorithms gradually approached 1, and MAE, MSE and $MediaAE$ also gradually approached zero. In addition, all ensemble learning algorithms was stable in terms of EV. Moreover, all algorithms showed a clear trend of decreasing errors. These results indicated that the generalization ability of ExtraTree was stable, and it had lower errors compared with other ensemble learning algorithms.

7. Conclusion

In this paper, the performance volatility of SPTs were studied and the corresponding volatility detection algorithm was proposed to accurately output the throughput of SPTs under a relatively stable state. In the throughput prediction phase, an ExtraTree-based algorithm was used for predicting the throughput. In the experiments, the prediction performance (i.e., errors, accuracy, and efficiency) of different ensemble learning algorithms on two benchmarks were compared. The results illustrated that the proposed algorithm had low error rates and high accuracy rates with a relatively high efficiency. Based on the research results, our future work will focus on performance prediction in heterogeneous environments, which requires a deeper study of the internal details of SPTs.

Acknowledgments. The authors are grateful to the anonymous referees for their insightful suggestions and comments. This research was supported by the National Natural Science Foundation of China under Grant Nos. 61862060, 61462079, 61562086 and 61562078, and the Doctoral Science and Technology Innovation Project in Xinjiang University under Grant No. XJUBSCX-201901.

References

1. Abe, G., Sato, K., Itoh, M.: Driver trust in automated driving systems: The case of overtaking and passing. IEEE Transactions on Human-Machine Systems 48(1), 85–94 (2017)
2. Borthakur, D.: The hadoop distributed file system: Architecture and design. Hadoop Project Website 11(2007), 21 (2007)
3. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)

4. Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., Tzoumas, K.: Apache flink: Stream and batch processing in a single engine. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 36(4) (2015)
5. Dasarathy, B.V., Sheela, B.V.: A composite classifier system design: concepts and methodology. Proceedings of the IEEE 67(5), 708–713 (1979)
6. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
7. Edwards, L.: Public relations, voice and recognition: a case study. Media, Culture & Society 40(3), 317–332 (2018)
8. Etter, M., Ravasi, D., Colleoni, E.: Social media and the formation of organizational reputation. Academy of Management Review 44(1), 28–52 (2019)
9. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences 55(1), 119–139 (1997)
10. Friedman, J.H.: Stochastic gradient boosting. Computational statistics & data analysis 38(4), 367–378 (2002)
11. Gedik, B., Schneider, S., Hirzel, M., Wu, K.L.: Elastic scaling for data stream processing. IEEE Transactions on Parallel and Distributed Systems 25(6), 1447–1463 (2013)
12. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning 63(1), 3–42 (2006)
13. Ghajargar, M., Wiberg, M., Stolterman, E.: Designing iot systems that support reflective thinking: A relational approach. International Journal of Design 12(1), 21–35 (2018)
14. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (iot): A vision, architectural elements, and future directions. Future generation computer systems 29(7), 1645–1660 (2013)
15. Handel, T., Schreiber, M., Rothmaler, K., Ivanova, G.: Data security and raw data access of contemporary mobile sensor devices. In: World Congress on Medical Physics and Biomedical Engineering 2018. pp. 397–400. Springer (2019)
16. Kulkarni, S., Bhagat, N., Fu, M., Kedigehalli, V., Kellogg, C., Mittal, S., Patel, J.M., Ramasamy, K., Taneja, S.: Twitter heron: Stream processing at scale. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. pp. 239–250. ACM (2015)
17. Moertini, V.S., Suarjana, G.W., Venica, L., Karya, G.: Big data reduction technique using parallel hierarchical agglomerative clustering. IAENG International Journal of Computer Science 45(1), 188 – 205 (2018)
18. Nasir, M.A.U., Morales, G.D.F., Garcia-Soriano, D., Kourtellis, N., Serafini, M.: The power of both choices: Practical load balancing for distributed stream processing engines. In: 2015 IEEE 31st International Conference on Data Engineering. pp. 137–148. IEEE (2015)
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12, 2825–2830 (2011)
20. Rossi, A., Vila, Y., Lusiani, F., Barsotti, L., Sani, L., Ceccarelli, P., Lanzetta, M.: Embedded smart sensor device in construction site machinery. Computers in Industry 108, 12–20 (2019)
21. Rossion, B.: Humans are visual experts at unfamiliar face recognition. Trends in cognitive sciences 22(6), 471–472 (2018)
22. Samosir, J., Indrawan-Santiago, M., Haghighi, P.D.: An evaluation of data stream processing systems for data driven applications. Procedia Computer Science 80, 439–449 (2016)
23. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine learning 37(3), 297–336 (1999)
24. Sun, D., Yan, H., Gao, S., Zhou, Z.: Performance evaluation and analysis of multiple scenarios of big data stream computing on storm platform. KSII Transactions on Internet & Information Systems 12(7) (2018)

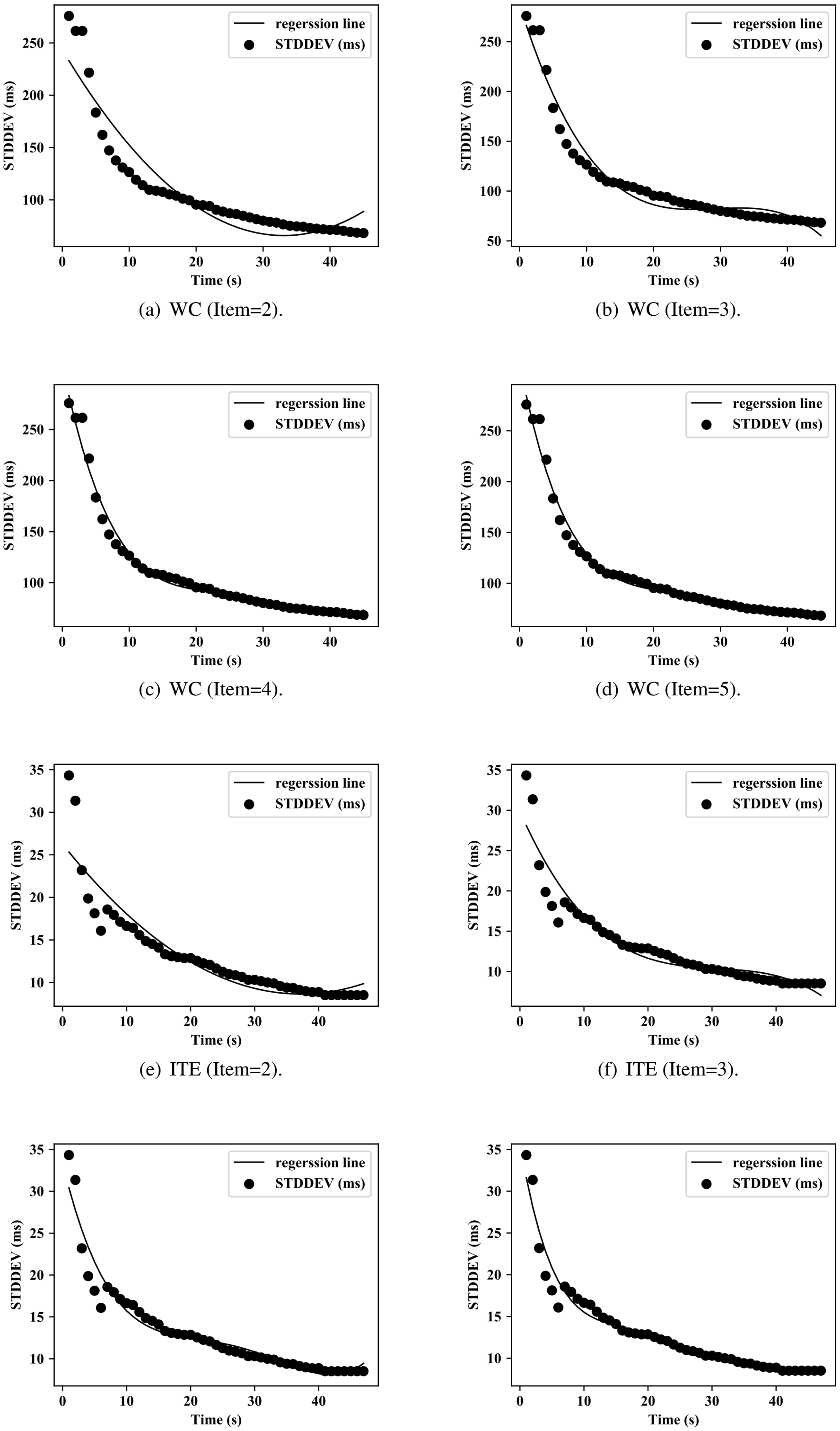
25. Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P., Murthy, R.: Hive: a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment 2(2), 1626–1629 (2009)
26. Toshniwal, A., Taneja, S., Shukla, A., Ramasamy, K., Patel, J.M., Kulkarni, S., Jackson, J., Gade, K., Fu, M., Donham, J., et al.: Storm@ twitter. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 147–156. ACM (2014)
27. Vora, M.N.: Hadoop-hbase for large-scale data. In: Proceedings of 2011 International Conference on Computer Science and Network Technology. vol. 1, pp. 601–605. IEEE (2011)
28. Wang, K., Khan, M.M.H.: Performance prediction for apache spark platform. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. pp. 166–173. IEEE (2015)
29. Xu, J., Chen, Z., Tang, J., Su, S.: T-storm: Traffic-aware online scheduling in storm. In: 2014 IEEE 34th International Conference on Distributed Computing Systems. pp. 535–544. IEEE (2014)
30. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine 13(3), 55–75 (2018)

Zheng Chu born in 1991. Ph.D. candidate in the School of Information Science and Engineering, Xinjiang University. His main research interests include distributed computing, in-memory computing, and machine learning.

Jiong Yu born in 1964. Professor and Ph.D. supervisor in the School of Information Science and Engineering, Xinjiang University. His main research interests include grid computing, parallel computing, etc.

Askar Hamdulla born in 1972. Professor and PhD supervisor in the School of Information Science and Engineering, Xinjiang University. His main research interest is natural language processing.

Received: January 31, 2020; Accepted: July 15, 2020.



(a) WC (Item=2).

(b) WC (Item=3).

(c) WC (Item=4).

(d) WC (Item=5).

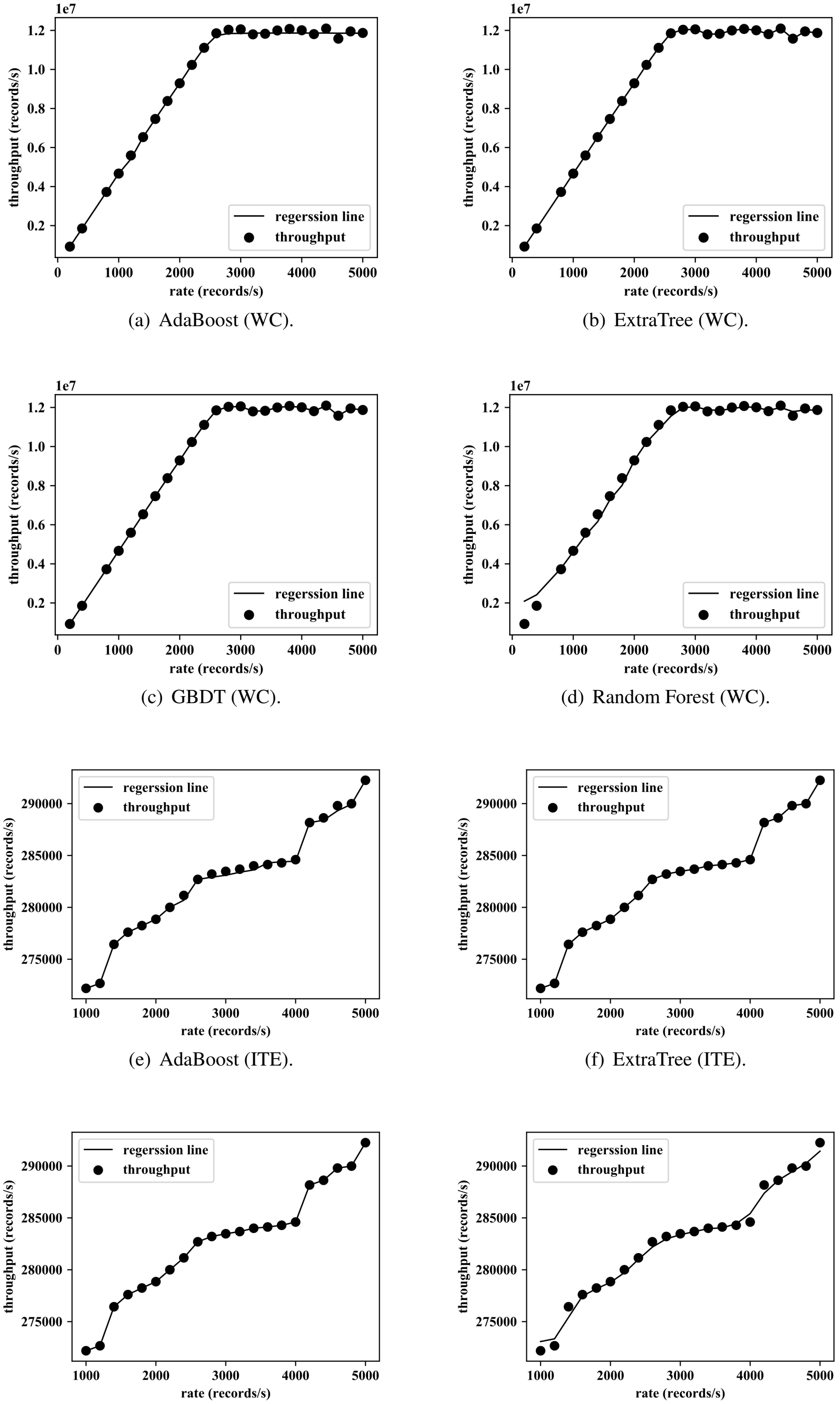
(e) ITE (Item=2).

(f) ITE (Item=3).

(g) ITE (Item=4).

(h) ITE (Item=5).

Fig. 5. Performance volatility regression using different regression Item. Note that STDDEV denotes the standard deviation of performance volatility.



(a) AdaBoost (WC).

(b) ExtraTree (WC).

(c) GBDT (WC).

(d) Random Forest (WC).

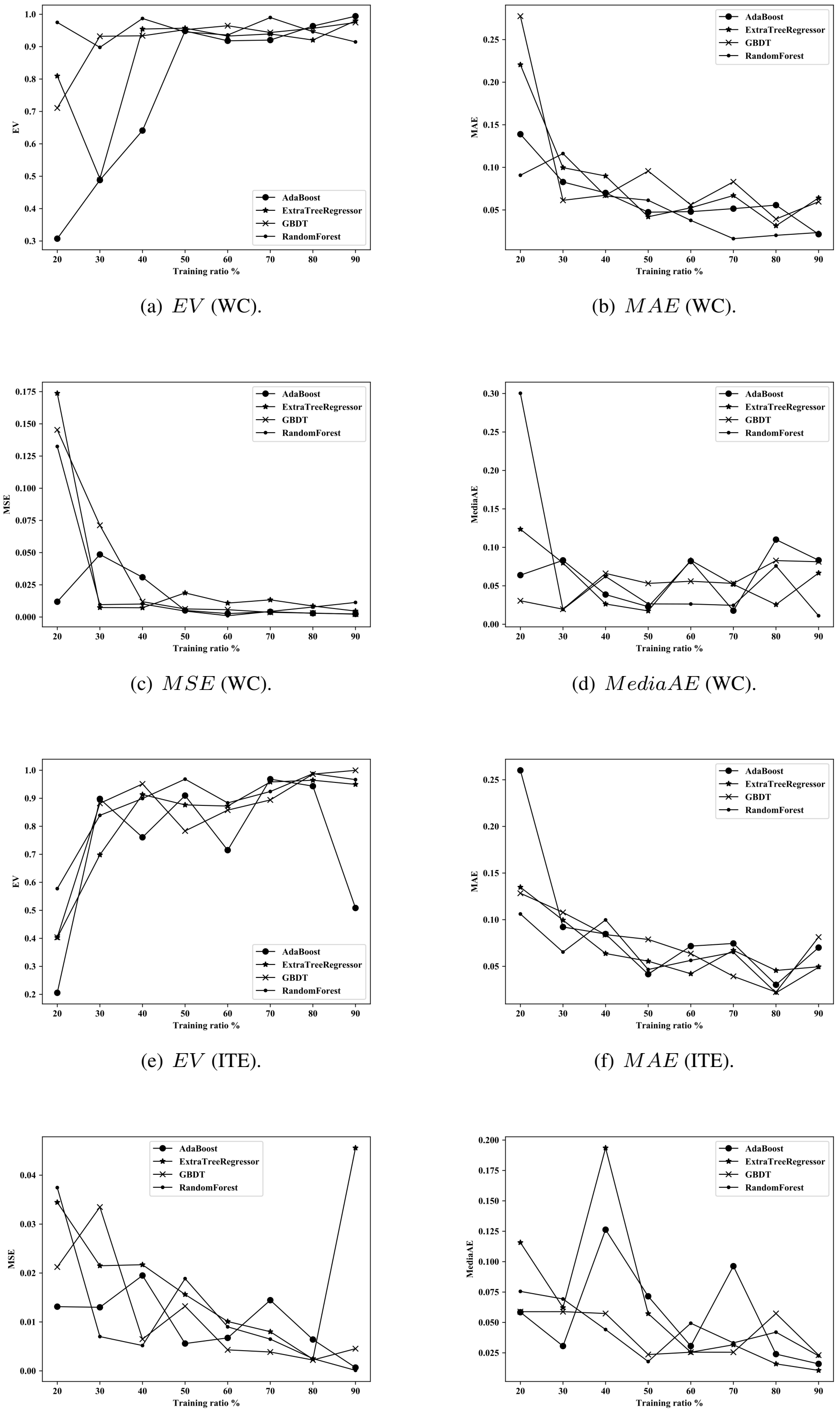
(e) AdaBoost (ITE).

(f) ExtraTree (ITE).

(g) GBDT (ITE).

(h) Random Forest (ITE).

Fig. 6. Throughput regression using different ensemble learning algorithms.



(a) EV (WC).

(b) MAE (WC).

(c) MSE (WC).

(d) $MediaAE$ (WC).

(e) EV (ITE).

(f) MAE (ITE).

(g) MSE (ITE).

(h) $MediaAE$ (ITE).

Fig. 7. Errors using different ensemble learning algorithms at different training data ratio.

Computer Science and Information Systems 18(1):23–42 https://doi.org/10.2298/CSIS200229041F

Multi-Objective Optimization of Container-Based Microservice Scheduling in Edge Computing

Guisheng Fan[1,2], Liang Chen[1], Huiqun Yu[1], and Wei Qi[1]

[1] Department of Computer Science and Engineering
East China University of Science and Technology, Shanghai, China
{gsfan,yhq}@ecust.edu.cn, chanliang_china@163.com
[2] Shanghai Key Laboratory of Computer Software Evaluating and Testing
Shanghai,China

Abstract. Edge computing provides physical resources closer to end users, becoming a good complement to cloud computing. With the rapid development of container technology and microservice architecture, container orchestration has become a hot issue. However, the container-based microservice scheduling problem in edge computing is still urgent to be solved. In this paper, we first formulate the container-based microservice scheduling as a multi-objective optimization problem, aiming to optimize network latency among microservices, reliability of microservice applications and load balancing of the cluster. We further propose a latency, reliability and load balancing aware scheduling (LRLBAS) algorithm to determine the container-based microservice deployment in edge computing. Our proposed algorithm is based on particle swarm optimization (PSO). In addition, we give a handling strategy to separate the fitness function from constraints, so that each particle has two fitness values. In the proposed algorithm, a new particle comparison criterion is introduced and a certain proportion of infeasible particles are reserved adaptively. Extensive simulation experiments are conducted to demonstrate the effectiveness and efficiency of the proposed algorithm compared with other related algorithms.

Keywords: edge computing, microservice, container orchestration, multi-objective optimization, particle swarm optimization.

1. Introduction

Recently, the emerging edge computing paradigm is seen as an effective solution to the problem of big data, which brings the processing to the edge of the network [1]. It has the advantage of shorter response time and can save bandwidth and energy required for data transmission in cloud computing [2, 3]. At the same time, microservice architecture [4] has become increasingly popular in the process of application design and development, which is commonly used to develop cloud native applications. However, there are few researches on microservice scheduling in edge computing.

As a lightweight virtualization technology, container is the perfect tool to encapsulate and deploy microservices. With the development of container technology and the widespread use of microservice architecture, some practical container scheduling strategies have been proposed. However, there are still some important problems to be solved in container-based microservice scheduling in edge computing. Current container cluster management tools, including Docker Swarm, Apache Mesos, and Google Kubernetes, only implement simple strategies of assigning containers to physical nodes. These strategies only consider physical resources usages [5], without implementing optimization strategies for the reliability of applications, network transmission latency, etc. It is possible for researchers to obtain better results in terms of network transmission latency, reliability of microservice applications and load balancing of the cluster.

Container scheduling in edge computing is a typical NP-hard problem. Such problems must be addressed using heuristic algorithms. Particle swarm optimization (PSO) is one of the most common heuristic algorithms. Many researchers have adopted PSO to solve the problem of task scheduling or scientific workflow scheduling in distributed computing. Thus, we propose a method to implement a container resource scheduling strategy by using PSO algorithm.

In order to tackle the container-based microservice scheduling problem in edge computing, we first formulate it as a multi-objective optimization problem, in which network transmission latency among microservices, reliability of microservice applications and load balancing of the cluster can be optimized. Then we propose a latency, reliability and load balancing aware scheduling algorithm for microservice scheduling system to determine the deployment of container-based microservices. The main contributions of this paper are as follows:

- We mathematically model the container-based microservice scheduling problem in edge computing to reduce network transmission latency among microservices, improve reliability of microservice applications and balance the cluster load, the resource capacity constraints of edge nodes are also considered.
- An LRLBAS algorithm based on particle swarm optimization (PSO) is proposed to solve the multi-objective optimization problem for container-based microservice scheduling. It can be used to separate the fitness function from constraints, so that each particle has two fitness values. The new comparison criterion for particles is introduced and a certain proportion of the infeasible particles are reserved adaptively.
- Several experiments are done to evaluate the proposed algorithm. The experiment results demonstrate that our algorithm generally outperforms the other two methods in terms of objectives, fitness value and optimization speed when the number of user requests is large. And it can obtain optimization results with relatively little running overhead when the number of user requests is small.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes the system architecture and analytical models. Section 4 provides the problem formulation. Section 5 presents the implementation of our LRLBAS algorithm. Section 6 illustrates the experimental settings and the experimental results. Section 7 summarizes this paper and raises the future work.

2. Related Work

Resource management optimization is a hot research topic in the field of distributed computing. In this paper, the related research is presented in three main parts: container orchestration, multi-objective optimization in resource management and scheduling methods based on particle swarm optimization (PSO) algorithm.

First, some related works on container orchestration are showed here. Adam et al. [6] present Two-stage Stochastic Programming Resource Allocator (2SPRA). It optimizes resource provisioning for containerized n-tier web services in accordance with fluctuations of incoming workload to accommodate predefined service-level objectives (SLOs) on response latency and reduces resource over-provisioning. Li et al. [7] propose an optimal minimum migration algorithm (OMNM) which reduces the unnecessary migration of containers. By fitting the growth rate of Docker containers in the source server, the model can estimate the growth trend of each Docker container and determine which container needs to be migrated. The algorithm aims to reduce the number of the migration and improve the utilization ratio of the resource, while ensuring the load balancing of the cluster. Kaewkasi and Chuenmuneewong [8] present a container scheduling algorithm based on Ant Colony Optimization (ACO), aiming to balance the resource usages and finally lead to the better performance of applications. Their approach is compared with the results obtained with a greedy algorithm. Guerrero et al. [9] propose a genetic algorithm approach, using the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) to optimize container allocation and elasticity management. Their approach is compared with the container management policies implemented in Google Kubernetes. Tao et al. [10] introduce a schedule algorithm based on fuzzy inference system (FIS), for global container resource allocation by evaluating nodes' statuses using FIS. The algorithm aims to derive optimal resource configurations and improve the performance of the cluster. However, only the paper [9] considers the use of microservice architecture, but Guerrero et al. do not include the network transmission latency among container-based microservices in their models.

Second, in the research of resource management optimization in distributed computing, there may exist multiple conflicting objectives, and researchers need to optimize these objectives simultaneously. Therefore, there have been many researches on multi-objective optimization methods in this field. Guerrero et al. [11] present an approach based on NSGA-II to optimize the deployment of microservice applications using containers in multi-cloud architectures. The optimization objectives are three: cloud service cost, network latency among microservices, and time to start a new microservice when a provider becomes unavailable. Azimzadeh and Biabani [12] present a multi-objective optimization method for resource management and task assignment based on genetic algorithm, in order to reduce execution time and enhance reliability of service. Langhnoja and Joshiyara [13] propose a novel scheduling algorithm called multi-objective based Integrated Task scheduling which aims to solve task scheduling problem of cloud computing, considering three optimization objectives: execution time, execution cost and load balancing. Mireslami et al. [14] propose a multi-objective resource allocation model when deploying a Web application in cloud, considering deployment cost and quality of service (QoS) simultaneously. The algorithm aims to minimize cost, maximize QoS and get a balanced trade-off between the two conflicting objectives. Zhang et al. [15] introduce an adaptive container scheduler based on integer linear programming, which considers three factors: the container host energy conservation, the container image pulling costs from the image registry to the container hosts, and the workload network transition costs from the clients to the container hosts. Lin et al. [16] establish a multi-objective optimization model for the container-based microservice scheduling, and propose an ant colony algorithm to solve the scheduling problem. The algorithm aims to optimize cluster service reliability, cluster load balancing, and network transmission overhead. However, all of these work above focus on resource management in cloud, rather than the emerging edge computing paradigm. The work of Lin et al. [16] is the most similar one to our approach, but they do not provide a rigorous mathematical representation of the data transmission latency among microservices.

Third, as an intelligent algorithm, particle swarm optimization (PSO) is one of the most commonly used scheduling algorithms in the field of resource scheduling. Zhang and Yang [17] propose a task scheduling algorithm based on an improved PSO, which can schedule efficiently, shorten the task completion time and improve the utilization of resources in cloud computing. Pan and Chen [18] establish a resource-task allocation model and propose an improved PSO algorithm to achieve resource load balancing in the cloud environment. Chou et al. [19] propose the dynamic power-saving resource allocation (DPRA) mechanism based on a particle swarm optimization algorithm, aiming to improve energy efficiency for cloud data centers. Verma et al. [20] propose a hybrid PSO algorithm based on non-dominance sort for handling the workflow scheduling problem with multiple objectives in the cloud. Li et al. [21] propose a security and cost aware scheduling (SCAS) algorithm based on PSO for heterogeneous tasks of scientific workflow in cloud, aiming to minimize the total workflow execution cost while meeting the deadline and risk rate constraints. Li et al. [22] propose a PSO-based container scheduling algorithm of Docker platform, which aims to solve the problem of insufficient resource utilization and load imbalance. The algorithm distributed application containers on Docker hosts, balance resource usage, and ultimately improve application performance. However, only the paper [22] focuses on container scheduling, the paper [17], [18], [19], [20], [21] focus on task scheduling or workflow scheduling. Moreover, when solving constrained optimization problems, only the paper [21] separates the fitness function from constraints and adopts a novel comparison criterion of particles.

Despite a large number of solutions and implementations mentioned above, researches on container-based microservice scheduling in edge computing environment are still very limited. In this paper, we describe it as a multi-objective optimization problem and an LRLBAS algorithm based on PSO is implemented to solve it. This paper aims to optimize the network transmission latency among microservices, the reliability of microservice applications and the load balancing of the cluster simultaneously.

3. System Architecture and Analytical Models

As shown in Fig. 1, the system is mainly divided into two layers. The User Layer is used to send service requests to microservice applications. The Edge Cloud Layer consists of physical resources that are used to process requests from users. Users send their requests to microservice applications deployed on Edge Cloud. Then, the physical resources are allocated to related microservices encapsulated in containers by microservice scheduling system (MSM).

Container-based microservice scheduling in edge computing can be characterized by properties from three components, i.e., application model, network model, and computation model. The application model refers to the container-based microservice application being scheduled, the network model describes the infrastructure used to execute the microservice application, and the computation model corresponds to what we attempt to

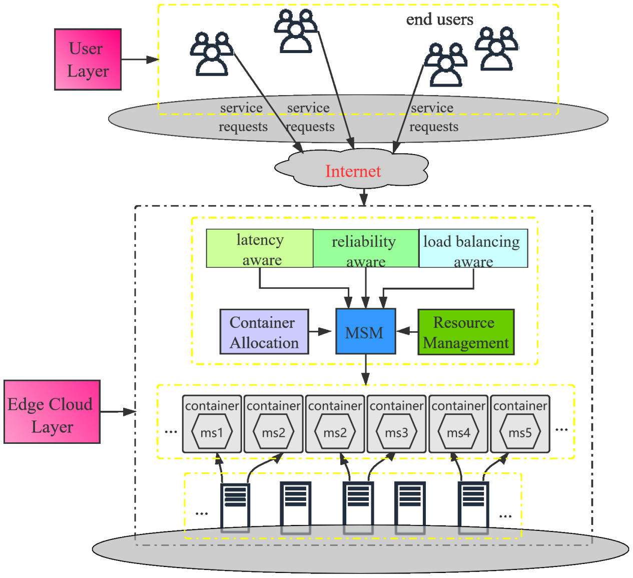


Fig. 1. System architecture

optimize. For convenience of reference, we summarize and tabulate the parameters and their descriptions used in the models in Table 1.

3.1. Application Model

We consider an application A developed by microservice architecture. A is modeled as a directed graph $G_a = \langle ms\_set, ms\_relation \rangle$, where $ms\_set = \{ms_1, ms_2, \ldots, ms_m\}$ is the set of microservices of application A; $ms\_relation$ is the set of dependencies among the microservices. When the execution of microservice ms_i requires the result generated by another microservice ms_k, the dependency between them is established, denoted by $(ms_i, ms_k) \in ms\_relation$.

Microservice ms_i is characterized as a tuple $\langle calc\_need_i, str\_need_i, max\_link_i \rangle$, where $calc\_need_i$ represents the computing resources required by one request for microservice ms_i; str_need_i is the storage resources required by one request for microservice ms_i; max_link_i is the maximum number of requests for one instance of microservice ms_i. In addition, pre_set_i is the preceeding set of microservices that provide data for microservice ms_i to execute, and when a microservice $ms_k \in pre\_set_i$, there exists $(ms_i, ms_k) \in ms\_relation$.

Application A receives service requests from users. User requests for microservice ms_k is mainly divided into two parts. One is the direct requests from users, denoted by $direct\_reqst_k$; the other is the indirect requests from other microservices. The number of indirect requests from microservice ms_i to ms_k is denoted by $link(ms_i, ms_k)$, so the total number of requests for microservice ms_k is calculated as $link_k = direct\_reqst_k +$

$\sum_{i=1 \wedge ms_k \in pre\_set_i}^{m} link(ms_i, ms_k)$; $trans(ms_i, ms_k)$ denotes the size of data transmitted in a request between microservice ms_i and ms_k. We do not consider the network transmission latency associated with direct requests from users to microservices. In addition, $scale_k$ is the number of instances of microservice ms_k in the cluster. According to the number of requests for microservice ms_k and the maximum number of requests for one instance of microservice ms_k, $scale_k$ can be calculated as $\lceil \frac{link_k}{max\_link_k} \rceil$.

3.2. Network Model

The underlying edge computing environment for running microservice applications is modeled as a fully-connected directed graph $G_e = \langle node\_set, link\_set \rangle$, where $node\_set$ $= \{node_1, node_2, \ldots, node_n\}$ is the set of edge nodes; $link\_set$ represents the set of directed links between edge nodes. Each communication link $l_{i,j}$ between $node_i$ and $node_j$ is related to bandwidth $b_{i,j}$ and network distance $d_{i,j}$.

Edge node $node_j$ is characterized as a tuple $\langle calc_j, str_j, fail_j \rangle$, where $calc_j$ is the computing resource capacity of edge node $node_j$; str_j indicates the storage resource capacity of edge node $node_j$; $fail_j$ represents the failure rate of edge node $node_j$.

3.3. Computation Model

Network Transmission Latency among Microservices. The network transmission latency among microservices is related to four key factors: the number of requests between two interoperable microservice instances, the size of data transmission in a request between the two microservice instances, the bandwidth and the network distance between the edge nodes where the two microservice instances are allocated. Considering that both consumer microservices and provider microservices may have multiple container-based instances, we allocate the requests between the two microservices evenly among their instance pairs. This is formalized in Equation (1):

$$trans\_latency = \sum_{k=1}^{m} \sum_{q=1}^{n} y_{k,q} \sum_{i=1 \wedge ms_k \in pre\_set_i}^{m} \sum_{p=1}^{n} y_{i,p} * lc(i,k,p,q), \tag{1}$$

where:

$$lc(i,k,p,q) = \frac{link(ms_i, ms_k)}{scale_i \times scale_k} \times (\frac{trans(ms_i, ms_k)}{b_{q,p}} + \frac{d_{q,p}}{c}). \tag{2}$$

Here, $lc(i,k,p,q)$ represents the network transmission latency between one instance pair of microservice ms_i and ms_k deployed on edge node $node_p$ and $node_q$ respectively, $y_{i,p}$ denotes the number of instances of microservice ms_i deployed on edge node $node_p$, and c is the propagation rate of the electromagnetic wave over the channel, approximately $3 \times 10^8 m/s$.

Average Number of Failures for Microservice Requests. The average number of failures for microservice requests measures the reliability of microservice applications, which is related to two key factors: the number of requests for microservices and the failure rates of edge nodes. Considering that edge nodes in the cluster may break down for some reason, microservices deployed on these edge nodes will not available and user requests will fail. This is mathematically modeled in Equation (4), which is used in [15]:

$$fail\_reqst = \sum_{j=1}^{n}\sum_{i=1}^{m} y_{i,j} \times fail_j \frac{link_i}{scale_i}. \tag{3}$$

where $y_{i,j}$ denotes the number of instances of microservice ms_i deployed on edge node $node_j$.

Imbalance Degree of Resource Usages of Edge Nodes. The imbalance degree of resource usages of edge nodes measures the load balancing of the cluster. In this paper, we consider the computing resources and storage resources simultaneously, so balancing cluster resource load is a Multi-Resource Load Balancing (MRLB) problem. To deal with the load balancing of the cluster, the standard deviations of utilization rate of physical resources in edge nodes are calculated, and then used as coefficient value for the utilization rate of corresponding resource in each node [15]. The maximum value of resource utilization rate with coefficient among edge nodes reflects the worst case about load blancing of the cluster. So, the imbalance degree of resource usages of edge nodes is formalized in Equation (4):

$$imbalance = \frac{Max(util_1, util_2, \ldots, util_j, \ldots, util_n)}{\sigma_{calc} + \sigma_{str}} \quad 1 \leq j \leq n, \tag{4}$$

where:

$$util_j = Max(\sigma_{calc} \times calc\_usage_j, \sigma_{str} \times str\_usage_j), \tag{5}$$

$$calc\_usage_j = \sum_{i=1}^{m} y_{i,j} \frac{link_i \times calc\_need_i}{scale_i \times calc_j}, \tag{6}$$

$$str\_usage_j = \sum_{i=1}^{m} y_{i,j} \frac{link_i \times str\_need_i}{scale_i \times str_j}. \tag{7}$$

Here,σ_{calc}, σ_{str} represents the standard deviation values of utilization rate of computing resources and storage resources of edge nodes in the cluster respectively; $util_j$ is the bigger value of resource utilization rate with coefficient of edge node $node_j$; $calc\_usage_j$, str_usage_j are the utilization rate values of computing resources and storage resources of edge node $node_j$.

4. Problem Formulation

4.1. Multi-Objective Optimization Model

According to the three objective functions mentioned in Section 3.3, we establish the following multi-objective optimization model of container-based microservice scheduling in edge computing.

$$min \quad trans\_latency, \tag{8}$$

$$min \quad fail\_reqst, \tag{9}$$

$$min \quad imbalance, \tag{10}$$

$subject \quad to:$

$$\sum_{i=1}^{m} y_{i,j} \frac{link_i}{scale_i} calc\_need_i - calc_j \leq 0 \quad \forall node_j, \tag{11}$$

$$\sum_{i=1}^{m} y_{i,j} \frac{link_i}{scale_i} str\_need_i - str_j \leq 0 \quad \forall node_j. \tag{12}$$

Equation (8)-(10) represent the three optimization objectives respectively: minimizing the network transmission latency among microservices, minimizing the average number of failing requests for microservices and minimizing the imbalance degree of resource usages of edge nodes. Equation (11)-(12) represent the computing and storage resource constraints of edge nodes respectively.

Table 1. Summary of parameters and their descriptions

Parameters	Descriptions
$G_a = \langle ms\_set, ms\_relation \rangle$	microservice application
m	the number of microservices in the application
ms_i	microservice with id. i
$(ms_i, ms_k) \in ms\_relation$	dependency link from microservice ms_i to ms_k
$calc\_need_i$	computing resources required by one request for microservice ms_i
str_need_i	storage resources required by one request for microservice ms_i
max_link_i	the maximum number of requests for one instance of microservice ms_i
pre_set_i	preceeding set of microservices of microservice ms_i
$direct\_reqst_i$	the number of direct requests for microservice ms_i from users
$link(ms_i, ms_k)$	the number of indirect requests from microservice ms_i to ms_k
$link_i$	the total number of requests for microservice ms_i
$trans(ms_i, ms_k)$	size of data transmitted between microservice ms_k and ms_i
$scale_i$	the number of instances of microservice ms_i
$G_e = \langle node\_set, link\_set \rangle$	edge computing environment
n	the number of edge nodes in the cluster
$node_j$	edge node with id. j
$calc_j$	computing resource capacity of edge node $node_j$
str_j	storage resource capacity of edge node $node_j$
$fail_j$	failure rate of edge node $node_j$
$l_{i,j}$	communication link between edge node $node_i$ and $node_j$
$b_{i,j}$	bandwidth of link $l_{i,j}$
$d_{i,j}$	network distance of link $l_{i,j}$

4.2. Fitness Function

Based on the aforementioned multi-objective optimization model, we use linear weighted sum method to modify the multi-objective optimization problem into a single-objective optimization problem and construct the fitness function of this paper as follows:

$$f(X) = w_1 \times \varphi(trans\_latency) + w_2 \times \varphi(fail\_reqst) + w_3 \times \varphi(imbalance). \quad (13)$$

where X is a scheduling scheme that maps microservices to edge nodes; $w_1, w_2, w_3 \geq 0$ and $w_1 + w_2 + w_3 = 1$. For the weight coefficients of optimization objectives, the most important objective generally has the maximum weight coefficient value according to the user preferences. $\varphi(l)$ is a normalized function for optimization objectives, defined as:

$$\varphi(l) = \frac{l - l_{min}}{l_{max} - l_{min}}. \quad (14)$$

In this paper, we repeatly do experiments on the three optimization objectives for 30 times, and replace the maximum and minimum values of the three objectives with their corresponding empirical constant values.

According to the analysis above, we formally define a container-based microservice scheduling problem in edge computing environment: given a directed graph structured microservice application $G_a = \langle ms\_set, ms\_relation \rangle$, a fully-connected edge computing environment $G_e = \langle node\_set, link\_set \rangle$, we wish to find a schedule X: $ms_i \rightarrow node_j$, $\forall ms_i \in ms\_set$, $\exists node_j \in node\_set$, such that minimizes the fitness function under the resource capacity of edge nodes. The problem can be formally described by Equation (15):

$$\begin{cases} find \quad X = \{x_1, x_2, \ldots, x_D\}, \\ which \quad min(f(X)), \\ subject \quad to: \sum_{i=1}^{m} y_{i,j} \frac{link_i}{scale_i} calc\_need_i - calc_j \leq 0 \quad \forall node_j, \\ \sum_{i=1}^{m} y_{i,j} \frac{link_i}{scale_i} str\_need_i - str_j \leq 0 \quad \forall node_j. \end{cases} \quad (15)$$

where D is the dimension of the schedule scheme X.

5. LRLBAS Algorithm Implementation

The above defined problem is a typical NP-hard problem. So, we consider using heuristic algorithms to obtain its near-optimal solution. Particle swarm optimization (PSO) is a frequently used heuristic algorithm, which is developed by Kennedy and Eberhart [23]. In this work, our latency, reliability and load balancing aware scheduling (LRLBAS) algorithm is based on PSO.

The basic idea of PSO is to search the optimal solution through the cooperation and information sharing among individuals in a population. Suppose that one population has N particles and the searching space is D dimensional. For a particle $P_i (i = 1, 2, \ldots, N)$, it has three typical parameters that are the position $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, velocity $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ and its optimal position $pbest_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. In the k_{th}

iteration of PSO, the velocity and position of particle P_i will be updated by the following two equations:

$$V_i^k = w^k \cdot V_i^{k-1} + c_1 \cdot r_1 \cdot (pbest_i - X_i^{k-1}) + c_2 \cdot r_2 \cdot (gbest - X_i^{k-1}). \tag{16}$$

$$X_i^k = X_i^{k-1} + V_i^k. \tag{17}$$

where c_1 and c_2 denote learning factors, r_1 and r_2 are random numbers from the range of [0,1], w^k is called inertia weight that influences search capability of particles, $gbest$ is the current global optimal position.

Shi and Eberhart [24] define the inertia weight w as a decreasing function, that is

$$w^k = w_{start} - (w_{start} - w_{end}) \cdot \frac{k}{M}. \tag{18}$$

where w_{start} and w_{end} are the initial value and ending value of inertia weight w, M indicates the maximum number of iterations in PSO.

5.1. Non-linear Inertia Weight

To better balance the global and local search abilities of particles, a novel method for updating the inertia weight w is used [25], as shown in Equation (19):

$$w^k = w_{end} + (w_{start} - w_{end}) \cdot \sin(\frac{\pi}{2}\sqrt{(1 - \frac{k}{M})^3}). \tag{19}$$

Compared with the linear inertia weight in Equation (18), the non-linear inertia weight in Equation (19) is larger at the beginning period, which can promote global search in the early stage of the optimization process. When the number of iterations gradually approaches the maximum value M, the non-linear inertia weight is smaller than the linear inertia weight, which can improve local search in the late stage of the optimization process.

5.2. Constraints Handing

To deal with the constraints, we adopt a strategy similar to that in [20]. Our constraint handling strategy separates the fitness function from constraints, so that each particle has two fitness values. In addition, a new comparison criterion for particles is introduced and a certain proportion of the infeasible particles are reserved adaptively.

In this paper, the general form of our problem is expressed in Equation (20):

$$min \quad f(X) \quad s.t. \quad g_j(X) \leq 0 \quad j = 1, 2, \ldots, q \tag{20}$$

After separating the fitness function from constraints, the original problem can be transformed into Equation (21):

$$fitness(i) = f(X), \quad violation(i) = \sum_{j=1}^{q} max(0, g_j(x)) \quad i = 1, 2, \ldots, N \tag{21}$$

Here, the former formula represents the fitness value of particle P_i in a certain iteration, namely the first fitness value; the latter is the constraint violation value of particle P_i, that is, the second fitness value. The constraint violation value of a feasible solution is 0.

Then, we use the following comparison criteria for particles: firstly, a constant β is given. (1) Between two feasible particles P_i and P_j, compare their fitness values $fitness(i)$ and $fitness(j)$, the smaller one is better; (2) between two infeasible particles P_i and P_j, compare their constraint violation values $violation(i)$ and $violation(j)$, the smaller one is better; (3) between the feasible particle P_i and the infeasible particle P_j, if $violation(j)$ is smaller than β , then compare their fitness values $fitness(i)$ and $fitness(j)$, the smaller one is better; otherwise, particle P_i is better.

During the optimization process, the proportion of infeasible solutions changes dynamically. If the proportion becomes too large, most particles will move towards infeasible solutions. If the proportion becomes too small, our algorithm will not work very well and the optimization efficiency will be compromised. So, we hope the proportion of infeasible solutions can fluctuate around a fixed value p. Based on the above comparison criteria, we can know that the larger the value of constant β, the larger the proportion of infeasible solutions is likely to be. To keep the proportion around p, the following adaptive adjustment strategy for β is used: (1) when the proportion is smaller than p, $\beta = 1.5\beta$; (2) when the proportion is larger than p, $\beta = 0.5\beta$; (3) when the proportion is equal to p, the value of β does not change.

5.3. Algorithm Implementation

Based on implementation steps mentioned above, we design the LRLBAS algorithm based on PSO for microservice applications. The implementation of our algorithm is shown as the pseudo-code of Algorithm 1.

This algorithm first initializes position (i.e., the scheduling scheme), velocity of all particles and other necessary parameters (see lines 1-7). Next, update velocity, position, inertia weight and the value of β (see lines 10-14). Then, evaluate the fitness value and constraint violation value of each particle according to constraints handling, update the optimal solution of each particle and select the global optimal solution (see lines 16-19). Finally, the algorithm returns the near-optimal scheduling scheme (see line 21).

6. Performance Evaluation

6.1. Experimental Setup

In this paper, the test data set is shown in Table 2 and Table 3. The microservice application in this test data set is composed of 17 microservices.

Table 2 shows the number of requests and the amount of data transmission among microservices when the microservice application receives a unit of user service requests (represented as 1.0reqs). Here, (-,ms_i) represents users consume microservice ms_i directly. For convenience, we use $link_{i,k}$ and $trans_{i,k}$ to denote $link(ms_i, ms_k)$ and $trans(ms_i, ms_k)$, respectively. Table 3 shows the parameters of microservices in the application; $link_i$ represents the number of requests for microservice ms_i when the microservice application receives 1.0reqs; $scale_i$ is the number of instances of microservice ms_i in the cluster.

Some details about the experimental setup are shown in Table 4. Table 4(a) presents parameters of our LRLBAS algorithm. Parameter settings for the edge node cluster are described in Table 4(b).

Algorithm 1 LRLBAS algorithm based on PSO

```
Input: related information about the microservice application, a set of edge nodes, maximum number of iteration $M$, size of particle swarm $N$.
Output: the near-optimal scheduling solution $X_{best}$.
for $i = 1$ to $N$ do
    Randomize the initialization of scheduling $X_i$, search velocity $V_i$ and some other necessary parameters;
end for
for $i = 1$ to $N$ do
    Set current position of scheduling $X_i$ as $pbest_i$;
end for
Select the best near-optimal scheduling plan of minimum fitness from $N$ scheduling plans as $gbest$;
for $j = 1$ to $M$ do
    for $i = 1$ to $N$ do
        Update the velocity of particle $V_i$ by Equation (16);
        Update the position of particle $X_i$ by Equation (17);
    end for
    Update the inertia weight $w^k$ by Equation (19);
    Compute the ratio of infeasible solutions and update;
    for $i = 1$ to $N$ do
        Evaluate the fitness value and the violation value of scheduling plan $X_i$ according to constraints handling;
        Compare the current particles fitness evaluation with $pbest_i$. If current value is better than $pbest_i$, then update $pbest_i$;
    end for
    Select the best near-optimal scheduling plan of minimum fitness from $N$ scheduling plans as $gbest$;
end for
return $X_{best}$
```

In addition, this paper assume that three objectives are equally important, so their weight coefficients w_1, w_2, w_3 are all set as 1/3.

6.2. The Comparison of both Objectives and Fitness Value

In this paper, we compare the LRLBAS algorithm with other scheduling algorithms including the original PSO (OPSO) based scheduling algorithm and the directional and non-local-convergent PSO (DNCPSO) based scheduling algorithm proposed in [25]. The main principles and steps of the two algorithms are shown below.

(1) Original PSO (OPSO). The original PSO has been described in Section 4 and proved to be a useful intelligent heuristic algorithm. It searches the optimal solution through the cooperation and information sharing among individuals in a population.

(2) Directional and non-local-convergent PSO (DNCPSO). The authors in [25] propose a directional and non-local-convergent particle swarm optimization algorithm to perform workflow scheduling in cloud-edge environment. This algorithm firstly uses

Table 2. Number of requests and amount of data transmission under 1.0reqs

(ms_i, ms_k)	$link_{i,k}$	$trans_{i,k}$(MB)	(ms_i, ms_k)	$link_{i,k}$	$trans_{i,k}$(MB)
(-,ms_1)	25	0	(ms_7, ms_{14})	5	2.1
(-,ms_3)	35	0	(ms_8, ms_{14})	8	2.1
(-,ms_6)	4	0	(ms_9, ms_5)	10	1.8
(-,ms_7)	15	0	(ms_9, ms_{11})	10	2.4
(-,ms_{10})	50	0	(ms_{10}, ms_5)	10	1.7
(-,ms_{13})	15	0	(ms_{10}, ms_9)	13	2.2
(ms_1, ms_2)	10	2.3	(ms_{10}, ms_{11})	10	2.5
(ms_1, ms_4)	5	1.6	(ms_{11}, ms_2)	10	1.6
(ms_1, ms_9)	10	2.0	(ms_{12}, ms_8)	23	3.2
(ms_2, ms_4)	5	1.8	(ms_{13}, ms_2)	10	2.3
(ms_2, ms_{12})	8	3.0	(ms_{13}, ms_8)	23	3.1
(ms_3, ms_{13})	30	0.9	(ms_{13}, ms_{16})	4	2.8
(ms_4, ms_{15})	15	2.8	(ms_{13}, ms_{17})	15	1.2
(ms_4, ms_{16})	4	2.9	(ms_{15}, ms_{16})	4	2.6
(ms_5, ms_{15})	15	2.7	(ms_{16}, ms_{14})	8	2.2
(ms_7, ms_2)	10	2.4	(ms_{17}, ms_{12})	8	3.1

Table 3. Microservices in the application

ms_i	pre_set_i	$calc\_need_i$	str_need_i	max_link_i	$link_i$	$scale_i$
ms_1	$\{ms_2, ms_4, ms_9\}$	2.1	1.4	10	25	3
ms_2	$\{ms_4, ms_{12}\}$	0.5	3.2	8	40	5
ms_3	$\{ms_{13}\}$	3.1	1.6	8	35	5
ms_4	$\{ms_{15}, ms_{16}\}$	4.7	0.2	5	10	2
ms_5	$\{ms_{15}\}$	1.8	3.1	8	20	3
ms_6	$\{\}$	2.5	5.1	4	4	1
ms_7	$\{ms_2, ms_{14}\}$	6.2	0.6	4	15	4
ms_8	$\{ms_{14}\}$	0.8	6.2	4	45	12
ms_9	$\{ms_5, ms_{11}\}$	3.9	2.3	5	23	5
ms_{10}	$\{ms_5, ms_9, ms_{11}\}$	0.2	4.8	4	50	13
ms_{11}	$\{ms_2\}$	2.8	2.6	8	20	3
ms_{12}	$\{ms_8\}$	5.3	0.9	4	15	4
ms_{13}	$\{ms_2, ms_8, ms_{16}, ms_{17}\}$	0.6	4.8	5	45	9
ms_{14}	$\{\}$	6.1	2.5	4	20	5
ms_{15}	$\{ms_{16}\}$	1.2	4.2	5	30	6
ms_{16}	$\{ms_{14}\}$	5.4	1.6	4	12	3
ms_{17}	$\{ms_{12}\}$	3.7	2.2	6	15	3

Table 4. Parameter settings

(a) Parameters of the LRLBAS algorithm

Parameter	Value
Population size	50
Maximum number of iterations	300
w_{start}	0.9
w_{end}	0.4
c_1, c_2	2
r_1, r_2	[0,1]
β	10
p	0.2

(b) Parameters of the edge node cluster

Parameter	Value
Number of edge nodes	120
$calc_i$	$\{100, 200, 400\}$
str_i	$\{100, 200, 400\}$
$fail_i$	$\{0.01, 0.02, 0.03\}$
$b_{i,j}$(Mbps)	$\{200, 400\}$
$d_{i,j}$(km)	[30,300]

non-linear inertia weight to better balance the global and local search abilities of particles. Then, it replace random search with directional search which can improve the optimization speed of the algorithm. Finally, selection and mutation operations are integrated into this algorithm, which is conducive to jump out of local optimum. So, this algorithm can get better near-optimal solution in a faster speed.

In this subsection,the performance comparisons of the three algorithms are performed in four aspects: network transmission latency, reliability of the microservice application, cluster load balancing and fitness value. We present the experimental results of three algorithms in the above four aspects under five experimental configurations. The number of user requests of the five experimental configurations varies between 1.0reqs and 3.0reqs, with an interval of 0.5reqs.

As shown in Fig. 2, the values have been normalized between 0.0 and 1.0. We can see that LRLBAS algorithm achieves better performance (smaller objective values) than OPSO in four aspects under five experimental configurations, and obtains better optimization results than DNCPSO in 12 of the total 20 scenarios.

In detail, as shown in Fig. 2(d-e), LRLBAS algorithm performs better than DNCPSO in all four aspects under 2.5reqs and 3.0reqs. However, in Fig. 2(a-c), LRLBAS algorithm obtains objective values that are slightly higher than DNCPSO in 7 scenarios. Because the proportion of infeasible solutions in the population is small when the number of user requests is small. So the infeasible solutions are not enough for LRLBAS algorithm to find better solutions.

6.3. The Comparison of Optimization Process for Fitness Value

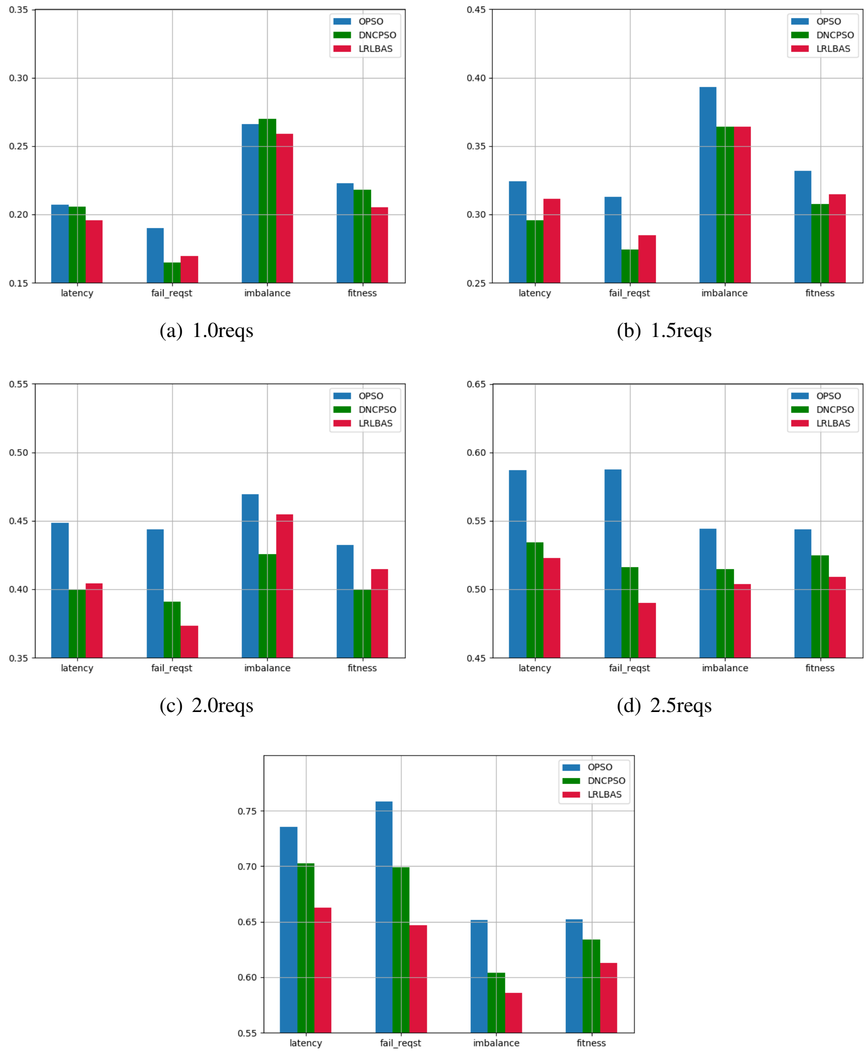
In this subsection, we compare the LRLBAS algorithm with other algorithms by the iterative trend of fitness value under five experimental configurations. The iterative trend of each algorithm for searching results includes two aspects, namely searching speed and nearest optimal solution. The searching speed indicates the fewest number of iterations that is required to find the near-optimal solution. The nearest optimal solution indicates the minimum fitness value that the algorithms can reach.

As shown in Fig. 3, LRLBAS algorithm can obtain smaller fitness values than the other two algorithms in most cases. Among the three algorithms, OPSO performs worst under five experimental configurations.

In detailed comparison, as shown in Fig. 3(c), the fitness value in DNCPSO declines significantly faster than that in our LRLBAS algorithm, indicating that DNCPSO can obtain the near-optimal solution with fewer iterations than LRLBAS algorithm. Moreover, DNCPSO obtains a smaller fitness value at the end of the iteration process. So DNCPSO performs better than LRLBAS algorithm under 2.0reqs. In addition, as shown in Fig. 3(d-e), LRLBAS algorithm performs better than DNCPSO in terms of searching speed and nearest optimal solution. The experimental results in Fig. 3 are consistent with the optimization results of fitness value in Fig. 2.

6.4. The Comparison of Sensitivity

As shown in Fig. 4, the values have been normalized between 0.0 and 1.0. We can see that our LRLBAS algorithm is the least sensitive algorithm as its curve slope has the



(a) 1.0reqs

(b) 1.5reqs

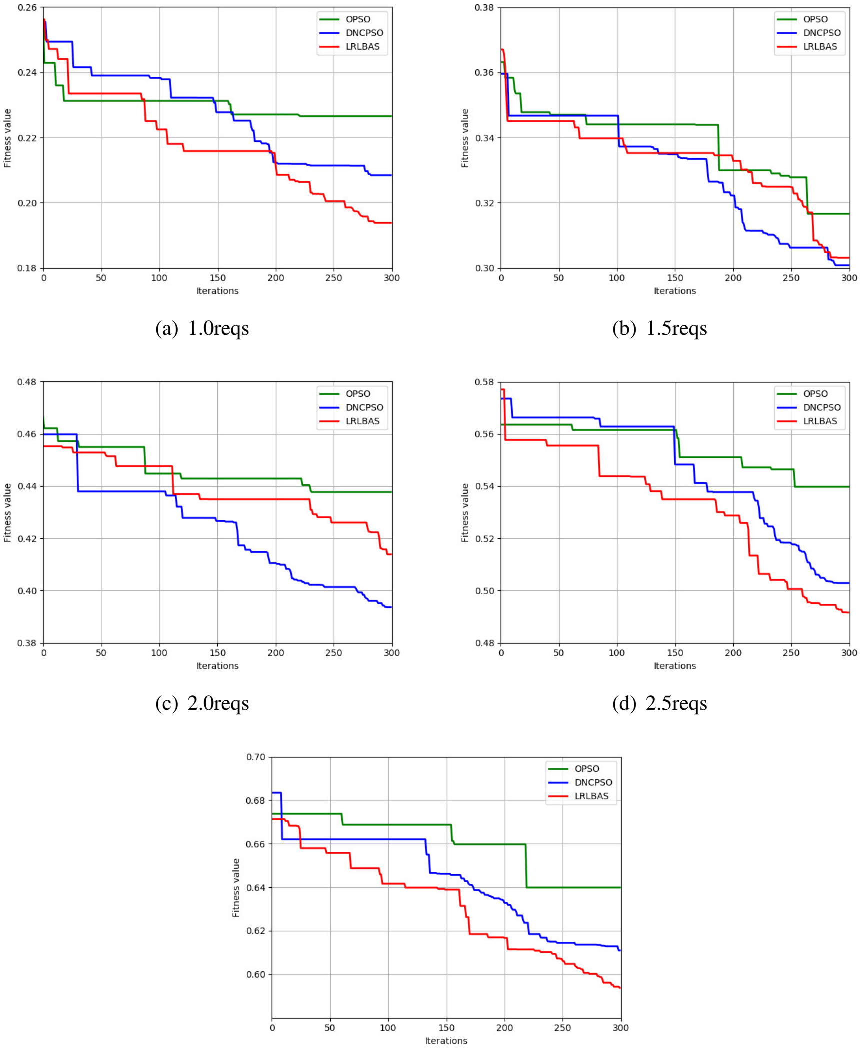
(c) 2.0reqs

(d) 2.5reqs

(e) 3.0reqs

Fig. 2. Normalized objective and fitness values obtained with three algorithms

least obvious change in terms of both objectives and fitness value as the number of user requests increases, which has more adaptability to the situation that the number of user requests increases. Also, LRLBAS can obtain smaller objective and fitness values in most cases, which is consistent with the experimental results in Fig. 2.



(a) 1.0reqs

(b) 1.5reqs

(c) 2.0reqs

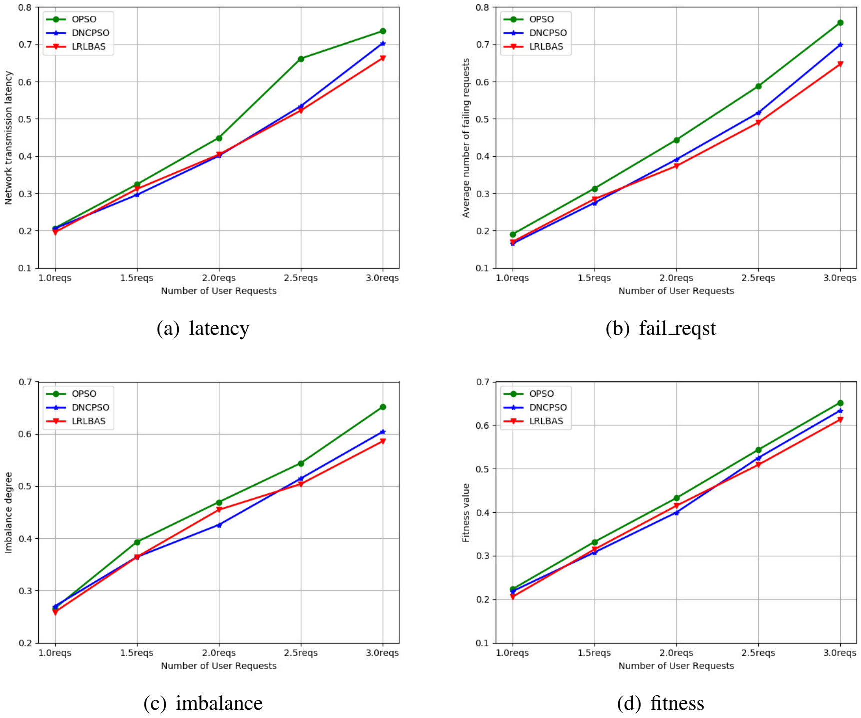
(d) 2.5reqs

(e) 3.0reqs

Fig. 3. The changing process of fitness value under different iterations

6.5. The Comparison of Running Overhead for Fitness Value

In this paper, the running time required to perform a optimization process for fitness value is used as the evaluation metric of algorithms running overhead. Here, The final result of the running time is calculated by running an average of 30 times. We compare our LRL-BAS algorithm with other algorithms by the running overhead under five experimental configurations.



(a) latency

(b) fail_reqst

(c) imbalance

(d) fitness

Fig. 4. The comparison of sensitivity among three algorithms

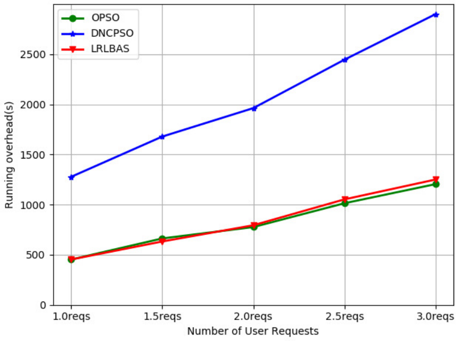


Fig. 5. Running overhead under different number of user requests

As shown in Fig. 5, we can see that the running overhead of the LRLBAS algorithm is nearly equal to that of OPSO, while the running overhead of DNCPSO is much higher than that of OPSO. As mentioned above, DNCPSO performs significantly better than our LRLBAS algorithm under 2.0reqs, but it costs significant running overhead.

Through the above several groups of experiments, it can demonstrate that LRLBAS algorithm achieves better optimization results than the other two algorithms in terms of objectives, fitness value and optimization speed when the number of user requests is large. When the number of user requests is small, although LRLBAS algorithm performs worse than DNCPSO in some cases, it consumes significantly less running overhead than DNCPSO. Therefore, it can be proved that the LRLBAS algorithm for container-based microservice scheduling in edge computing proposed in this paper is effective and efficient.

7. Conclusion

In this paper, container-based microservice scheduling in edge computing is described as a multi-objective optimization problem, aiming to reduce the network transmission latency among microservices, improve the reliability of microservice applications and balance the cluster load. We propose a latency, reliability and load balancing aware scheduling algorithm for microservice applications in edge computing. Our proposed algorithm is based on the PSO. Extensive experiments demonstrate the effectiveness and efficiency of our algorithm for microservice scheduling in edge computing.

In the future, we plan to take other optimization objectives into account. In addition, more scheduling algorithms can be added for performance comparison. Finally, we can study the results of our microservice scheduling algorithm in a real edge computing container cluster.

Acknowledgements. This work was partially supported by the NSF of China under Grant nos. 61702334 and 61772200, Shanghai Municipal Natural Science Foundation under Grant nos. 17ZR1406900 and 17ZR1429700, Action Plan for Innovation on Science and Technology Projects of Shanghai under Grant no. 16511101000, Collaborative Innovation Foundation of Shanghai Institute of Technology under Grant no. XTCX2016-20, and Educational Research Fund of ECUST under Grant no. ZH1726108.

References

1. Khan, W.Z., Ahmed, E., Hakak, S., Yaqoob, I., Ahmed, A.: Edge computing: A survey. Future Gener. Comput. Syst. 97, 219-235 (2019)
2. Mukherjee, M., Shu, L., Wang, D.: Survey of fog computing: fundamental, network applications, and research challenges. IEEE Commun. Surv. Tutorials 20(3), 1826-1857 (2018)
3. Souza, A., Wen, Z., Cacho, N., Romanovsky, A., James, P, Ranjan, R.: Using osmotic services composition for dynamic load balancing of smart city applications. In: Proceedings of 2018 IEEE 11th International Conference on Service-Oriented Computing and Applications. Paris, pp. 145-152 (2018)
4. Lewis, J., Fowler., M.: Microservices: a definition of this new architectural term. [Online]. Available: https://www.martinfowler.com/articles/microservices.html (2014)
5. Fazio, M., Celesti, A., Ranjan, R., Liu, C., Chen, L., Villari, M.: Open issues in scheduling microservices in the cloud. IEEE Cloud Comput. 3(5), 81-88 (2016)
6. Adam, O., Lee, Y.C., Zomaya., A.Y.: Stochastic resource provisioning for containerized multi-tier web services in clouds. IEEE Trans. Parallel Distrib. Syst. 28(7), 2060-2073 (2017)

7. Li, P., Nie, H., Xu, H., Dong, L.: A minimum-aware container live migration algorithm in the cloud environment. Int. J. Bus. Data Commun. Netw. 13(2), 15-27 (2017)
8. Kaewkasi, C., Chuenmuneewong, K.: Improvement of container scheduling for docker using ant colony optimization. In: 2017 9th International Conference on Knowledge and Smart Technology. Chonburi, pp. 254-259 (2017)
9. Guerrero, C., Lera, I., Juiz., C.: Genetic algorithm for multi-objective optimization of container allocation in cloud architecture. J. Grid Comput. 16(1), 113-135 (2018)
10. Tao, Y., Wang, X., Xu, X., Chen, Y.: Dynamic resource allocation algorithm for container-based service computing. In: 2017 IEEE 13th International Symposium on Autonomous Decentralized System. Bangkok, pp. 61-67 (2017)
11. Guerrero, C., Lera, I., Juiz., C.: Resource optimization of container orchestration: A case study in multi-cloud microservices-based applications. J. Supercomput. 74(7), 2956-2983 (2018)
12. Azimzadeh, F., Biabani., F.: Multi-objective job scheduling algorithm in cloud computing based on reliability and time. In: 2017 3th International Conference on Web Research. Tehran, pp. 96-101 (2017)
13. Langhnoja, H.K., Hetal Joshiyara, P.A.: Multi-objective based integrated task scheduling in cloud computing. In: 2019 3rd International conference on Electronics, Communication and Aerospace Technology. Coimbatore, pp. 1306-1311 (2019)
14. Mireslami, S., Rakai, L., Far, B.H., Wang, M.: Simultaneous cost and QoS optimization for cloud resource allocation. IEEE Trans. Netw. Service Manage. 14(3), 676-689 (2017)
15. Zhang, D., Yan, B., Feng, Z., Zhang, C., Wang, Y.: Container oriented job scheduling using linear programming model. In: 2017 3th International Conference on Information Management. Chengdu, pp. 174-180 (2017)
16. Lin, M., Xi, J., Bai, W., Wu, J.: Ant colony algorithm for multi-objective optimization of container-based microservice scheduling in cloud. IEEE Access 7, 83088-83100 (2019)
17. Zhang, Y., Yang, R.: Cloud computing task scheduling based on improved particle swarm optimization algorithm. In: Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society. Beijing, pp. 8768-8772 (2017)
18. Pan, K., Chen, J.: Load balancing in cloud computing environment based on an improved particle swarm optimization. In: 2015 6th IEEE International Conference on Software Engineering and Service Science. Beijing, pp. 595-598 (2015)
19. Chou, L., Chen, H., Tseng, F., Chao, H., Chang, Y.: DPRA: Dynamic power-saving resource allocation for cloud data center using particle swarm optimization. IEEE Syst. J. 12(2), 1554-1565 (2018)
20. Verma, A., Kaushal S.: A hybrid multi-objective particle swarm optimization for scientific workflow scheduling. Parallel Comput. 62, 1-19 (2017)
21. Li, Z., Ge, J., Yang, H., Huang, L., Hu, H., Hu, H., Luo, B.: A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. Future Gener. Comput. Syst. 65, 140-152 (2016)
22. Li, L.W., Chen, J.X., Yan, W.Y.: A particle swarm optimization-based container scheduling algorithm of docker platform. In: Proceedings of 2018 4th International Conference on Communication and Information Processing. Qingdao, pp. 12-17 (2018)
23. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, pp. 1942-1948 (1995)
24. Shi, Y., Eberhart R.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation. Washington, pp. 1945-1950 (1999)
25. Xie, Y., Zhu, Y., Wang, Y., Cheng, Y., Xu, R., Sani, A.S., Yuan, D., Yang, Y.: A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud-edge environment. Future Gener. Comput. Syst. 97, 361-378 (2019)

Guisheng Fan received his B.S. degree from Anhui University of Technology in 2003, M.S.degree from East China University of Science and Technology (ECUST) in 2006, and Ph.D. degree from ECUST in 2009, all in computer science. He is presently a research assistant of the Department of Computer Science and Engineering at ECUST. His research interests include formal methods for complex software system, service oriented computing, and techniques for analysis of software architecture.

Liang Chen received his B.S. degree in computer science from Nanjing Tech University in 2018. He is currently a graduate student at East China University of Science and Technology (ECUST). His research interests include software engineering, edge computing and microservice.

Huiqun Yu received his B.S. degree from Nanjing University in 1989, M.S.degree from East China University of Science and Technology (ECUST) in 1992, and PhD.degree from Shanghai Jiaotong University in 1995, all in computer science. He is currently a Professor of computer science with the Department of Computer Science and Engineering at ECUST. From 2001 to 2004, he was a Visiting Researcher in the School of Computer Science at Florida International University. His research interests include software engineering, high confidence computing systems, cloud computing and formal methods. He is a senior member of IEEE, China Computer Federation.

Wei Qi received her B.S. degree in computer science from East China University of Science and Technology (ECUST) in 2018. She is currently a graduate student at ECUST. Her research interests include software engineering, cyber-physical system.

Received: February 29, 2020; Accepted: July 20, 2020.

Computer Science and Information Systems 18(1):43–66 https://doi.org/10.2298/CSIS200301042M

PureEdgeSim: A Simulation Framework for Performance Evaluation of Cloud, Edge and Mist Computing Environments

Charafeddine Mechalikh, Hajer Taktak, and Faouzi Moussa

University of Tunis El Manar, Faculty of Sciences of Tunis,
LIPAH-LR11ES14, 2092Tunis, Tunisia
{charafeddine.mechalikh, taktakhajer, faouzimoussa}@gmail.com

Abstract. Edge and Mist Computing are two emerging paradigms that aim to reduce latency and the Cloud workload by bringing its applications close to the Internet of Things (IoT) devices. In such complex environments, simulation makes it possible to evaluate the adopted strategies before their deployment on a real distributed system. However, despite the research advancement in this area, simulation tools are lacking, especially in the case of Mist Computing [11], where heterogeneous and constrained devices cooperate and share their resources. Motivated by this, in this paper, we present *PureEdgeSim*, a simulation toolkit that enables the simulation of Cloud, Edge, and Mist Computing environments and the evaluation of the adopted resources management strategies, in terms of delays, energy consumption, resources utilization, and tasks success rate. To show its capabilities, we introduce a case study, in which we evaluate the different architectures, orchestration algorithms, and the impact of offloading criteria. The simulation results show the effectiveness of *PureEdgeSim* in modeling such complex and dynamic environments.

Keywords: Simulation, modeling, tasks orchestration, load balancing, Mist Computing, Edge Computing.

1. Introduction

With the emergence of IoT, connected devices are gradually invading our daily lives with increasingly broad fields of application: personal health equipment, smart buildings, smart grids, connected vehicles, etc. A recent study estimates that the number of connected devices will exceed 38.6 billion by 2025, with economic benefits in the health, energy, transportation, and construction sectors [1]. However, due to this growth, Cloud Computing has faced many challenges. Not only has it become unable to support the growing number of IoT devices and the data they continually generate, but it is also unable, due to its remote location, to meet their quality of service requirements such as low latency. To face this, a new paradigm is needed. The latter must provide computing, storage, and services like the conventional Cloud and meet the quality of service requirements of IoT applications such as low latency, high scalability, and mobility.

This need for a new computing paradigm gave birth to Edge and Mist Computing. While Edge Computing covers a wide range of applications such as Fog Computing, Mobile Edge Computing, and Cloudlets all of which extend the Cloud by providing resources in the network layer of the IoT architecture [10, 16], Mist Computing allows resources to be harvested through the computation and communication capabilities offered in the perception layer [11]. As a result, most of the data generated by these devices can be processed locally, which reduces the latency, increases scalability, and minimizes energy consumption by saving the energy that would have been used to transfer data. However, in these complex and distributed environments, many issues need to be solved (e.g., load balancing, application placement, and resource discovery) and experimenting on a real distributed environment or testbeds [24] is not practical due to the cost and limited scalability.

The simulation makes it possible to evaluate the performance of the proposed approaches in a repeatable and controllable manner before their actual deployment in a real distributed system. Nevertheless, due to their heterogeneous, dynamic, and distributed nature, the simulation of Edge and Mist Computing environments is not such a simple task. Each IoT application (smart cities, connected vehicles, etc.) uses a heterogeneous mix of sensing and actuation devices. These devices, connected to telecommunication networks, can interact with one another or with computing infrastructures in order to compute their tasks. Simulating such environments will, therefore, require modeling the network, computation resources, the heterogeneity of devices, their behaviors, and the data they generate. Fig. 1 presents the aspects of modeling of Edge Computing environments. To model the virtualized resources (e.g., CPU, memory, storage), many existing solutions have extended and exploited Cloud Computing simulators such as *CloudSim* [3], which is a rich and highly extensible framework that enables the simulation of Cloud resources (virtual machines, hosts, data centers) and services. However, since transmission delays are directly proportional to the network workload, the use of fixed transmission delays as in these existing simulators is not practical, especially when evaluating the scalability of the system. On the other hand, the use of network simulators, such as *OMNET++* [20] and *NS-3* [7], allows efficient network modeling. However, users have to define all the other aspects of the simulation (Fig. 1) such as load generation, tasks orchestration, mobility model, and resources utilization models in order to assess the performance of their solutions, which takes a lot of time and effort.

Motivated by this, in this paper, we present *PureEdgeSim*, a simulation framework that enables the evaluation of resources management strategies and the performance evaluation of Cloud, Edge, and Mist Computing environments [11]. It covers all the modeling and simulation aspects of Edge Computing that are given in Fig. 1. *PureEdgeSim* offers a modular architecture where each of its modules deals with a specific part of the simulation. The *Network Module*, for example, is responsible for data transfer and bandwidth allocation. The *Location Manager* module deals with the geo-distribution of devices and their mobility. The *Data Centers Manager* module takes care of the generation of devices and their heterogeneity. Finally, the *Orchestrator* module, which is responsible for tasks offloading decisions. These modules also provide a default implementation and a set of adjustable parameters in order to ease experimentation and prototyping. As a result, researchers can quickly implement their solutions without wasting time on the specification of low-level details. To demonstrate its capabilities, a case study is introduced, in which we propose a simulation scenario that mimics a smart university campus. During this case study, we propose a multi-tier architecture that takes advantage of smart edge devices that have sufficient computing capacity. To support their heterogeneity and meet the QoS, we present a tasks orchestration algorithm that is based on the Fuzzy Decision Tree. The simulation results show the effectiveness of *PureEdgeSim* in modeling such complex, heterogeneous, and dynamic environments. They also highlight the advantages of adopting Mist Computing and the effectiveness of the proposed algorithm that outperformed the competitor algorithms in every aspect of the comparison.

This paper is organized as follows: In section 2, the related work is presented. Section 3 describes *PureEdgeSim* architecture. In section 4, a use case scenario is proposed. The simulation results are assessed in section 5. Finally, section 6 concludes the paper and highlights future directions.

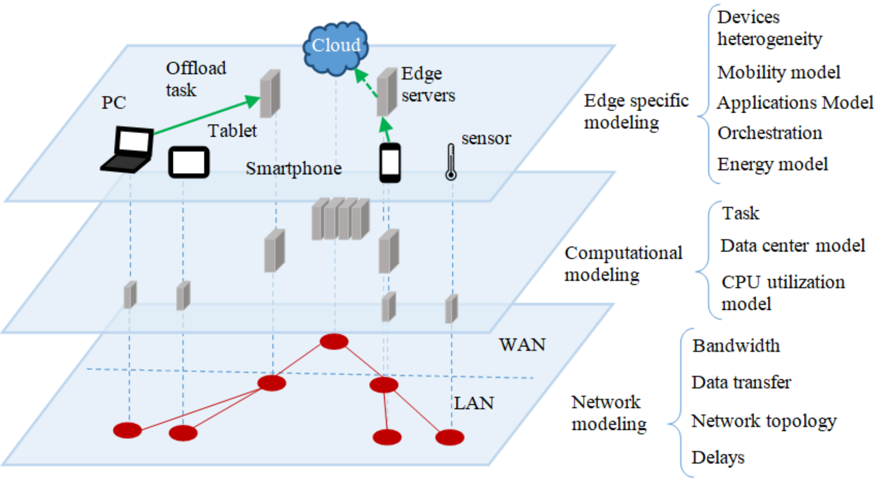


Fig. 1. The aspects of modeling Edge and Mist Computing environments

2. Related Work

In traditional Cloud Computing, devices at the edge of the network offload their tasks to the Cloud for processing them. This task offloading may be necessary for several reasons: some devices offload their tasks because of their low computing capabilities, devices with capacity-limited batteries must offload their tasks to extend their life, and so on.

Edge and Mist Computing use the same offloading process. Tasks offloading allows edge nodes to work cooperatively in order to increase system throughput [18]. In [12], the authors proposed a mechanism that offloads tasks between mobile devices to balance their power consumption. This mechanism has extended network life by 400%. A related system has been developed in [25] called Serendipity. It allows mobile devices to remotely access the resources of other devices to run their applications, which resulted in minimizing the local power consumption while reducing the overall tasks completion time by 6.6 times. A. Mukherjee et al. [15] have introduced a framework that takes advantage of smart devices available at the network edge in order to perform data analytics in IoT. To do so, capacity-based partitioning was introduced, where data is partitioned according to the capacities of those devices. Although performance has decreased when using those devices, the workload of the Cloud has also been reduced, which can solve the Cloud scalability challenge. In [22], an energy-sensitive tasks offloading algorithm has been proposed. It allows mobile devices to dynamically choose the Cloud or the Fog to offload their tasks according to their delay tolerance and power consumption. The results show that this approach outperforms Cloud-only and Fog-only strategies. In [24], the authors have proposed a platform that orchestrates tasks between IoT gateways, Fog servers, and the Cloud, depending on the availability of resources. In [14], V. Chamola et al. focused on reducing latency in Mobile Edge Computing, by searching for the best Fog node to execute tasks when the nearest node is overloaded. This algorithm has achieved a very low latency compared to the traditional scheme that only uses the closest Fog node. Always in Mobile Edge Computing, a Fuzzy Logic based orchestration algorithm was introduced in [28]. The results show the effectiveness of adopting Fuzzy Logic. However, fuzzy logic is not applicable to unknown systems that lack information, and setting exact fuzzy rules is a complicated task. Consequently, the results may not always be correct and are perceived on the basis of assumptions.

With the advancement of research in this field, simulation began raising much interest, leading to the development of many simulation frameworks such as *iFogSim*. *iFogSim* is a *CloudSim*-based simulation framework designed to simulate Fog Computing environments [4]. It allows several types of components (sensors, actuators, gateways, etc.) to be added and linked to form a topology. Nevertheless, this topology remains static, making it lacks mobility support, which is one of the main reasons for adopting Fog Computing. Also, simulating large-scale scenarios that involve hundreds (if not thousands) of devices will require adding and linking them one by one, which is inconvenient, involves a lot of effort, and time-consuming. Moreover, a fixed delay is assigned to each link, ignoring the effect of the network load on the transmission delays.

IoTSim [21] is another simulator that is based on *CloudSim*. It simulates batch-oriented IoT applications where data is sent in large amounts to a processing system, using a MapReduce large data processing model. *SimIoT* [9] is a toolkit that simulates the communication between IoT devices and the remote Cloud. It allows the experimentation of multi-user submission dynamically in the IoT environment. Nevertheless, it does not consider the heterogeneity of IoT devices, and their energy consumption is ignored. IoT will count more than 38.6 billion devices by 2025 [1]; all of them generate data continuously, making energy consumption a major concern. *EmuFog* [6] is an emulation framework for Fog Computing environments. It allows the simulation of Docker-based applications. Since *EmuFog* is based on *MaxiNet* [8], the events of each node (including CPU and memory utilization) are saved in a log. However, because it lacks a generic interface, it cannot deal with global metrics, such as response delay.

Finally, *EdgeCloudSim* [13] is another *CloudSim*-based simulator for Mobile Edge Computing that addresses some *iFogSim* limitations. It automatically generates the required number of edge devices, making it more scalable. It supports mobility to a certain extent, and the network model is more realistic. However, its mobility model is over-simplified. It also lacks modeling the power consumption of edge devices, their remaining energy, and their death on simulation runtime (i.e., when they run out of energy). It cannot execute tasks locally on these devices or offload them to other edge devices, limiting its use to Mobile Edge Computing scenarios.

Although Mist Computing has attracted a lot of interest [12, 15, 22, 25], simulation tools are still lacking. To the best of our knowledge, there is no simulator capable of modeling Mist Computing environments (Fig. 2), which involves processing data on edge devices, modeling their heterogeneity, measuring their energy consumption and their resources utilization, etc. Motivated by this, we introduce *PureEdgeSim*, a simulation toolkit that is designed to simulate Cloud, Edge, and Mist Computing environments. Thus, enabling the simulation of a multitude of scenarios such as Mobile Devices Clouds [12, 25], Mobile Edge Computing [13, 14], and multi-level scenarios where different computing paradigms are used simultaneously (such as Foggy [24]).

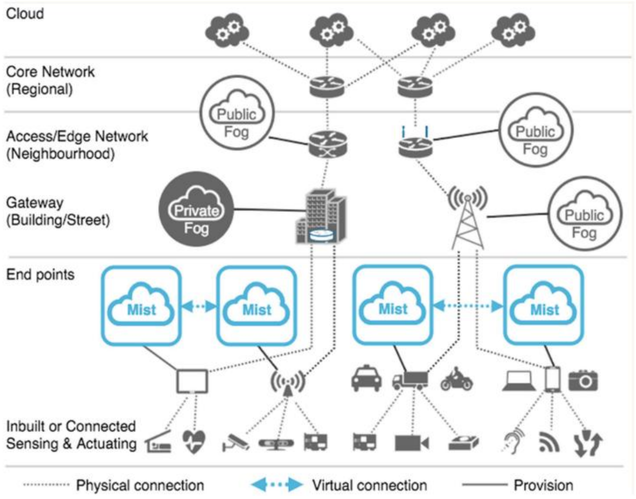


Fig. 2. The role of Mist Computing on the internet of things [26]

3. *PureEdgeSim* Architecture and Design

PureEdgeSim takes advantage of *CloudSim Plus* features [2], including the native support for the discrete events simulation, that is used during the communication between its components. It also leverages its rich and very extensible library that covers all the aspects of Cloud Computing from resources (i.e., data centers, hosts, etc.) to services (i.e., virtual machine allocation policies, CPU schedulers, etc.) enabling it to model computational tasks effectively (the middle layer of Fig. 1). Hence, only a few classes were added to model Edge and Mist Computing environments. Therefore, the

CloudSim Plus layer is responsible for providing key components that are extended by *PureEdgeSim* (Fig. 3), and it is also behind the interactions of its components.

3.1. *PureEdgeSim* Modular Design

The simulation of Edge and Mist Computing environments allows evaluating the adopted resources management strategies before their actual deployment. However, the heterogeneity of possible scenarios complicates the task, especially when using a simulator such as *Omnet++* or *NS-3* where the user has to define all the aspects of the simulation from the specification of resources, networking, energy, mobility, etc. which requires a lot of time and effort. To cope with it, *PureEdgeSim* follows a modular architecture that consists of seven modules, where each one of them deals with a specific Edge Computing modeling issue. To facilitate prototyping and experimentation, each module offers a default implementation with a ready to use set of adjustable parameters. These modules are:

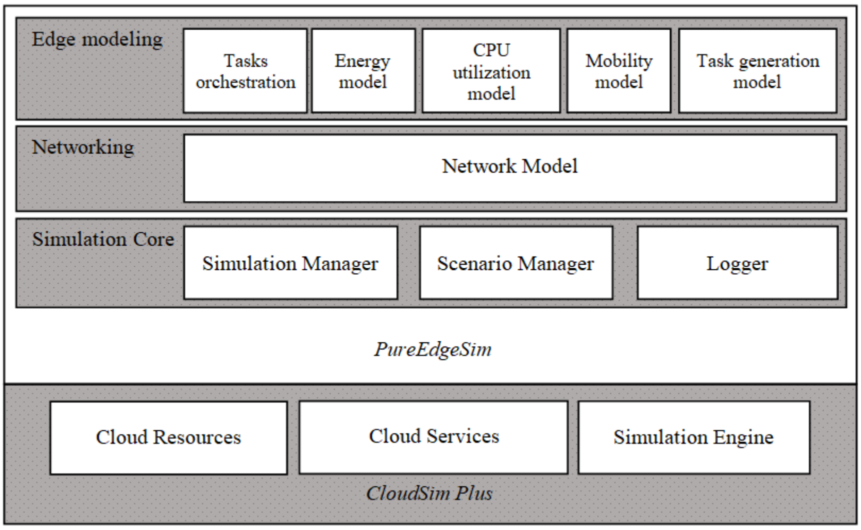


Fig. 3. *PureEdgeSim* layered architecture

Simulation Manager. This module is responsible for initiating and managing the simulation environment, scheduling events, and generating the output files. It contains three essential classes, the *Simulation Manager* class, which initializes the simulation environment, starts the simulation, and schedules its end. The second class is the *Simulation Logger*. It is responsible for generating the simulation output; it calculates the results, shows them at the end of every iteration, and saves them to a comma-separated value (CSV) format to easily exploit them later. Finally, the *Real-Time Display* class that displays real-time information such as the simulation map and other charts (network utilization, CPU utilization, and tasks failure rate). This helps to understand the course of the simulation better, and above all, to analyze the proposed solution in real-time, especially the mobility model.

Data Centers Manager Module. It consists of three classes (Fig. 4): (i) The *Edge Data Center* class that extends the *Data Center Simple* class of *CloudSim Plus* to model the heterogeneity of edge devices, (ii) the *Server Manager* that generates the required data centers and edge devices, their hosts, and their virtual machines according to the configuration files, and finally, (iii) the *Energy Model* class which is responsible for updating their energy consumption.

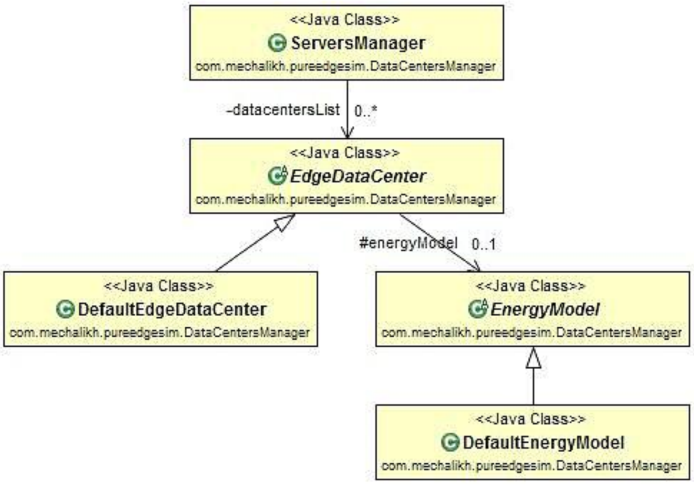

<<Java Class>>
ServersManager
com.mechalikh.pureedgesim.DataCentersManager
-datacentersList
0..*
<<Java Class>>
EdgeDataCenter
com.mechalikh.pureedgesim.DataCentersManager
#energyModel
0..1
<<Java Class>>
DefaultEdgeDataCenter
com.mechalikh.pureedgesim.DataCentersManager
<<Java Class>>
EnergyModel
com.mechalikh.pureedgesim.DataCentersManager
<<Java Class>>
DefaultEnergyModel
com.mechalikh.pureedgesim.DataCentersManager

Fig. 4. The Data Centers Manager classes

Tasks Generator. *PureEdgeSim* supports the generally used applications models: the sense-process-actuate and the stream-processing models. In the first one, the data collected by sensors is sent to computing nodes for processing. The results of the processing are then sent back to the actuator to take the necessary actions. The second model involves a network of application modules that continuously process the data streams generated by sensors. The extracted information is stored for large-scale and long-term analysis [4].

In *PureEdgeSim*, the data emitted from sensors and edge devices are modeled as tasks. By default, the tasks generator assigns an application such as e-health, infotainment, and augmented reality (which can be defined in the applications XML file) to edge devices, where each application has its specific characteristics (i.e., data size, CPU utilization, latency-sensitivity, etc.). After that, it will generate the tasks of every device according to the assigned application type. This module consists of two main classes (Fig. 5): The first one is the *Task* class, which is inherited from the *Cloudlet Simple* class of *CloudSim Plus*, and the second is the *Task Generator*. The latter generates the tasks that will be offloaded during the simulation.

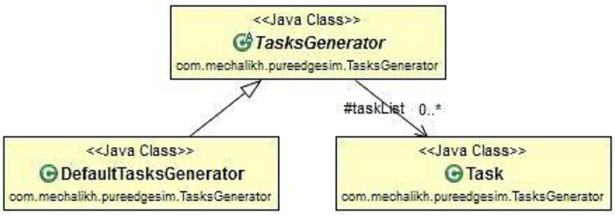

<<Java Class>>
TasksGenerator
com.mechalikh.pureedgesim.TasksGenerator
#taskList 0..*
<<Java Class>>
DefaultTasksGenerator
com.mechalikh.pureedgesim.TasksGenerator
<<Java Class>>
Task
com.mechalikh.pureedgesim.TasksGenerator

Fig. 5. The Tasks Generator classes

Location Manager. Geographical distribution and mobility are the main attributes of Edge and Mist Computing. Thanks to the geographical distribution of their computing nodes [5], they can continue serving mobile devices during their mobility while providing the lowest latency. This has spawned a new generation of latency-sensitive applications such as connected vehicles. To support such scenarios, this module assigns an initial location to each device, and realistically manages their mobility. It contains two main classes (Fig. 6): The *Location* class, which represents the X and Y coordinate of the device, and the *Mobility* class that generates the next location for each mobile device.

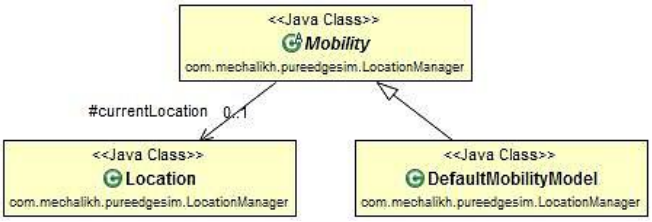

<<Java Class>>
Mobility
com.mechalikh.pureedgesim.LocationManager
#currentLocation 0..1
<<Java Class>>
Location
com.mechalikh.pureedgesim.LocationManager
<<Java Class>>
DefaultMobilityModel
com.mechalikh.pureedgesim.LocationManager

Fig. 6. The Mobility Manager classes

The Task Orchestration Module. The generation, capture, and analysis of data will be in volumes, variety, and orders of magnitude larger than before. Effective implementation of the infrastructure requires several key decisions: mainly how the data will be collected and how it will be processed. These decisions are influenced by two competing pressures: the use of the infrastructure and the Quality of Service required by the end-user [19]. Hence, the simulators must allow the implementation of custom resources management techniques in order to enable their wider applicability [17].

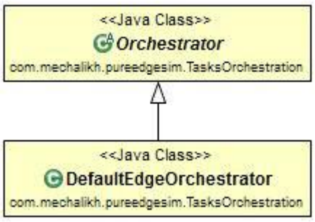

<<Java Class>>
Orchestrator
com.mechalikh.pureedgesim.TasksOrchestration
<<Java Class>>
DefaultEdgeOrchestrator
com.mechalikh.pureedgesim.TasksOrchestration

Fig. 7. The Tasks Orchestrator classes

PureEdgeSim allows this through the *Tasks Orchestration Module.* It consists mainly of the *Orchestrator* (Fig. 7), which represents the decision-maker. Depending on the used policy, it decides whether to offload the task or execute it locally and where to offload it. Users can quickly implement their orchestration policies (i.e., the tasks orchestration algorithm) by extending the *Orchestrator* class.

The Network Module. This module addresses the networking layer presented in Fig. 1. It primarily consists of the *Network Model* (Fig. 8). Unlike in *CloudSim Plus* (same for *CloudSim*), where the bandwidth allocated to each virtual machine remains static, this network model takes into account the network load at each instant of the simulation. When transferring data, at each instant of the transfer (i.e., from the beginning until the end), its allocated bandwidth will vary based on the network load at that moment. This network model also takes into consideration the bandwidth limit caused by WAN or WLAN congestion. As a result, if multiple devices connect to the same WLAN access point, the bandwidth allocated to each device decreases. If this allocated bandwidth is below that of the WAN, the transfer speed of the data sent to (or received from) the Cloud will be limited by the low bandwidth of the WLAN and not by that of the WAN.

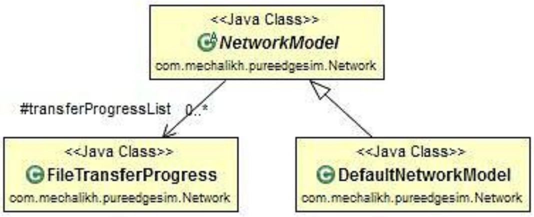


Fig. 8. The Network Model classes

The Scenario Manager Module. Each use case requires a heterogeneous combination of devices. The heterogeneity involves the devices mobility, energy source, computing capacity, the heterogeneity of applications, and their requirements (e.g., latency). Therefore, the simulation framework must be able to support the diversity of devices and their different Quality of Service requirements [17, 19]. Besides, Edge Computing simulators need to be easily extended with new types of devices and applications without modifying their internals [17]. The *Scenario Manager* module guarantees this by loading the simulation parameters and the user scenario from the input files. It contains two principal classes, the *Simulation Parameters* class, which acts as a placeholder for the different parameters, and the *Files Parser* that loads the user scenario and settings from specific configuration files representing the input method.

3.2. The Simulation Duration and Realism

As mentioned previously, *PureEdgeSim* relies on *CloudSim Plus*, one of the commonly used Cloud Computing simulators that provides a reliable code base for modeling computational tasks. To fulfill the remaining simulation requirements (Fig. 1), *PureEdgeSim* offers the most realistic network, energy, and mobility models as compared to existing solutions. However, since it is a discrete event simulator, the simulation time complexity will depend on the number of events generated at runtime, as seen in Fig. 9, where there is a clear correlation between the number of events and the duration of the simulation. To reduce simulation time, *PureEdgeSim* offers a quick and full control of the simulation environment through its set of parameters, where users can trade-off between simulation realism and duration. Hence, the realism will depend on the user settings, especially the update intervals (Table 3). The shorter these intervals, the more accurate and realistic the simulation will be, but also, the longer it will take (Fig. 9).

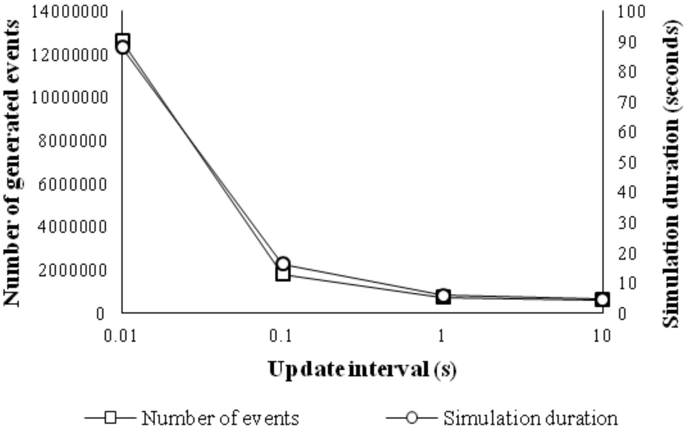


Fig. 9. The impact of update interval on the number of generated events and the simulation duration

3.3. Ease of Use and Extensibility

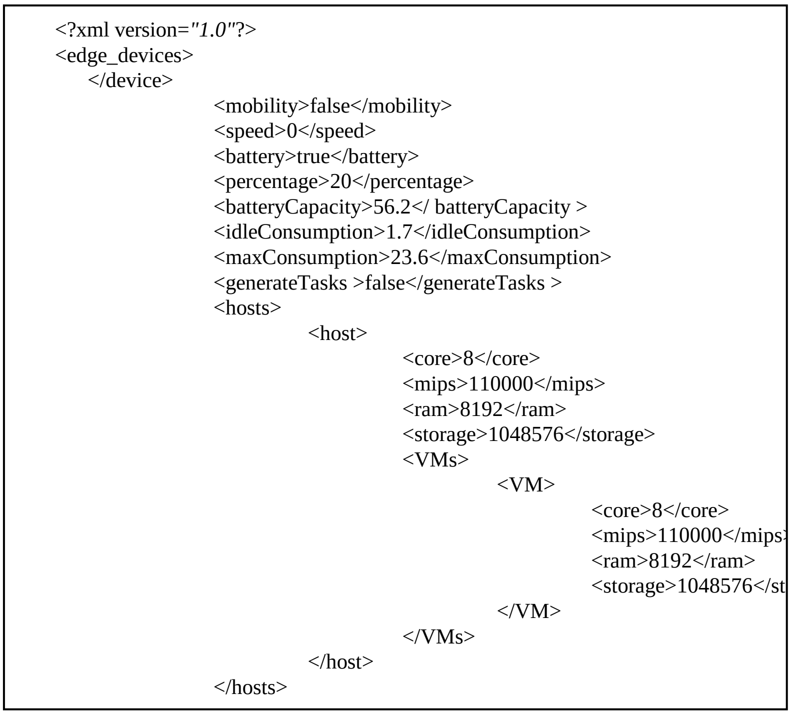
Simulators can be used to compare the performance of different configurations in order to determine the factors that affect performance the most, e.g., the network settings, the number of entities used in the simulation, the amount of resources, etc. Treating all these variables programmatically is a challenge. To reduce time and effort, each module provides a default implementation (e.g., the *Default Edge Data Center* class in the *Data Centers Manager* (Fig. 4), the *Default Mobility Model* (Fig. 6), etc.). These ready-to-use models also offer a fully customizable environment through a multitude of parameters and configuration files, allowing users to customize the components behavior without changing the original code.

Extensibility is another essential feature of *PureEdgeSim*. Even though *PureEdgeSim* uses its pre-built models by default, users can always create and integrate their custom models when building their simulation scenarios if any of these default implementations do not meet their needs, without having to modify the *PureEdgeSim* code base.

As a result, the simulation scenario can be quickly built by following these simple steps:

a) The implementation of custom models, if needed.
b) The definition of Cloud and Edge resources and the application characteristics: This can be done by editing the following configuration files that are located under the *settings/* folder:

- The Cloud data centers file: In this file, the user defines the Cloud data centers, their power consumption rate, describes their hosts (CPU, storage, ram), and the virtual machines of each of these hosts.
- The Edge data centers (i.e., Cloudlets, servers) file: Like the Cloud data centers file, this file defines the edge data centers, their specifications, their locations, and their energy consumption rates.
- The edge devices file: Instead of defining the devices one by one (which takes a considerable amount of time and effort due to the large number of devices), in *PureEdgeSim*, the user will only define the types of devices and the percentage of each one of them. Then, according to this percentage and the total number of devices (which is set in the simulation parameters file), the *Data Centers Manager* will generate the devices of each type. To support the heterogeneity of devices, *PureEdgeSim* offers endless possibilities, varying from simple sensors (i.e., without any computing capacity) to smartphones, laptops, as well as sophisticated servers that run hundreds of virtual machines. It can be done by specifying the mobility of the device (i.e., whether the devices of this type are mobile), the power source (i.e., if they are battery-powered), the capacity of the battery in watt-hour if it is battery-powered, the power consumption rates in watt-Hour, and the computing capacity in MIPS (Million Instructions Per Second). Fig. 10 gives an example of a device type (see the laptop type in Table 1).
- The applications file: This file defines the set of IoT applications that are needed by the *Tasks Generator* (see Table 2). Each application is defined by: the usage percentage that represents the proportion of devices running that application, the generation rate which is the number of tasks generated per minute, its delay tolerance in seconds, the task length which refers to the number of its instructions in MI (Million Instructions) and determines its execution time, the size in Kbytes of the offloading request, the container image, and the returned results.



```
<?xml version="1.0"?>
<edge_devices>
    </device>
                <mobility>false</mobility>
                <speed>0</speed>
                <battery>true</battery>
                <percentage>20</percentage>
                <batteryCapacity>56.2</ batteryCapacity >
                <idleConsumption>1.7</idleConsumption>
                <maxConsumption>23.6</maxConsumption>
                <generateTasks >false</generateTasks >
                <hosts>
                        <host>
                                <core>8</core>
                                <mips>110000</mips>
                                <ram>8192</ram>
                                <storage>1048576</storage>
                                <VMs>
                                        <VM>
                                                <core>8</core>
                                                <mips>110000</mips
                                                <ram>8192</ram>
                                                <storage>1048576</st
                                        </VM>
                                </VMs>
                        </host>
                </hosts>
```

Fig. 10. The edge devices XML file

c) Setting the simulation parameters: To ease the implementation of simulation scenarios, each of the previous modules provides a set of adjustable parameters (Table 3), that can be found in the *simulation_parameters.properties* file.

To demonstrate the capabilities of *PureEdgeSim*, especially when it comes to the simulation of Edge Computing environments and its support for the heterogeneity of both edge devices and scenarios, in the next section, we propose a case study.

4. Application and Case Study

As proof of concept, in this section, we propose a simulation scenario. We will focus on aspects that can only be modeled by *PureEdgeSim,* such as the support for Mist Computing scenarios, network utilization, mobility of devices, and their energy consumption. First, we will introduce a tasks orchestration platform. We also propose a tasks orchestration algorithm that is based on Fuzzy Decision Tree. Finally, a simulation scenario is given by which we evaluate their performance.

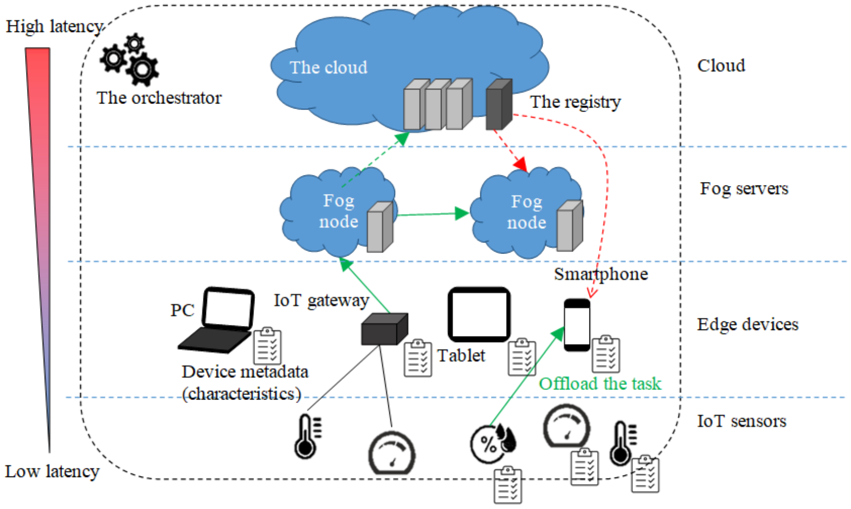


Fig. 11. The proposed architecture

4.1. The Tasks Orchestration Platform

We introduce a multi-tier architecture (Fig. 11). It consists of: The IoT sensors layer, that represents the source of data, the smart edge devices layer, the Fog layer, and the Cloud one. By relying on close devices equipped with enough computing capacity (PCs, smartphones, tablets, smart TVs, etc.), it enables QoS compliance and better scalability. While these devices may offer low computing capabilities compared to the Fog or the Cloud, their potential lies in their ever-growing number, which is expected to exceed 10 billion by 2025 (excluding IoT sensors) [1]. Additionally, they can deliver the lowest latency and support mobility to some extent, given their massive geographical distribution and location (i.e., a hope away from each other). Hence, latency-sensitive applications can be placed in these devices, while computationally intensive ones can be placed on the Cloud or Fog servers based on their delay tolerance.

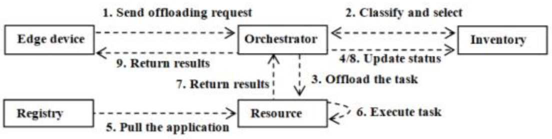


Fig. 12. The task offloading flow and the role of the orchestrator

It consists of the following entities (Fig. 12):

a) IoT sensors, which are resource-limited devices. They must offload their tasks elsewhere for processing. For this purpose, a task offloading request will be sent to the orchestrator. The request provides information about the status of the device and its requirements, e. g. application ID (i.e., container ID), data to be processed, task latency sensitivity, that are necessary to find the best offloading destination.

b) The application: A piece of software hosted on a container (e.g., Node.js script).
c) The registry: The repository from where the required application will be pulled.
d) The resource: The potential destination of the task, where it will be executed.
e) The orchestrator (i.e., the decision-maker): decides where the task will be executed, using a specific orchestration algorithm. This algorithm will also be hosted on a container, to facilitate its download, execution, and update.
f) The inventory (i.e., the list of available resources): When a device joins the network, it communicates its meta-data (Fig. 11), including its resources and its remaining energy, to the orchestrator. The orchestrator will add this device to its inventory. Then, when receiving a task offloading request, it will classify the resources (its inventory) to find the one that best suits this task.

To minimize energy consumption and delays, a decentralized orchestration strategy will be used, in which, the orchestrators will be selected using a cluster head selection.

4.2. The Orchestration Algorithm

The proposed architecture can guarantee a high quality of service. However, managing these heterogeneous resources is not an easy task. Several factors, such as the remaining power, resource utilization, and network condition, should be taken into account. These dynamic factors can vary unexpectedly. Traditional multi-constraint optimization cannot be applied due to insufficient information about the nature of the tasks and arrival times, which necessitates an online solution that can adapt to this ever-changing environment. To guarantee this, we present a tasks orchestration algorithm that is based on the fuzzy decision tree. Fuzzy decision trees combine the advantages of fuzzy logic and decision tree, among which are: it can handle uncertainties without requiring a complex mathematical model and support multi-criteria decision processes, its computing complexity is low, which is essential for an online decision algorithm, and it requires less preparation effort. This algorithm is based on Yuan et al. [29] fuzzy decision tree approach, and consists of two stages (Algorithm 1):

a) The first stage: During this stage, the tasks are classified into Cloud, Fog, or Mist tasks based on the following criteria: (i) the task latency-sensitivity: the reason for adopting Edge and Mist computing; if a task is tolerant to delay it will be sent to the Cloud, otherwise to the Fog or edge devices depending on Fog utilization and the device mobility. (ii) Fog resources utilization: Fog servers are not supposed to be as powerful as the Cloud; for this reason, they may be overloaded. This may lead to high delays, causing the failure of many tasks. In this case, the Cloud can be a good alternative if the WAN bandwidth is high enough. (iii) Device mobility: if the device offloading the task is mobile, edge devices should be avoided. (iv) WAN bandwidth: if it is below a certain threshold, the Cloud should be avoided.
b) The second stage: If the Cloud or the Fog has been chosen in the first stage, the task will be directly offloaded. However, if the choice has been made on edge devices, the algorithm will classify them during the second stage to find the most

suitable one. This classification will be based on the following criteria: (i) The utilization of device resources. (ii) Energy source: battery-powered devices should be skipped when possible. (iii) The mobility of both devices: The failure of a task due to mobility happens when a device (the one offloading the task or the one executing it) relocate before finishing it. Thus, offloading the task to a mobile device may cause its failure. This risk of failure becomes even higher when both devices are mobile.

Algorithm 1. The proposed algorithm

```
procedure findDestination(request, inventory)
type ← fuzzyDecisionTree1.classify (request)
if (type = ‘Cloud’)
   offload(offloadingRequest, Cloud)
else if (type = ‘Fog)
   offload(offloadingRequest, Fog)
else
   maxTruthLevel ← −1
   destination ← null
   finestClass ← low
   for each resource ∈ inventory do
   (class, truthLevel) ← fuzzyDecisionTree2.classify(resource)
   if (class > finestClass) //get the optimal destination
   finestClass ← class
   maxTruthLevel ← truthLevel
   selectedDevice ← resource
   else if (class = finestClass)
   newTruthLevel ← truthLevel
   if (maxTruthLevel = −1 or newTruthLevel > maxTruthLevel)
   maxTruthLevel ← newTruthLevel
   selectedDevice ← resource
   end_if
   end_if
   end_for
   offload(request, selectedDevice)
   end_if
end
```

4.3. The Simulation Scenario

To demonstrate the capabilities of *PureEdgeSim*, we introduce a simulation scenario that imitates a smart university campus. A smart campus consists of integrating information and communication technologies by deploying sensors in several locations to get useful information that will be exploited to manage and optimize resources, increase energy efficiency, and improve education [23]. In this scenario, students who own mobile devices (e.g., smartphones) relocate after a random amount of time. To do this, the *Default Mobility Model* will be used. The area also involves other devices (Table 1): simple sensors (e.g., wearables) and non-mobile devices (e.g., laptops and gateways).

Table 1. The types of Edge devices

Edge devices types	Laptop	Smartphone	IoT Gateway	Sensor
Mobility	No	Yes	No	No
Speed (meters per second)	0	1.4	0	0
Battery-powered	Yes	Yes	No	No
Generate tasks	No	Yes	No	Yes
Percentage of devices (%)	20	30	10	40
Battery-capacity (Wh)	56.2	18.75	-	-
Idle energy consumption rate (Wh)	1.7	0.078	1.6	0.036
Max energy consumption rate (Wh)	23.6	3.3	5.1	-
CPU (GIPS)	70	25	16	-
CPU Cores	8	8	4	-
Ram (Gbyte)	8	4	2	-
Storage (Gbyte)	1024	128	32	-

The last type refers to simple sensors that do not have sufficient computing capacity (only generate data/tasks). The energy consumption rates were measured from the following devices under different workloads: a laptop running Windows 10 (Intel® processor Core™ i7-8550U), a smartphone running Android 10 (HiSilicon Kirin 710), and a Raspberry Pi 3 Model B+ running Raspbian. On the other hand, the CPU values are obtained by running a Dhrystone benchmark [27].

To extend their lives, battery-powered devices should, if possible, offload their tasks, while devices with computationally intensive tasks should offload them to minimize execution time. The *Proposed* multi-tier architecture will be evaluated against:

1. The *Cloud-Only* architecture where all the tasks are offloaded to the Cloud.
2. The widely adopted *Fog-and-Cloud* architecture: in this case, the tasks can be offloaded either to the Cloud or Fog servers.

The Cloud will be represented by one data center with a total of 4000 GIPS, distributed over 16 virtual machines, while the Fog will have a similar data center but with lower computing capacity (3200 GIPS), which should be enough for this small simulation area.

To model the different possibilities, four types of applications with different characteristics are used (Table 2): (i) A health application: The data generated by wearables will not exceed a few kilobytes and will not use significant processing power [28]. (ii) An augmented reality application [30]: The data sent is an image, usually about 1 Mbyte in size. (iii) Other heavy computing tasks requiring additional computing power, e.g., machine learning, may also be included [30]. (iv) An infotainment application [31].

Table 2. The types of applications

Applications types	**Health**	**Augmented reality**	**Computation-intensive**	**Infotainment**
Usage percentage (%)	20	30	20	30
Generation rate (tasks per minute)	20	20	2	4
Latency sensitivity	Yes	Yes	No	No
Task length in Giga Instructions (GI)	1500	5	50	10
Request size (Kbytes)	20	1500	3000	50
Results size (Kbytes)	20	50	200	50

When offloading the task, the orchestrator will choose the offloading destination using one of the following orchestration algorithms:

1. *Proposed*: The proposed Fuzzy Decision Tree based algorithm.
2. *ECOOA:* The energy-oriented tasks orchestration algorithm [22], which is the closest in terms of criteria.
3. *Fuzzy Logic*: The Fuzzy Logic based tasks orchestration algorithm [28].

The simulation parameters for this scenario are resumed in Table 3.

Table 3. The simulation parameters

Parameter	Value
Simulation duration	30 (min)
Update interval	0.01 (s)
Min number of Edge devices	100
Max number of Edge devices	500
Edge devices counter step size	100
Edge devices range	10 (meters)
Simulation area size	200 x 200 (meters)
Network update interval	0.1 (s)
WLAN bandwidth	300 (Mbits/s)
WAN bandwidth	20 (Mbits/s)
WAN propagation delay	0.2 (s)
Orchestrators deployment	Decentralized.
Orchestration algorithm	Proposed, ECOOA, Fuzzy Logic.
Architectures	Cloud-Only, Fog-and-Cloud, Proposed.

5. Simulation Results and Discussion

To demonstrate the effectiveness of *PureEdgeSim* in modeling Cloud, Edge, and Mist computing environments, in this section, the proposed platform will be evaluated against existing solutions. Although *PureEdgeSim* can generate charts automatically, the figures presented in this section have been created from the output CSV file using Microsoft Excel. This file offers more than 60 different ready-to-use metrics, from which the user can also derive others as well. For instance, the network delay in Fig. 13 is obtained by dividing the total network utilization by the number of sent tasks. Similarly, the service time is the sum of network delay and execution time. During this evaluation, we will focus on meaningful metrics that determine the Quality of Service and reflects the scalability of the proposed solution, such as the tasks failure rate, energy consumption, delays, network usage, and CPU utilization.

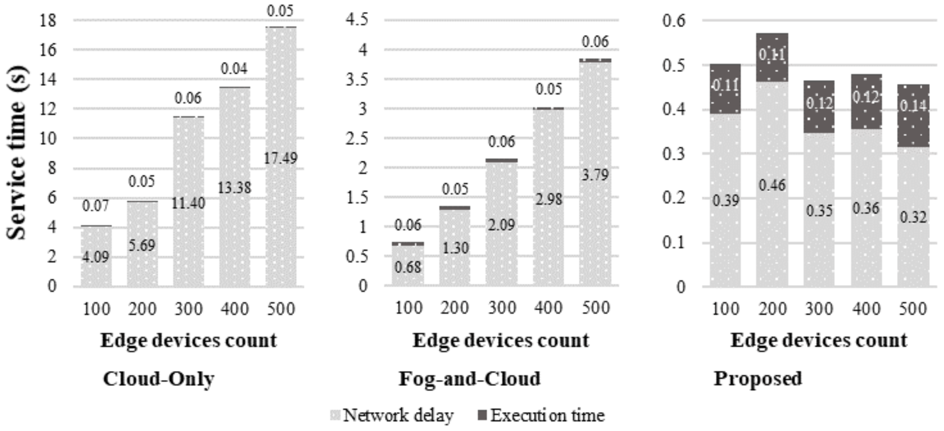
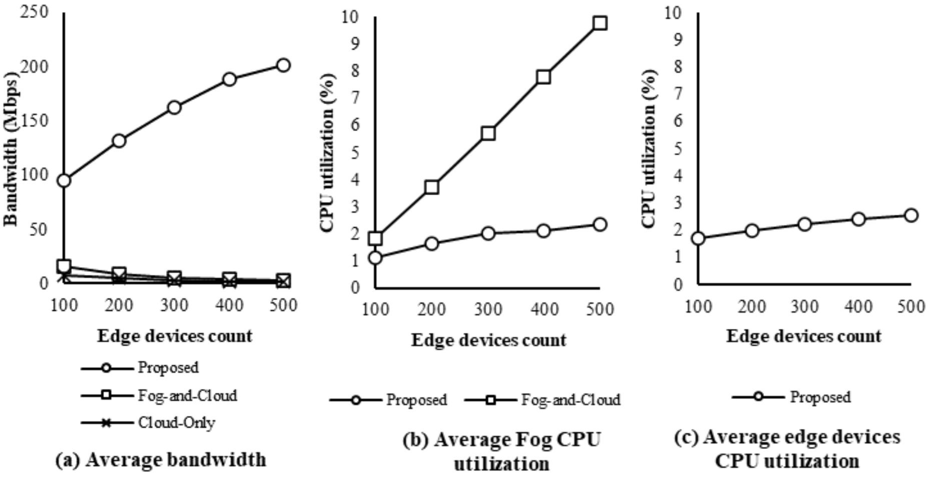


Fig. 13. Average tasks service time



(a) Average bandwidth

(b) Average Fog CPU utilization

(c) Average edge devices CPU utilization

Fig. 14. A comparison between the different architecture

5.1. Evaluating the Architectures

The average service time, which consists of execution time and network delay, is given in Fig. 13. The latter is considered an essential factor that has a direct effect on the Quality of Service. When using the *Cloud-Only* architecture in which all the tasks are offloaded to the remote Cloud, the service time has been very long, although the Cloud provides the highest computing capacity, due to the high use of the backhaul network. By using the Fog along with the Cloud, the delay has been reduced remarkably. However, it increases as the number of devices grows. On the other hand, the tasks completion delay stayed almost stable when using the proposed architecture; despite their low computing capacity, the use of edge devices has managed to decrease the average delay regardless of the number of devices to an average of 0.5 seconds.

Fig. 14 shows the average CPU usage of the Fog and edge devices, as well as the allocated bandwidths using the different architectures. Since all tasks are transferred to the Cloud when using the *Cloud-Only* architecture, the backhaul network becomes overloaded, resulting in the lowest allocated bandwidth, as shown in Fig. 14 (a). This low bandwidth justifies the high service time depicted in Fig. 13. The use of Fog servers has managed to double the average allocated bandwidth for each task. However, the Fog CPU utilization rose dramatically when the number of devices grows (Fig. 14(b)). This increase will force it to offload their tasks surplus to the Cloud, thus, raising the network utilization again, as seen in Fig. 14(a).

Thanks to its horizontal scalability, Mist Computing benefits from the rapid growth of devices. The growth of devices, in this case, means the availability of more resources. Thus, the workload remains stable regardless of the number of devices, which is confirmed by Fig. 14(c), where the average CPU usage of edge devices has remained stable at around 2% when the proposed multi-tier architecture was used. As edge resources grow, there is less need to offload tasks to the Fog or the remote Cloud, resulting in low Fog CPU usage, and an increase in allocated bandwidth.

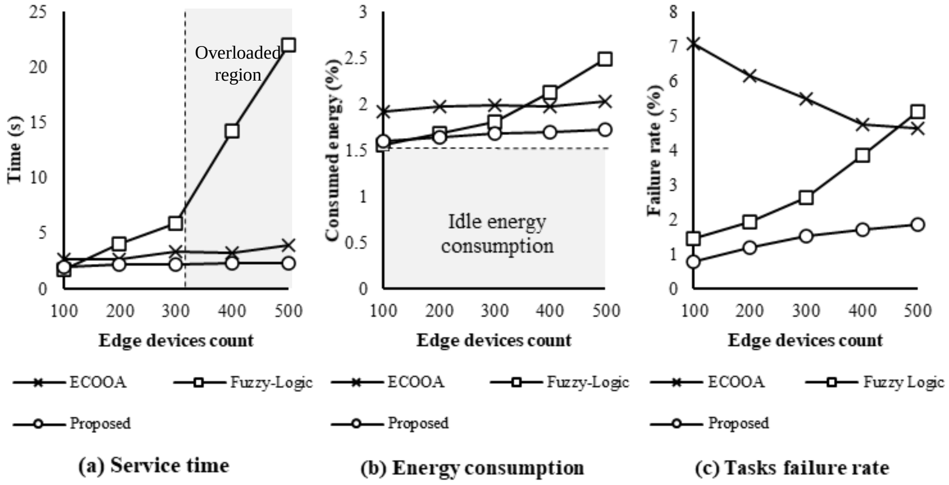


Fig. 15. A comparison between the different algorithms in terms of service time, energy consumption, and failure rate

5.2. Evaluating the Orchestration Algorithms

Fig. 15 compares the different algorithms in terms of service time, the power consumption of battery-powered devices, and tasks failure rate, which are the main performance criteria. The *Fuzzy-Logic* algorithm offloads delay-tolerant tasks to the Cloud when network bandwidth is sufficient, while the other ones are offloaded to the Fog or edge devices as long as their utilization is low. When the number of devices grows (the overloaded area that is highlighted in gray in Fig. 15(a)), the Cloud workload increases. To compensate, this algorithm will offload the excess to the Fog and edge devices, increasing their utilization and consuming their power. When their CPU utilization exceeds a certain threshold, the algorithm will switch to the Cloud regardless of bandwidth, resulting in long service time. This high delay has caused the failure of many tasks, which justifies the correlation between all these three charts. The *ECOOA* has performed better since it aims to minimize delays and energy consumption. However, because it does not distinguish between tasks, a significant portion of latency-sensitive tasks were offloaded to the Cloud. As a result, higher tasks failure rate.

Besides avoiding mobile devices, the proposed algorithm avoids battery-powered devices as much as possible. As a result, the energy consumption of those devices has been reduced by 79.9% as compared to competitor algorithms (if we exclude the idle energy consumption highlighted in gray in Fig. 15(b)). Because it avoids those devices, a considerable number of tasks are offloaded to the Fog (as long as its utilization is low), reducing the service time by 50.8%, and the failure rate by 60% compared to the closest competitor algorithm.

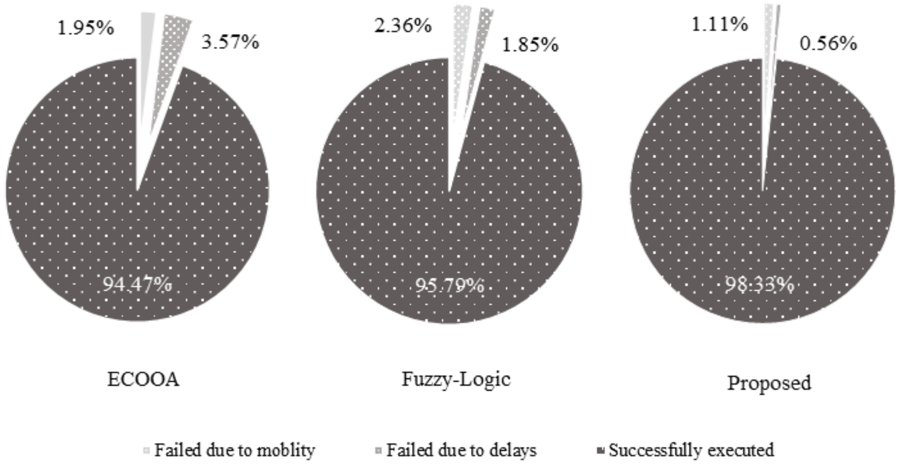


Fig. 16. The tasks failure reasons (400 devices)

The rates of the tasks that are failed due to mobility and high delays are depicted in Fig. 16. As opposed to competitor algorithms, the proposed one considers the heterogeneity of devices and the requirements of their applications. As a result, if a task is not latency-sensitive, it will be offloaded to the Cloud if available. Otherwise, it will be offloaded to the Fog or another edge device. However, because mobile devices are avoided as much as possible, more tasks have been offloaded to the Fog. Hence, reducing failure due to mobility by 43.2%, and since it has decreased the service time, as seen in Fig. 15 (a), the failure due to high delays has also been reduced by 69.8% compared to competitor algorithms.

6. Conclusion and Future Work

Edge and Mist Computing are two emerging computing paradigms that bring Cloud applications close to IoT devices. As a result, decreasing the latency and leading to a more scalable network. In such complex systems, the simulation makes it possible to evaluate the adopted strategy and to analyze its performance before its deployment. Motivated by this, in this paper, we introduced *PureEdgeSim*, a simulation toolkit designed to simulate the Cloud, Edge, and Mist Computing environments and to evaluate their performances. To demonstrate its effectiveness, a case study was introduced. We focused on the aspects that can only be simulated using *PureEdgeSim*, such as the support for the heterogeneity of devices, support for mobility, the realistic network model, etc. which reflects the effectiveness of *PureEdgeSim* and its extensibility.

To conclude, Fog servers will only delay the scalability problem rather than resolving it permanently. They suffer from that same issue as the conventional Cloud, which requires continuously scaling them up to accommodate the rapid growth of connected devices. On the other hand, Mist Computing represents a cost-effective alternative that makes the growth in the number of connected devices in its favor, thanks to its horizontal scalability. With the appropriate pricing model, the latter could easily help to overcome the Cloud limitations. However, due to the heterogeneity of IoT devices, conventional optimization techniques are not sufficient. Other aspects, such as the mobility of the devices, their residual energy, and the latency sensitivity of its application, must be taken into account. By considering them, the proposed orchestration algorithm, which is based on the Fuzzy Decision Tree, outperformed the state-of-the-art solutions in all aspects of the comparison. It has reduced the tasks failure rate by 60%, the energy consumption by 79.9%, and service time by 50.8%, thanks to its set of criteria.

As future work, we are planning to add a pricing model to this simulator and the support for virtual machines migration as well. By introducing this simulator, we hope that this modest work encourages the adoption of Mist Computing in the Internet of Things and enables the development of novel resource management strategies.

Software Availability. The *PureEdgeSim* simulator and the examples are available for download at: https://github.com/CharafeddineMechalikh/PureEdgeSim

References

1. Strategy Analytics (2019). Strategy Analytics: Internet of Things Now Numbers 22 Billion Devices But Where Is The Revenue?.
2. Silva Filho, M. C., Oliveira, R. L., Monteiro, C. C., Inácio, P. R., & Freire, M. M. (2017). CloudSim plus: a Cloud Computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on (pp. 400-406). IEEE.
3. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of Cloud Computing environments and evaluation of resource provisioning algorithms. Software: Practice and experience, 41(1), 23-50.
4. Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K., & Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog Computing environments. Software: Practice and Experience, 47(9), 1275-1296.
5. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog Computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile Cloud Computing (pp. 13-16). ACM.
6. Mayer, R., Graser, L., Gupta, H., Saurez, E., & Ramachandran, U. (2017). Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures. In 2017 IEEE Fog World Congress (FWC) (pp. 1-6). IEEE.
7. Riley, G. F., & Henderson, T. R. (2010). The ns-3 network simulator. In Modeling and tools for network simulation (pp. 15-34). Springer, Berlin, Heidelberg.
8. Wette, P., Dräxler, M., Schwabe, A., Wallaschek, F., Zahraee, M. H., & Karl, H. (2014). Maxinet: Distributed emulation of software-defined networks. In 2014 IFIP Networking Conference (pp. 1-9). IEEE.

9. Sotiriadis, S., Bessis, N., Asimakopoulou, E., Mustafee, N., Towards simulating the internet of things. 2014 28th International Conference on Advanced Information Networking and Applications Workshops; 2014; Victoria, Canada.
10. Marín-Tordera, E., Masip-Bruin, X., García-Almiñana, J., Jukan, A., Ren, G. J., & Zhu, J. (2017). Do we all really know what a Fog node is? Current trends towards an open definition. Computer Communications, 109, 117-130.
11. Dogo, E. M., Salami, A. F., Aigbavboa, C. O., & Nkonyana, T. (2019). Taking cloud computing to the extreme edge: A review of mist computing for smart cities and industry 4.0 in Africa. In Edge Computing (pp. 107-132). Springer, Cham.
12. Mtibaa, A., Fahim, A., Harras, K. A., & Ammar, M. H. (2013). Towards resource sharing in mobile device Clouds: Power balancing across mobile devices. In ACM SIGCOMM Computer Communication Review (Vol. 43, No. 4, pp. 51-56). ACM.
13. Sonmez, C., Ozgovde, A., & Ersoy, C. (2018). Edgecloudsim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies, 29(11), e3493.
14. Chamola, V., et al. (2017). Latency aware mobile task assignment and load balancing for edge Cloudlets. In Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on (pp. 587-592). IEEE.
15. Mukherjee, A., et al. (2014). Angels for distributed analytics in iot. In Internet of Things (WF-IoT), 2014 IEEE World Forum on (pp. 565-570). IEEE.
16. Ai, Y., Peng, M., & Zhang, K. (2018). Edge Computing technologies for Internet of Things: a primer. Digital Communications and Networks, 4(2), 77-86.
17. Kecskemeti, G., Casale, G., Jha, D. N., Lyon, J., & Ranjan, R. (2017). Modelling and simulation challenges in internet of things. IEEE Cloud Computing, 4(1), 62-69.
18. Aazam, M., & Huh, E. N. (2015). Fog Computing micro datacenter based dynamic resource estimation and pricing model for IoT. In Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on (pp. 687-694). IEEE.
19. Svorobej, S., Takako Endo, P., Bendechache, M., Filelis-Papadopoulos, C., Giannoutakis, K. M., Gravvanis, G. A., ... & Lynn, T. (2019). Simulating Fog and Edge Computing scenarios: An overview and research challenges. Future Internet, 11(3), 55.
20. Varga A, Hornig R. An overview of the OMNeT++ simulation environment. In: Simutools '08 Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshop; 2008; Marseille, France.
21. Zeng, X., Garg, S. K., Strazdins, P., Jayaraman, P. P., Georgakopoulos, D., & Ranjan, R. (2017). IOTSim: A simulator for analysing IoT applications. Journal of Systems Architecture, 72, 93-107.
22. Zhao, X., Zhao, L., & Liang, K. (2016). An Energy Consumption Oriented Offloading Algorithm for Fog Computing. In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (pp. 293-301). Springer, Cham
23. Fernández-Caramés, T. M., & Fraga-Lamas, P. (2019). Towards Next Generation Teaching, Learning, and Context-Aware Applications for Higher Education: A Review on Blockchain, IoT, Fog and Edge Computing Enabled Smart Campuses and Universities. Applied Sciences, 9(21), 4479.
24. Santoro, D., Zozin, D., Pizzolli, D., De Pellegrini, F., & Cretti, S. (2017). Foggy: a platform for workload orchestration in a Fog Computing environment. In Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on (pp. 231-234). IEEE.
25. Shi, C., Lakafosis, V., Ammar, M. H., & Zegura, E. W. (2012). Serendipity: enabling remote computing among intermittently connected mobile devices. In Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing (pp. 145-154). ACM.

26. Liyanage, M., Chang, C., & Srirama, S. N. (2018). Adaptive mobile Web server framework for Mist computing in the Internet of Things. International Journal of Pervasive Computing and Communications.
27. Weicker, R. P. (1984). Dhrystone: a synthetic systems programming benchmark. Communications of the ACM, 27(10), 1013-1030.
28. Sonmez, C., Ozgovde, A., & Ersoy, C. (2019). Fuzzy workload orchestration for edge computing. IEEE Transactions on Network and Service Management, 16(2), 769-782.
29. Yuan, Y., & Shaw, M. J. (1995). Induction of fuzzy decision trees. Fuzzy Sets and systems, 69(2), 125-139.
30. Dong, Z. Y., Zhang, Y., Yip, C., Swift, S., & Beswick, K. (2020). Smart campus: definition, framework, technologies, and services. IET Smart Cities, 2(1), 43-54.
31. Guo, J., Song, B., He, Y., Yu, F. R., & Sookhak, M. (2017). A survey on compressed sensing in vehicular infotainment systems. IEEE Communications Surveys & Tutorials, 19(4), 2662-2680.

Charafeddine Mechalikh is a PhD student at the faculty of science of Tunis. He was born on July 13th, 1993 in Algeria and got his master’s degree in industrial computer science (2017), from Kasdi Merbah University of Ouargla, Algeria. His PhD project is about modeling and simulation of IoT-Edge Computing environments, which has been the subject of this article, and other papers that have been published in well-rated conferences such as “A Scalable and Adaptive Tasks Orchestration Platform for IoT” (IWCMC 2019). His area of interests is the Internet of Things, simulation and modeling, and Edge Computing.

Hajer Taktak is a Doctor that was born on July 26th, 1988 in Tunisia and got a software Engineering from the national institute of applicate sciences and technologies (INSAT) in Tunis, Tunisia. She did the final study project in LIRIS laboratory in France. The project is about A Privacy-aware Execution Model for Data Services. This latter was the subject of a paper which was published in CAiSE conference. “A semantic-based method for natively context-aware web services discovery” is the second paper which was submitted in Mobile Data Management conference. The thesis has been focused on a common method for both adaptive services and user interfaces. This research has been published in two international journals and several international conferences.

Faouzi Moussa is a professor at the faculty of sciences of Tunis, was born in Tunis, Tunisia. He got a doctoral dissertation from Valenciennes university in France after getting a master’s degree in computer management. He also obtained an IT clearance in 2005 and wrote many papers on several topics such as human-computer interaction, context awareness and pervasive computing. He is currently managing a PhD students’ team in the faculty of science of Tunis regarding web services, artificial intelligence, mobile computing etc. He published many papers in well-rated conferences. Among his editorials: the journal Towards a Runtime Evolutionary Model of User-Adapted Interaction, Petri Nets Context Modeling for the Pervasive Human-Computer Interfaces (CONTEXT, 2013), XML in Formal Specification, Verification and Generation of Mobile HCI (HCI, 2011) etc. He is also a producer and a radio host.

Received: March 01, 2020; Accepted: August 01, 2020

Computer Science and Information Systems 18(1):67–91https://doi.org/10.2298/CSIS200330035A

DroidClone: Attack of the Android Malware Clones - A Step Towards Stopping Them ⋆

Shahid Alam[1] and Ibrahim Sogukpinar[2]

[1] Department of Computer Engineering, Adana Alparslan Turkes Science and Technology University, Adana, Turkey
salam@atu.edu.tr

[2] Department of Computer Engineering, Gebze Technical University, Gebze, Turkey
ispinar@gtu.edu.tr

Abstract. Code clones are frequent in use because they can be created fast with little effort and expense. Especially for malware writers, it is easier to create a clone of the original than writing a new malware. According to the recent Symantec threat reports, Android continues to be the most targeted mobile platform, and the number of new mobile malware clones grew by 54%. There is a need to develop techniques and tools to stop this attack of Android malware clones. To stop this attack, we propose *DroidClone* that exposes code clones (segments of code that are similar) in Android applications to help detect malware. *DroidClone* is the first such effort uses specific control flow patterns for reducing the effect of obfuscations and detect clones that are syntactically different but semantically similar up to a threshold. *DroidClone* is independent of the programming language of the code clones. When evaluated with real malware and benign Android applications, *DroidClone* obtained a detection rate of 94.2% and false positive rate of 5.6%. *DroidClone*, when tested against various obfuscations, was able to successfully provide resistance against all the trivial (Renaming methods, parameters, and nop insertion, etc) and some non-trivial (Call graph manipulation and function indirection, etc.) obfuscations.

Keywords: Android, Code Clones, MAIL, Malware Analysis and Detection, TF-IDF, Machine Learning.

1. Introduction

According to the McAfee threat report [34], number of malware (*clones*) found in the Google play increased by 30% in 2017. According to the Symantec [42, 43] threat reports, Android continues to be the most targeted mobile platform, and the number of new discovered mobile malware (*clones*) grew by 54% from 2016 to 2017. Further to this simple attack of *clones*, there are also Android malware *clones of clones*, i.e., *clones* of a malware family which itself is a *clone*. For example, *DroidKungFu1*, *DroidKungFu2*, *DroidKungFu3* and *DroidKungFu4* are 4 different families of the original Android *DroidKungFu* malware, and each of these 4 families have there own *clones* [47].

⋆ The work presented in this paper is an expansion of the authors' previously published work in conference paper [2].

Malware writers are using stealthy mutations (obfuscations) to continuously develop malware clones, thwarting detection by signature-based detectors. To protect Android applications against reverse engineering attacks, even legitimate applications are obfuscated [15]. Similar techniques are used by malware developers to prevent analysis and detection. Obfuscation can be used to make the code more difficult to analyze, or to create *clones* of the same malware in order to evade detection.

Code clones are frequent in use because they can be created fast with little effort and expense. Especially for malware writers, it is easier to create a clone of the original than writing a new malware. There are different ways to create code clones, some of them are: when a programmer copies and paste fragment of code after minor editing; part of a code is embedded (piggybacked) inside another code/program; when a code is obfuscated to create copies, which are syntactically different but semantically similar. Finding code clones can be useful for: detecting malicious software, plagiarism and copyright infringement; bug detection; code simplification; and code maintainability.

In general, clones are divided into four types [39]. *Type*-1 (Exact clones): Exact copies of each other except white spaces, blanks and comments, etc. *Type*-2 (Renamed clones): Similar copies except name of variables, literals, functions, etc. *Type*-3 (Near miss clones): Similar except all the above and some added/removed statements, etc. *Type*-4 (Semantic clones): Syntactically different but semantically similar.

There are several researches [9, 10, 16, 18, 22–25, 27–30, 32, 35, 45, 46] that have focused on code clone detection using different approaches, such as: (1) *textual*, based on token [9] and pattern [18] matching, and longest common subsequence [16]. (2) *lexical*, based on frequent subsequence mining [30], cosine similarity [46], and suffix array [35]. (3) *syntactical*, based on abstract syntax tree (AST) [10, 28, 45], hashed blocks [24], and AST to vectors [25]. (4) *semantic*, based on program dependence graph (PDG) [23, 27, 29, 32], and serialized AST [22].

Only two [27, 29] of the above approaches claim to detect *Type*-4 cloning, but the DR (detection rate) of [29] is low ranging from 17.3% – 45.8% and there is no DR to report for [27]. These two approaches can only find source code and not native code clones, and hence is also dependent on the programming language of the code. Both of them find isomorphic similarity of subgraph PDGs to detect clones, which is compute-intensive and is not scalable.

Clone detection technique can be improved by combining several different types of methods or reimplementing systems using a different programming language. It is hard to determine which is the best tool for detection because every tool has its strengths and weaknesses. Since text-based and token-based techniques have high recall and AST-based techniques have high precision, these techniques may be merged in a tool to get high recall and precision results. A PDG-based technique detects only Type-3 clones; this technique may be extended to detect Type-1 and Type-2 clones besides Type-3 clones.

Type-1 and Type-2 clones are easier to detect than Type-3 clones. Sequence alignment algorithms with gaps may potentially be used to detect Type-3 clones. To make clone detection independent of the programming language of the clone and also combine different techniques in one we use a new intermediate language MAIL (Malware Analysis Intermediate Language) [3] to improve clone detection.

As mentioned earlier, Android mobile platform is facing an attack of clones. We need to find ways for detecting and stopping this attack. There are several researches [5,13,17,20,31,41] that have focused on Android malware clones detection. [17] uses API call graphs, [41] uses components based API calls to find code reuse, and [13] uses a text-based near-miss source code clone detector. [5] mines dominant API calls to find the reuse of malicious modules to detect malware. [31] captures system calls at thread level to detect malicious clones embedded inside an Android application and [20] uses SDHash [38] to detect application similarity for detecting malware. We believe using control flow patterns is a more general technique for malware analysis and detection than using API call patterns, and it is difficult for a malware to change control flow patterns than changing API call patterns of a program, for evading detection.

In this paper, we propose *DroidClone* as a step towards detecting and stopping these clones in Android malware. For this purpose, we utilize the new intermediate language MAIL [3] that helps us use specific control flow patterns to reduce the effect of obfuscations and unlike [5, 17, 41] can detect malware clones at a much-refined level that helps detect smaller size clones. Our technique, unlike [13, 20], can detect malware clones that are syntactically different but semantically similar up to a certain threshold. A malware writer has to employ an excessive (beyond a certain threshold, which is difficult to find) control flow obfuscation to create a clone to evade detection by such an anti-malware. [31] is based on dynamic analysis, may not cover all the program paths and hence can miss some malicious behaviors. *DroidClone* performs static analysis and covers all the program paths. *DroidClone* finds clones at a much refined level than [5]. Moreover, *DroidClone* can process and analyse Android native code clones.

DroidClone, when tested with 4180 real malware and benign Android applications using different validation methods, obtained detection rates (DRs) in the range of 90.3% − 94.2% and false positive rates (FPRs) in the range of 4% − 11%. *DroidClone*, when tested against various obfuscations (malware variants), was able to successfully provide resistance against all the trivial (Renaming methods, parameters, and nop insertion, etc) and some non-trivial (Call graph manipulation and function indirection, etc.) obfuscations.

Following are the major contributions of this paper:

- *DroidClone* is the first such effort that uses a new intermediate language MAIL for building the signatures to find Android clones for malware detection. We first build a MAIL CFG (control flow graph) and then extract specific control flow patterns to reduce the effect of obfuscations and detect code clones that are syntactically different but semantically similar (i.e., *Type*-3 and *Type*-4 clones) up to a threshold. Sometimes malicious code (bytecode or native code) clone consists of only a few statements, such as setting up a few registers and a jump to the actual malicious code location. To accommodate such clones, *DroidClone* uses MAIL blocks at a statement level, and can detect clones at a much-refined level (smaller size clones $\geq$ 3 statements) than other such techniques.

- We extend TF-IDF with a new weighting scheme for feature (control flow patterns) selection and improve the clone detection scheme. Moreover, we perform serialization of a MAIL CFG into strings of specific control flow patterns at block level, which also improves application matching.
- Most of the Android applications are written in C/C++ and Java. It is necessary to build a cross-platform clone detector for such applications. *DroidClone* designs cross-platform signatures for Android at the native code level, and is able to process, analyse and detect malware cloned as either bytecode or native code. This makes *DroidClone* independent of the programming language of the Android code clone.
- This paper significantly enhances the previous version of *DroidClone* [2] by:
  - updating and enhancing the clone detection scheme;
  - using only MAIL blocks for building the signatures, in turn improving accuracy and also runtime of the overall system;
  - improving the feature selection method;
  - lowering the false positive rate (8.5% ⇒ 5.6%);
  - increasing the Accuracy (91% ⇒ 94.3%);
- We provide cross-validation of *DroidClone*, using two methods holdout and *n*-fold, which is a more systematic way of determining the performance and accuracy of a system than is provided by most other similar works. We also test the resistance of *DroidClone* against various obfuscations.

The remainder of this paper is organized as follows. We discuss related works in Section 2. We present a detailed overview of our approach, its design and implementation in Section 3. Section 4 presents the evaluation and comparison of our approach with six other such works. Section 5 finally concludes the paper and presents some future works.

2. Related Works

A very detailed survey of the research done on code clones is presented in [39]. In this section, we briefly highlight seven recent research works of finding Android clones for detecting malware.

Lin et al. [31] propose SCSdroid, which captures thread-grained system call sequences during runtime to detect malicious clones embedded inside an Android application. A thread-grained system call sequence is the system calls recorded for a thread. The authors believe that the malicious behavior happens during a single thread, so mixing the system calls recorded for a process and for a thread can make it difficult to identify the malicious behavior. Android applications are usually multi-threaded, so it is possible to miss some malicious actions that encompass multiple threads.

Sun et al. [41] propose a technique using CBCFG (component-based control flow graph). CBCFG is a graph of Android APIs as nodes and their control flow precedence relationship as edges. These CBCFGs are then used to detect code reuse in Android repackaged applications and malware variants. The technique may not be able to detect malware applications that obfuscate by using fake API calls, hiding API calls (e.g, using class loading to load an API at runtime), inlining APIs (e.g, instead of calling an external API, include the complete API code inside the app), and reflection (i.e, creation of programmatic class instances and/or method invocation using the literal strings, and a subsequent encryption of the class/method name can make it impossible for any static analysis to recover the call).

Deshotels et al [17] use API call graph signatures and machine learning to identify piggybacked applications. Piggybacking can be used to inject malicious code into a benign app. First, a signature is generated of a benign application and used to identify whether another application is a piggyback of this app. This technique has the same limitations as discussed above regarding detection schemes based on API calls.

Chen et al. [13] present a technique that uses NiCad [16], a near-miss clone detector, to detect Android malware. First, they develop signatures from a subset of malware applications by finding clone classes in these applications and then use these signatures to find similar malware applications in the rest of the malware applications. Their clone detector works at the Java source code level and hence is not able to detect native Android clones. NiCad compares the source code linewise using an optimized *longest common subsequence* algorithm to detect similar clones. This is a good text-based technique, but may not be efficient for detecting malware clones. For example, it may be defeated just by changing the names of the functions and variables, and hence may not be able to detect clones that are syntactically different but semantically similar.

Faruki et al. [20] propose a technique to use the similarity of applications to detect Android malware. They use SDHash [38], a statistical approach for selecting fingerprinting features. Therefore the results in the paper have a better false positive rate. Although the FPR reported in [20] is low (1.46%), because of the SDHash technique used, whose main goal is to detect very similar data objects, the ability to detect malware clones is much lower than the technique proposed in this paper.

Alam et al. [5] propose a technique that mines the dominant tree of API calls in an Android application to detect malware. Reused dominant API modules in an Android application are extracted to find clones. The technique works at the dominant API level and is more suitable for finding coarse (higher) level clones. Whereas, *DroidClone* works at MAIL CFG block level and is more refined.

Kalysch et al. [26] propose a technique based on the centroid of CFGs to measure the similarity between Android native codes. The centroid approach is faster than other approaches for matching CFGs. They achieve a DR of 89% and FPR of 10.8%. Disassembling native code to an intermediate code (e.g., a CFG or MAIL) is a non-trivial problem. The work presented in [26] processes only native code libraries and not native code applications. Moreover, they cannot process Intel x86 and 64 bit ARM native code, because of the limitations of the tools used for disassembling. Whereas, DroidClone processes both native code libraries and standalone applications, for both Intel x86 (32 and 64 bit) and ARM (32 and 64 bit) architectures.

Except [26], none of the other above works can find clones in Android native code. Some of them, such as [13] and [20] may not be able to detect syntactically different but semantically similar clones. The technique proposed in [31] is based on dynamic analysis and hence suffers from the known fact that it does not cover all the paths of a program, and may miss certain malicious behaviors.

The techniques proposed in [41], [17] and [5] are based on API calls. In general, changing (obfuscating) control flow patterns is more difficult (i.e, it needs a comprehensive change in the program) than changing just API call patterns of a program to evade detection. API based techniques look for specific API call patterns (including call sequences) in Android malware programs for there detection, which may also be present in Android benign applications that are protected against reverse engineering attacks. These API call patterns can be packing/unpacking, calling a remote server for encryption/decryption, dynamic loading, and system calls, etc.

3. Overview of the System

Figure 1 provides an overview of the *DroidClone* architecture. Android applications are present either in bytecode or native code. First, the Android applications in bytecode are compiled to native code. Then, we use the tool DroidNative [4] to translate the native code to a MAIL program which is transformed into a CFG, called MAIL CFG for malware analysis and detection. In the next Sections, we explain these steps, and why and how MAIL is adapted for analysis and detection of Android malware code clones.

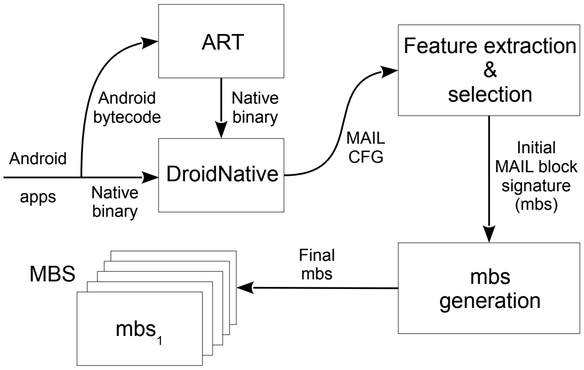


Fig. 1. Overview of *DroidClone*.

3.1. Android Runtime (ART) and DroidNative

Android applications are distributed in the form of APKs (Android application packages), and contains code as Android bytecode (in a *dex* file) and precompiled native binaries. The *dex* (Dalvik Executable Format) [44] files are specific to Android platform. These *dex* files were used to run under the Dalvik Virtual Machine [6] on older Android versions, and are similar to Java class files. Starting with Android 5.0, *dex* files are compiled to native binaries before they are installed. This task is performed by the ART (Android runtime) [7]. ART uses ahead-of-time (dex2oat) compiler for this purpose, which improves the overall execution efficiency and reduces the power consumption of an Android application.

In this paper we use DroidNative [4] to translate an Android native binary (x86 and ARM) to a MAIL program. This allows us to build cross-platform signatures, and process, analyse and detect malware cloned as either bytecode or native code. DroidNative first uses *dex2oat* to compile APKs into native code and then translate the native code to MAIL programs for clone detection.

3.2. Malware Analysis Intermediate Language (MAIL)

Intermediate languages have long been used in compilers to translate the source code into a form that is easy to optimize and provide portability. We apply the same concepts to malware analysis and detection. Several intermediate languages [3, 12, 14, 19, 40] have been developed for optimized analysis and detection of malware. The reason for using MAIL in *DroidClone* is that it has certain advantages over the other languages [12, 14, 19, 40], such as automating and minimizing the effect of obfuscations that makes it suitable for finding clones. Moreover, its publicly available formal model and open-source tools make it easy to use.

There are eight basic statements (e.g., assignment, control and conditional, etc.) in MAIL that can be used to represent the structural and behavioral information of an assembly program. Each statement in a MAIL program is assigned a type also called a *pattern*. This pattern can be used for matching to assist in clone detection.

Patterns for Annotation The MAIL language contains a total of 21 patterns as shown in Table 1. Each pattern represents the type of a MAIL statement and can be used for easy comparing and matching of MAIL programs.

To assist in matching, a MAIL program is annotated with these patterns. In this paper, an annotated MAIL program is used for matching clones to find malicious code in Android applications. For example, a MAIL jump statement with a constant value and one without a constant value are two different statements, and a MAIL jump statement with a reference to the stack and one with no reference to the stack are two different statements. The MAIL program annotations help make this distinction.

3.3. Control flow analysis

To evade detection, various obfuscations are implemented to create different types of clones. To build resistance against various obfuscations and successfully find

Table 1. Patterns used in MAIL. r_0, r_1, r_2 are the general purpose registers, zf and cf are the zero and carry flags respectively, and sp is the stack pointer.

Pattern	**Description**
`ASSIGN`	Assignment statement, *e.g.* $r_0 = r_0 + r_1$;
`ASSIGN_CONSTANT`	Assignment statement with a constant, *e.g.* $r_0 = r_0 +$ 0x1234;
`CONTROL`	Control statement with unknown jump, *e.g.* if ($zf = 1$) JMP [$r_0 + r_1 +$ 0x1234];
`CONTROL_CONSTANT`	Control statement with known jump, *e.g.* if ($zf = 1$) JMP 0x1234;
`CALL`	Call statement with unknown call, *e.g.* CALL r_2;
`CALL_CONSTANT`	Call statement with known call, *e.g.* CALL 0x1234;
`FLAG`	Statement with a flag, *e.g.* $cf = 1$;
`FLAG_STACK`	Statement that includes flag with stack, *e.g.* $eflags$ = [$sp = sp$ – 0x1234];
`HALT`	Halt statement, *e.g.* halt;
`JUMP`	Jump statement with unknown jump, *e.g.* JMP [$r_0 + r_2 +$ 0x1234];
`JUMP_CONSTANT`	Jump statement with known jump, *e.g.* JMP 0x1234
`JUMP_STACK`	Return jump, *e.g.* JMP [$sp = sp$ – 0x1234]
`LIBCALL`	Library call, *e.g.* compare(r_0, r_2);
`LIBCALL_CONSTANT`	Library call with a constant, *e.g.* compare(r_0, 0x1234);
`LOCK`	Lock statement, *e.g.* lock;
`STACK`	Stack statement, *e.g.* r_0 = [$sp = sp$ – 0x1];
`STACK_CONSTANT`	Stack statement with a constant, *e.g.* [$sp = sp$ + 0x2341] = 0x1234;
`TEST`	Test statement, *e.g.* r_0 and r_2;
`TEST_CONSTANT`	Test statement with a constant, *e.g.* r_0 and 0x1234;
`UNKNOWN`	Unknown assembly instruction that cannot be translated.
`NOTDEFINED`	The default pattern, and is assigned to every newly created statement.

clones, we extract control flow patterns in a MAIL program of an Android application. For extracting control flow patterns we perform control flow analysis and build a control flow graph (CFG) [1] of a MAIL program as follows.

Definition 1 *A* ***basic block*** *is a sequence of MAIL statements, and there are no branches except at the entry and exit points. MAIL statements starting a basic block can be: the first statement; a call to a function or a return statement; a statement following a branch; and target of a branch or a function call. MAIL statements ending a basic block can be: the last statement; call to a function; a return statement; and an unconditional or conditional branch.*

Definition 2 *Control flow edge is an edge between two basic blocks. A CFG is a directed graph $G = (V, E)$, where V is a set of basic blocks and E is a set of control flow edges. The CFG of a MAIL program represents all the paths that can be taken* *during program execution. An annotated **MAIL CFG** is a CFG such that each statement of the CFG is assigned a MAIL Pattern.*

An annotated MAIL CFG is built for each function in a MAIL program. An example of an annotated MAIL CFG of a function of an Android malware program is shown in Table 2. For simplification, in the rest of the paper, an *annotated MAIL CFG* is just called a *MAIL CFG*. We describe in the following sections, how a MAIL CFG is used at a block level for matching clones to find malicious code in Android applications.

3.4. Preprocessing and Feature Extraction

This Section describes how a MAIL CFG on a block-level is serialized to a string for efficient matching. A MAIL CFG (program) consists of functions. The end of a function is tagged with the symbol `EOF`. In a MAIL CFG, a block starts with the tag `START` and ends with the tag `END`. All the MAIL statements inside these two tags become part of the block. Table 2 shows one of the CFG's for one portion of a MAIL program (a malware), and contains 1 function and 4 blocks. These blocks are parsed using MAIL patterns into block strings as follows:

Table 2. An annotated MAIL CFG (control flow graph) of a function of an Android malware program.

```
Num Offset           MAIL Statement             Pattern     Block    Jump To

 0  19018  r12 = sp - #8192; start_function_0 [  ASSIGN_C]  START
 1  1901c                    r12 = [r12, #0]; [  ASSIGN_C]
 2  19020  [sp=sp+0x1] = r7;[sp=sp+0x1] = lr; [     STACK]
 3  19024                      sp = sp - #16; [     STACK]
 4  19026                            r7 = r0; [    ASSIGN]
 5  19028         [sp, #0] = r0;sp = sp - 0x1 [     STACK]
 6  1902a                            r5 = r1; [    ASSIGN]
 7  1902c                            r6 = r2; [    ASSIGN]
 8  1902e                      [r5, #8] = r6; [  ASSIGN_C]
 9  19030           if (r6 == 0 jmp 0x1903a); [ CONTROL_C]   END       1903a
10  19032                    r2 = [r9, #120]; [  ASSIGN_C]  START
11  19036                    r3 = r5  >>  #7; [  ASSIGN_C]
12  19038                      [r2, r3] = r2; [    ASSIGN]   END
13  1903a                        lr = #12401; [  ASSIGN_C]  START
14  1903e                        lr = #29198; [  ASSIGN_C]
15  19042                        r0 = #13152; [  ASSIGN_C]
16  19046                        r0 = #28596; [  ASSIGN_C]
17  1904a                            r1 = r5; [    ASSIGN]
18  1904c                             jmp lr; [      JUMP]   END
19  1904e                       sp = sp + 20; [  ASSIGN_C] START
20  1904f  r6 = [sp=sp-0x1];pc = [sp=sp-0x1]; [    JUMP_S]   END EOF
```

```
block 1: ACACSSASAAACCC, block 2: ACACA,
block 3: ACACACACAJ and block 4: ACJS
where: ASSIGN ⇒ A, ASSIGN_C ⇒ AC, JUMP ⇒ J, JUMP_S⇒ JS,
CONTROL_C ⇒ CC, STACK ⇒ S
```

Each block in a MAIL CFG (program), in addition to the above string, is also assigned an initial weight, i.e., the number of times (frequency) it appears in the MAIL CFG. After these initial assignments, we select features and assign the final weight to each block and then build the database of MAIL block signatures for malware detection.

With these initial assignments to each block in a MAIL CFG, in the next two Sections, we describe how features are selected and the final MAIL block signatures are build from a dataset of malware and benign samples, for malware/clone detection.

3.5. Feature Selection

Term Frequency and Inverse Document Frequency (TF-IDF) [33] is widely used, and often considered as an empirical method, in data mining to separate/select relevant features in a set of documents/samples. TF-IDF is used as the amount of information of a term weighted by its occurrence of probability. This Section describes how we adapt TF-IDF weighting method to assign the final weight to a MAIL block, and how this weight is used to select features (MAIL blocks) from a MAIL program.

Let $P = \{p_1, p_2, p_3, ..., p_N\}$ denote the N MAIL programs (preprocessed, as described in the above Section) in a dataset of either malware or binary samples, and $p = \{b_1, b_2, b_3, ..., b_n\}$, where n is the total number of MAIL blocks in program (sample) p. We define the TF and IDF of a MAIL block $b_i \in p$ as follows:

$$TF_i = \frac{f_i}{n} \qquad and \qquad IDF_i = log\left(\frac{N}{M_i}\right)$$

where, f_i is the number of times (frequency) b_i appears in a MAIL program p; and M_i is the number of all the MAIL programs with b_i in it.

Based on these definitions, we formulate our weight assigning approach to the MAIL block b_i as follows:

$$W_i = TF_i \times IDF_i \tag{1}$$

We only keep b_i, if $S_i \geq 3$ and $W_i \geq 0.5$, where S_i is the number of statements in b_i. These minimum values of S_i and W_i are computed empirically.

3.6. Signatures of MAIL blocks

We define signature of a MAIL block $b_i \in p$, in the vector space, as $s_i = \{Sig_i, W_i\}$, where Sig_i represents the MAIL statements in the block as a string of MAIL patterns, as described in Section 3.4.

As an example, for the MAIL CFG shown in Table 2, the following vector is generated: {ACACSSASAAACCC:1.58, ACACA:6.28, ACACACACAJ:4.04 and ACJS:15.35}. There are 4 blocks in this MAIL program and are also found in other portions of the same program (not shown here). 1.58, 6.28, 4.04 and 15.35 are the weights assigned, as defined in equation (1), to each block with respect to the block frequency in the whole MAIL program. Only blocks with 3 or more statements are used for generating the signature. The last block in this MAIL program is discarded because it contains only 2 statements. As we can see, this last block contains a typical epilogue of an assembly program, which is not essential for malware analysis and detection.

After building the signature of a MAIL block, the final signature of a MAIL program p is build as: $M_p = \{s_1, s_2, s_3, ..., s_n\}$. We take the common block signatures among MAIL programs out, and build our database of signatures as follows:

$$V = \bigcup_{p=0}^{N} M_p \tag{2}$$

We build signatures' database of both malware (V_m) and benign (V_b) MAIL programs separately using equation (2). To further improve feature selection, we take malware block signatures (V_m) that are common in benign (V_b) out, and build the final database of malware block signatures (MBS) as follows:

$$MBS = \{x \mid x \in V_m \wedge x \notin V_b\} \tag{3}$$

3.7. Malware/Clone Detection

Figure 2 gives an overview of how malware/clone detection is carried out in *DroidClone*. For malware/clone detection we process a new sample as described in Section 3.4. At this time signature of the new sample contains all its block strings and there respective initial frequencies. After this, a similarity score is computed for the new sample as follows:

$$SimScore = (\frac{\sum_{i=0}^{n} x_i}{N} \times 100) \times (\frac{\sum_{i=0}^{n} y_i}{n} \times 100) \tag{4}$$

where, n is the total number of blocks in the new sample; N is the total number of blocks in MBS; x_i is the similarity value of the ith block (b_i) in the new sample; and x_i and y_i are computed as follows:

$$x_i = \begin{cases} f_i \times W_i & b_i \in MBS \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad y_i = \begin{cases} 1 & b_i \in MBS \\ 0 & \text{otherwise} \end{cases}$$

where f_i is the initial frequency (Section 3.4) and W_i is the final weight (Equation (1)) of b_i.

A *SimScore* is assigned to each sample in the dataset using equation (4). The samples with there *SimScore* values are used for training a classifier for malware detection/classification. A new sample is tagged as malware if *SimScore* of the sample is $\geq$ a certain threshold.

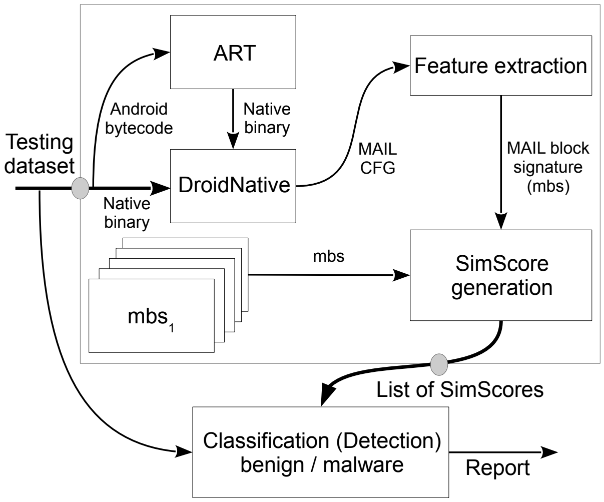


Fig. 2. Overview of malware/clone detection in *DroidClone*.

4. Experimental Evaluation

We carried out an empirical study to analyse the correctness and efficiency of our approach. We carried out two experiments, each using a different validation technique. In the first experiment, we used the holdout cross-validation and in the second experiment, we used n-fold cross-validation. We present in this section the dataset, evaluation metrics, the empirical study (the two experiments), obtained results and analysis. We also present the results of *DroidClone* resistance test against various obfuscations.

4.1. Dataset

Our dataset for the experiments consists of 4180 Android applications. Of these, 2050 are real Android malware programs collected from three different resources [8, 36, 47], and the other 2130 are benign programs containing applications downloaded from Google Play, Android 5.0 system programs, and shared libraries. Tables 3 and 4 shows distribution of the 2130 benign and 2050 malware samples respectively. The dataset also includes 284 Android malware variants. We picked 32 samples from the Miscellaneous class of malware families to generate 9 different classes of malware variants using the obfuscation techniques listed in Table 8. The purpose of generating these variants were to test *DroidClone* against various obfuscation techniques. These variants were also included in the other two validation tests (Sections 4.5 and 4.6).

Table 3. Distribution (native code or byte-code) of the 2130 Android benign samples

Code type	Number of samples
Byte code	1830
Native code	300

Partitioning of the dataset for different experiments, including threshold computation, holdout cross-validation, and n-fold cross-validation is shown in Table 5.

The benign dataset includes both native code (executables and libraries) and byte code. Some of the benign native code applications are: *atrace* – captures kernel events; *bugreport* – reports stack traces and diagnostic information etc; and *bmgr* – backup manager.

The malware dataset shows a variety of samples from different families and includes both native and byte code. The 15 native code malware are standalone applications and the other 4 are libraries. Some of the malware native code applications are: *asroot* – A root exploit and an ELF32 ARM executable file detected by 31 anti-malware programs on virustotal.com, such as Kaspersky, McAfee, and Sophos, etc. *droidpak* – A PE32 Intel x86 executable file detected by 50 anti-malware programs on virustotal.com. It spreads to Windows PC from an infected Android phone. Some of its malicious features of the Android version include sending, uploading and deleting SMS messages, and uploading contacts and location, etc. Most of the Android byte code malware classes are piggybacked applications (ADRD, all of the DroidKungFu families, DroidDream, DroidDreamLight, Geinimi, JSMSHider, and Pjapps). GoldDream, YZHC and most of the malware in the Miscellaneous class are standalone malware applications. DREBIN also contains both piggybacked and standalone malware applications, such as GingerMaster and FakeInstaller families of malware.

4.2. Test Platform

All experiments were run on an Intel® Core(TM) i-7-4510U CPU @ 2.00 GHz with 8 GB of RAM, running Windows 8.1. The ART compiler [7], cross built on the above machine, was used to compile Android applications (malware and benign) to native code.

4.3. Metrics

Before performing the evaluation, we first define our metrics. **DR** (Detection Rate), also called the true positive rate, corresponds to the percentage of samples correctly recognized as malware out of the total malware dataset. **FPR** (False Positive Rate) corresponds to the percentage of samples incorrectly recognized as malware out of the total benign dataset. **Accuracy** is the fraction of samples, including malware and benign, that are correctly detected as either malware or

Table 4. Class distribution of the 2050 Android malware samples. The first 19 (upto DroidPak in the left column) are Android native code (ARM and Intel x86) malware. The rest of the 2031 are Android byte code malware.

Class/family	Number of samples	Class/family	Number of samples
Asroot	3	DroidDream	16
CarrierIQ	8	DroidDreamLight	46
ChathookPtrace	2	DroidKungFu1	35
Cuttherope	1	DroidKungFu2	30
DroidPak	5	DroidKungFu3	310
ADRD	21	DroidKungFu4	95
Airpush	11	Geinimi	68
Appinventor	17	GoldDream	45
AnserverBot	186	JSMSHider	15
BaseBridge	120	KMin	50
Coinkrypt	3	Pjapps	75
DREBIN	494	YZHC	22
DroidChameleonVariants	284	Miscellaneous[1]	88

[1] Includes Android malware samples, AntiObamaScan, DeathRing, FakeDefender, FakeJobOffer, FakeNotify, FakeTimer, FBIRansomLocker, RansomCollection, VoiceChange, and WindSeeker, etc.

benign. ROC (Receiver Operating Characteristic) curve is a graphical plot used to depict the performance of a binary classifier. **AUC** (Area Under the ROC Curve) [21] is equal to the probability that a detector/classifier will correctly classify a sample.

4.4. Threshold Computation

If the *SimScore* of a sample is $\geq$ a certain threshold the sample is detected as malware. That threshold must be determined experimentally. In order to compute the threshold, we separated out 3344 samples, including 1640 malware and 1704 benign. To optimize the results and pick the best threshold automatically we used a RandomTree classifier. During this testing, we perform 10 iterations, each time with a different 334 samples in the testing set and the remaining 3010 samples in the training set. RandomTree built a Decision Tree of size 825 nodes for this dataset, every time, i.e., at each decision node, picking a different *SimScore* ranging from ~0 to ~200, same as the range of the *SimScore* of the 1704 benign samples. Our priority is the DR, therefore at the end of this experiment, a threshold of 15 was picked based on the best DR results. This threshold was then used in the holdout cross-validation of *DroidClone* in Section 4.5.

Table 5. Partitioning of the dataset (total 4180 samples ⇒ 2130 benign and 2050 malware) for different experiments.

Experiment	Total number of samples (benign/malware)	Training samples	Testing samples
Threshold computation	3344[1] (1704/1640)	3010	334
Holdout cross validation	4180 (2130/2050)	3344[1]	836
N-fold cross validation	4050 (2025/2025)	3645[2]	405[2]

[1] Only the training data was used for computing the threshold.

[2] In this experiment we perform 10 (N = 10) iterations, each time with a different 405 samples in the testing set and the remaining 3645 samples in the training set.

Distribution of the 2025 benign and 2025 malware samples (used in n-fold cross-validation) based on their SimScore along with the computed threshold is shown in Figure 3. This distribution of 4050 samples contains 836 samples that were not used to compute the threshold.

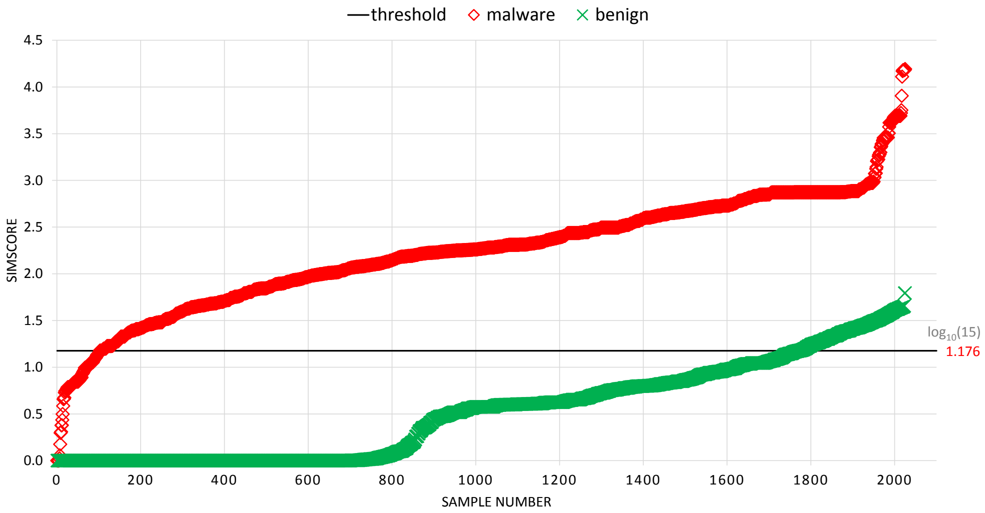


Fig. 3. Distribution of the 2025 malware and 2025 benign samples based on there *SimScore* as defined in equation (4). The threshold of 15 plotted here is computed by RandomTree algorithm (classifier) as explained in Section 4.4.

4.5. Holdout Cross Validation

After selecting the threshold, we carried out the holdout validation using our dataset of 4180 samples. In this method, we randomly divided the data into two parts. The larger part was used for training and the smaller part was used for testing. To keep the training set separate from the testing set, for the larger part we used the same dataset, i.e., the 3344 samples (1704 benign and 1640 malware samples), as used in Section 4.4 to compute the threshold. The smaller part consisted of a total of 836 samples, out of which 426 were benign and 410 malware.

Using the threshold of 15, we carried out the holdout validation experiment as follows.

First, we built the *MBS* database of the 3344 training samples (already labeled as malware or benign) using equation (3). Then we computed SimScore for each of the 836 testing samples (not yet labeled, i.e., unknown) using equation (4). Computing SimScore for each of the testing samples depends on the *MBS* database as explained in Section 3.7. If the SimScore of a testing sample was $\geq$ 15, it was tagged/labeled as malware otherwise benign.

The results of this experiment, in the form of a confusion matrix, are shown in Table 6. Based on these results we compute DR, FPR and Accuracy of *DroidClone* as follows:

$$DR = \frac{386}{410} \times 100 = 94.2\%$$

$$FPR = \frac{24}{426} \times 100 = 5.6\%$$

$$Accuracy = \frac{386 + 402}{836} \times 100 = 94.3\%$$

Table 6. Results (Confusion Matrix) of *DroidClone* using the holdout validation method.

	Malware	Benign
Malware	386	24
Benign	24	402

From the confusion matrix shown in Table 6, 24 samples were falsely detected as benign and also 24 were falsely detected as malware, and hence *DroidClone* was able to achieve a DR of 94.2% and an FPR of 5.6% with an accuracy of 94.3%. Almost the same results are achieved by the majority of the classifiers during 10-fold cross-validation, as shown in Table 7.

4.6. N-fold Cross Validation

We also use n-fold cross-validation to evaluate the performance of our technique. In n-fold cross-validation, the dataset is divided randomly into n equal size subsets. $n - 1$ sets are used for training, and the remaining set is used for testing. The cross-validation process is then repeated n times, with each of the n subsets used exactly once for validation. The purpose of this cross-validation is to produce very systematic and accurate testing results, to limit problems such as overfitting, and to give an insight on how the technique will generalize to an independent (unknown) dataset.

To evaluate the performance of our proposed technique, we trained multiple classifiers using the following machine learning algorithms: *BayesNetwork*: Based on the Bayesian theorem; *BFTree*: Best first decision tree; *NBTree*: Hybrid of decision tree and NaiveBayes classifiers; *RandomForest*: Forest of random trees; *RandomTree*: A decision tree built on a random subset of columns; and *REPTree*: Regression tree representative.

The results of 10-fold cross-validation with these classifiers are shown in Table 7.

Table 7. Results of *DroidClone* using 10-fold cross validation with six different classifiers.

Classifier	DR	FPR	Accuracy	AUC
BayesNetwork	94.2%	0.11	91.3%	0.975
RandomForest	93.1%	0.07	92.8%	0.969
RandomTree	93.1%	0.07	92.8%	0.917
NBTree	93.1%	0.10	91.2%	0.973
REPTree	90.6%	0.04	92.8%	0.969
BFTree	90.3%	0.04	92.9%	0.943

DroidClone successfully achieved DR $\geq$ 90.3% with all the classifiers. Highest DR reached by *DroidClone* is 94.2% with BayesNetwork. The highest AUC 97.5% reached is also with BayesNetwork. The range of FPR attained by *DroidClone* with the six classifiers is from 4% – 11%. The lowest FPR reached by *DroidClone* is with NBTree.

DroidClone reached similar results during n-fold cross-validation with four of the classifiers, as was attained with the holdout validation method in Section 4.5.

4.7. Resistance against Obfuscation

We also tested the resistance of *DroidClone* against various obfuscations. For this purpose, we used the 284 Android malware variants generated by DroidChameleon [37] as part of our dataset. The purpose of this experiment is to only check the resistance of *DroidClone* against various obfuscations and to make this a fair experiment, we took out those malware variants whose original sample was not detected as malware by *DroidClone*. Therefore, out of 284, we selected 273 malware variants for this experiment.

We trained *DroidClone* with the original 28 Android malware and 28 benign samples, and tested with the 273 Android malware variants. Out of the 273 malware variants *DroidClone* was able to successfully classify 247. The description of different Android bytecode obfuscations implemented to test the *DroidClone* and the results obtained are shown in Table 8.

Table 8. Description of different Android bytecode obfuscations implemented to test the resistance of *DroidClone* against various obfuscations.

Obfuscation	Description	Type of clone	DR
ICI	Manipulating call graph of the application.	Type 4	31/31 = 100%
IFI	Hiding function calls through indirection.	Type 4	30/30 = 100%
JNK	Inserting non-trivial junk code, including sophisticated sequences and branches that change the control flow of a program.	Type 3 & 4	15/28 = 53.6%
NOP	Inserting No operation instruction.	Type 1 & 3	32/32 = 100%
RDI	Removing debug information, such as source file names, local and parameter variable names, etc.	Type 2	31/31 = 100%
REO	Reordering the instructions and inserts non-trivial goto statements to preserve the execution sequence of the program. Inserting goto statements changes the control flow of a program.	Type 3 & 4	17/29 = 58.6%
REV	Reverse ordering the instructions and inserting trivial *goto* statements to preserve the execution sequence of the program. Hence changing the control flow of a program.	Type 3 & 4	29/30 = 96.7%
RNF	Renaming fields, such as packages, variables and parameters, etc.	Type 2	31/31 = 100%
RNM	Renaming methods.	Type 2	31/31 = 100%

The results shown in Table 8 demonstrate that *DroidClone* successfully provides resistance to all the trivial (Type 1, 2 & 3 clones) and some non-trivial obfuscations (Type 3 & 4 clones). Type 3 clones can be created by using trivial and non-trivial obfuscations. For example, it depends on the complexity of the reordering of the statements carried out while creating the clone.

This experiment also highlights the limitations of *DroidClone*. Whenever there is a significant change in the control flow of a program it becomes difficult for

DroidClone to generate a matching signature, and hence it fails to detect the similarity. The obfuscation technique JNK, for example, not only adds trivial but also some non-trivial junk code, such as sophisticated sequences and jumps. This makes a significant change in the control flow of a program and making it difficult to detect the malware program based on control flow patterns. To improve this shortcoming, in the future we will add other patterns, such as call flow, etc., to *DroidClone*.

4.8. Comparison with Other Researches

Table 9 shows a comparison of *DroidClone* with other malware detection techniques discussed in Section 2. The reasons for including these works are: (1) all of them are using the similarity/cloning of Android applications to detect malware; (2) have used machine learning to improve the performance and reported at least the DR obtained; (3) have used almost similar kind of Android applications for training and testing as used in this paper.

Table 9. Comparison of *DroidClone* with other malware detection techniques discussed in Section 2

Technique	DR	FPR	Dataset size
			Benign / malware
DroidClone [1]	94.2%	5.6%	2130 / 2050
SCSdroid [31]	97.9%	2%	100 / 49
DroidSim [41] [2]	96.6%	NA [3]	0 / 706
DomTree [5]	94.3%	4%	150 / 200
NiCad [13] [2]	94.5%	81% [4]	473 / 1170
DroidLegacy [17]	92.7%	21%	48 / 1052
AndroSimilar [20]	76.5%	2%	21,132 / 3309

[1] The results of *DroidClone* reported here are obtained with Naive-Bayes classifier.

[2] No n-fold cross validation was used to evaluate the technique.

[3] The technique was only evaluated with malware samples (no benign samples). Therefore, there is no FPR to report.

[4] For an equitable comparison, FPR of the *Type*-3 clone detector is reported here.

Out of the six techniques compared, *DroidClone* obtained a DR $\sim\geq$ to four of them. Only SCSdroid and DroidSim have a better DR. SCSdroid is tested with only 49 malware and 100 benign samples, whereas *DroidClone* is tested with a much greater number of samples. DroidSim is not tested with benign samples. SCSdroid, DomTree, and AndroSimilar achieved a lower FPR than *DroidClone*. Like SCSdroid, the number of samples used by DomTree is much lower than *DroidClone*. Although the FPR of AndroSimilar is low because of the SDHash technique [38] used, whose main criteria is to detect closely similar data objects, the ability to detect malware clones is much lower than *DroidClone*.

Like *DroidClone*, DomTree [5] also adapts TF-IDF [33] to improve feature selection. DomTree is the closest technique to *DroidClone*. Therefore, here we present some of the major differences between the two techniques: (1) DomTree does not provide native code malware analysis, and is not independent of the programming language of the code clone. (2) The similarity of two Android applications in *DroidClone* is based on the initial frequency and final weight of a MAIL block, whereas in DomTree it is only based on the presence of a dominant API module. (3) DomTree works at the dominant API level and is more suitable for finding coarse level clones. Whereas, *DroidClone* works at MAIL CFG block (statement) level and is more refined, and can detect clones of smaller size ≥ 3 statements.

Unlike the six works compared here *DroidClone*: uses an intermediate language MAIL to find Android malware clones; it is cross-platform, i.e., independent of the programming language of the Android code clone; can detect clones at a much-refined level; and achieves a DR better or comparable to others.

4.9. Malware Family Classification

The main purpose of the technique proposed in this paper is for binary classification, i.e., only two classes, *malware* and *benign*, and has been successfully used for this purpose. Because of the ability of DroidClone to detect clones we wanted to test if it can classify families of malware, i.e., grouping the samples into there respective families. DroidClone malware detection is based only on the *SimScore* of an Android application. It is difficult to find patterns specific to each family/class of malware just based on their *SimScore* values.

We carried out another experiment, with only selected malware families from the dataset, to test the potential of our proposed technique for predicting the family of a malware sample-based only on its *SimScore*. For this purpose, during training, we separated these classes into their respective *SimScore* groups. For example, $KMin$ was in the $288 - 298$ and $JSMSHider$ in the $41 - 47$ *SimScore* group. We have successfully used *DroidClone* for binary (two classes $\Rightarrow$ *benign* and *malware*) classification in Sections 4.5 and 4.6. Therefore, we used a variation of *one-versus-all* technique [11], which helps build a multiclass classifier from a binary classifier. A binary classifier was built for each class, that predicted the current class based on its *SimScore* group, i.e., if *SimScore* of a sample is in the current class group then it is predicted as positive and all the other samples are predicted as negative. This process was repeated for each class.

Our dataset for this experiment included a total of 197 malware samples from 11 different classes/families. All these samples have successfully been detected in Section 4.5 by *DroidClone* as malware. The results of these classifications into respective families are shown in Table 10. The results show that *DroidClone* was able to successfully separate (classify) most of the malware samples into their families based only on their *SimScore*, however, the results are not as accurate as general malware classification.

To get good results, in multiclass (in our case 11 classes) classification the input, e.g., the *SimScore* of the input sample, should belong to exactly one class out of the 11 classes and not two (*benign* and *malware*). It is just like mapping the *SimScore* values from 2 dimensions to 11 dimensions. The accuracy is lower because it is difficult to find such mapping successfully just based on the *SimScore* values. *One-versus-all* technique may not always work, as some classes were not predicted accurately by the single binary classifier build for each class.

Table 10. Prediction of some of the selected families of the malware samples.

Malware Family	DR
Appinventor	6/6 = 100%
YZHC	6/6 = 100%
DroidDream	3/3 = 100%
AnserverBot	71/72 = 98.6%
KMin	12/13 = 92.3%
DroidKungFu4	19/21 = 90.5%
DroidKungFu3	37/46 = 80.4%
DroidKungFu2	4/5 = 80%
ADRD	4/5 = 80%
DroidKungFu1	9/12 = 75%
JSMSHider	6/8 = 75%

In the future, we would like to improve this classification by taking into account and combining other features (such as permissions, API and system calls, etc.) with a strong correlation for predicting the family (specific class) of a malware program. We would also like to work on selecting and using some of the parameters, that have been used in this paper to calculate the similarity score (*SimScore*) of a program sample, as a set of features to improve family classification.

4.10. Limitations

DroidClone is based on static analysis of a sample, therefore it requires that the malicious code be available for static analysis. If an Android application contains compressed or encrypted code or requires to download malicious code upon initial execution (dynamic code loading), then the sample will not be correctly analysed by the system. If an Android application dynamically (while executing) link a third party library, which is not included in the application, then the library will not be processed by *DroidClone*.

DroidClone excels at detecting clone of a malware that has been previously known, but will only detect an unknown (zero-day) clone of a malware, if its control structure is similar, up to a threshold, to an existing malware sample in the training database.

If a clone of a malware is created, by obfuscating a statement in a basic block, in such a way that changes its MAIL pattern (e.g., control flow patterns) beyond a certain percentage (threshold), then *DroidClone* may not be able to detect such an obfuscated clone.

5. Conclusion

Android mobile platform is facing an attack of clones. In this paper, we propose *DroidClone* as a step towards detecting and stopping these clones in Android malware. *DroidClone* uses a new language MAIL to expose control flow patterns in a program, which helps in finding clones that are semantically similar up to a threshold. *DroidClone* is independent of the programming language of the code clones, as it builds cross-platform signatures. When evaluated with real malware and benign Android applications, *DroidClone* obtained a detection rate of 94.2% and false positive rate of 5.6%. *DroidClone*, when tested against various obfuscations, was able to successfully provide resistance against all the trivial and some non-trivial obfuscations.

The research carried out in this paper is just one step towards detecting and stopping Android malware clones. Some of the other works that need to be done in the future are: combining the static analysis performed in this paper with dynamic analysis to detect compressed, encrypted and dynamic loaded code clones; combining control flow patterns of MAIL with other structural features of Android applications to improve clone detection.

In the near future we will further improve the performance of *DroidClone* by combining the *SimScore* with other features, such as permissions, API and system calls, etc. We would also like to adapt the technique proposed in this paper for multiple (into families) classification by combining it with other such techniques. Different parameters were used to calculate the similarity score (*SimScore*) of a program sample. In the future, we would select some of these parameters as features to improve the family classification of *DroidClone*. When a significant change is made in the control flow of a program, it becomes difficult to detect the malware program based on control flow patterns. To improve this shortcoming in *DroidClone*, in the future we will add other patterns, such as call flow, etc.

References

1. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques, and Tools. Addison-Wesley, Inc. (2006)
2. Alam, S., Riley, R., Sogukpinar, I., Carkaci, N.: DroidClone: Detecting android malware variants by exposing code clones. In: DICTAP. pp. 79–84. IEEE (July 2016)

3. Alam, S., Horspool, R.N., Traore, I.: MAIL: Malware Analysis Intermediate Language - A Step Towards Automating and Optimizing Malware Detection. In: Security of Information and Networks. pp. 233–240. ACM SIGSAC (November 2013)
4. Alam, S., Qu, Z., Riley, R., Chen, Y., Rastogi, V.: DroidNative: Automating and optimizing detection of Android native code malware variants. Comput. Secur. 65(C), 230–246 (Mar 2017)
5. Alam, S., Yildirim, S., Hassan, M., Sogukpinar, I.: Mininng Dominance Tree of API Calls for Detecting Android Malware. In: 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). pp. 1–4. IEEE (2018)
6. Android-Development-Team: Dalvik Virtual Machine. `https://en.wikipedia.org/wiki/Dalvik_(software)` (2016)
7. Android-Development-Team: Android Runtime (ART). `http://en.wikipedia.org/wiki/Android_Runtime` (2021)
8. Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K.: Drebin: Effective and explainable detection of android malware in your pocket. In: NDSS (2014)
9. Baker, B.S.: On finding duplication and near-duplication in large software systems. In: Proceedings of 2nd Working Conference on Reverse Engineering. pp. 86–95. IEEE (1995)
10. Baxter, I.D., Yahin, A., Moura, L., Sant'Anna, M., Bier, L.: Clone detection using abstract syntax trees. In: Proceedings. International Conference on Software Maintenance (Cat. No. 98CB36272). pp. 368–377. IEEE (1998)
11. Bishop, C.M.: Pattern recognition and machine learning. springer (2006)
12. Cesare, S., Xiang, Y.: Wire–a formal intermediate language for binary analysis. In: TrustCom. pp. 515–524. IEEE (2012)
13. Chen, J., Alalfi, M.H., Dean, T.R., Zou, Y.: Detecting android malware using clone detection. Journal of Computer Science and Technology 30(5), 942–956 (2015)
14. Christodorescu, M., Jha, S., Seshia, S.A., Song, D., Bryant, R.E.: Semantics-Aware Malware Detection. In: Security and Privacy. pp. 32–46. SP '05, IEEE Computer Society (2005)
15. Collberg, C., Thomborson, C., Low, D.: A Taxonomy of Obfuscating Transformations. Tech. rep., University of Auckland (1997)
16. Cordy, J.R., Roy, C.K.: The nicad clone detector. In: Program Comprehension (ICPC), 2011 IEEE 19th International Conference on. pp. 219–220. IEEE (2011)
17. Deshotels, L., Notani, V., Lakhotia, A.: DroidLegacy: Automated Familial Classification of Android Malware. In: SIGPLAN. p. 3. ACM (2014)
18. Ducasse, S., Rieger, M., Demeyer, S.: A language independent approach for detecting duplicated code. In: Proceedings IEEE International Conference on Software Maintenance-1999 (ICSM'99).'Software Maintenance for Business Change'(Cat. No. 99CB36360). pp. 109–118. IEEE (1999)
19. Dullien, T., Porst, S.: Reil: A platform-independent intermediate representation of disassembled code for static code analysis. Proceeding of CanSecWest (2009)
20. Faruki, P., Laxmi, V., Bharmal, A., Gaur, M., Ganmoor, V.: Androsimilar: Robust signature for detecting variants of android malware. Journal of Information Security and Applications 22, 66–80 (2014)
21. Fawcett, T.: An Introduction to ROC Analysis. Pattern Recogn. Lett. 27, 861–874 (2006)
22. Funaro, M., Braga, D., Campi, A., Ghezzi, C.: A hybrid approach (syntactic and textual) to clone detection. In: Proceedings of the 4th International Workshop on Software Clones. pp. 79–80. ACM (2010)
23. Higo, Y., Yasushi, U., Nishino, M., Kusumoto, S.: Incremental code clone detection: A pdg-based approach. In: 2011 18th Working Conference on Reverse Engineering. pp. 3–12. IEEE (2011)

24. Hotta, K., Yang, J., Higo, Y., Kusumoto, S.: How accurate is coarse-grained clone detection?: Comparision with fine-grained detectors. Electronic Communications of the EASST 63 (2014)
25. Jiang, L., Misherghi, G., Su, Z., Glondu, S.: Deckard: Scalable and accurate tree-based detection of code clones. In: Proceedings of the 29th international conference on Software Engineering. pp. 96–105. IEEE Computer Society (2007)
26. Kalysch, A., Milisterfer, O., Protsenko, M., Müller, T.: Tackling androids native library malware with robust, efficient and accurate similarity measures. In: Proceedings of the 13th International Conference on Availability, Reliability and Security. p. 58. ACM (2018)
27. Komondoor, R., Horwitz, S.: Using slicing to identify duplication in source code. In: International static analysis symposium. pp. 40–56. Springer (2001)
28. Koschke, R., Falke, R., Frenzel, P.: Clone detection using abstract syntax suffix trees. In: 2006 13th Working Conference on Reverse Engineering. pp. 253–262. IEEE (2006)
29. Krinke, J.: Identifying similar code with program dependence graphs. In: Proceedings Eighth Working Conference on Reverse Engineering. pp. 301–309. IEEE (2001)
30. Li, Z., Lu, S., Myagmar, S., Zhou, Y.: Cp-miner: Finding copy-paste and related bugs in large-scale software code. IEEE Transactions on software Engineering 32(3), 176–192 (2006)
31. Lin, Y.D., Lai, Y.C., Chen, C.H., Tsai, H.C.: Identifying android malicious repackaged applications by thread-grained system call sequences. Computers & Security 39, 340–350 (2013)
32. Liu, C., Chen, C., Han, J., Yu, P.S.: Gplag: detection of software plagiarism by program dependence graph analysis. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 872–881. ACM (2006)
33. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
34. McAfee, C.: McAfee mobile threat report Q1 (© McAfee Corporation 2017)
35. Murakami, H., Hotta, K., Higo, Y., Igaki, H., Kusumoto, S.: Folding repeated instructions for improving token-based code clone detection. In: 2012 IEEE 12th International Working Conference on Source Code Analysis and Manipulation. pp. 64–73. IEEE (2012)
36. Parkour, M.: Mobile Malware Dump. http://contagiominidump.blogspot.com (2021)
37. Rastogi, V., Chen, Y., Jiang, X.: DroidChameleon: Evaluating Android Anti-malware Against Transformation Attacks. In: ASIA CCS. pp. 329–334. ASIA CCS '13, ACM (2013)
38. Roussev, V.: Data Fingerprinting with Similarity Digests. In: Chow, K.P., Shenoi, S. (eds.) Advances in Digital Forensics VI, pp. 207–226. Springer (2010)
39. Roy, C.K., Cordy, J.R.: A survey on software clone detection research. Tech. rep., 541, Queen's University at Kingston, ON, Canada (2007)
40. Song, D., Brumley, D., Yin, H., Caballero, J., Jager, I., Kang, M.G., Liang, Z., Newsome, J., Poosankam, P., Saxena, P.: Bitblaze: A new approach to computer security via binary analysis. In: Information systems security, pp. 1–25. Springer (2008)
41. Sun, X., Zhongyang, Y., Xin, Z., Mao, B., Xie, L.: Detecting Code Reuse in Android Applications Using Component-Based Control Flow Graph. In: ICT, pp. 142–155. Springer (2014)
42. Symantec, C.: Symantec security threat report (© Symantec Corporation 2017)
43. Symantec, C.: Symantec security threat report (© Symantec Corporation 2018)

44. Team, A.D.: Android Dalvik Virtual Machine Opcodes. http://developer.android.com/reference/dalvik/bytecode/Opcodes.html (2021)
45. Wahler, V., Seipel, D., Wolff, J., Fischer, G.: Clone detection in source code by frequent itemset techniques. In: Source Code Analysis and Manipulation, Fourth IEEE International Workshop on. pp. 128–135. IEEE (2004)
46. Yuan, Y., Guo, Y.: Boreas: an accurate and scalable token-based approach to code clone detection. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. pp. 286–289. ACM (2012)
47. Zhou, Y., Jiang, X.: Dissecting android malware: Characterization and evolution. In: Security and Privacy. pp. 95–109. IEEE (2012)

Shahid Alam is currently working as an assistant professor in the department of Computer Engineering at Adana Alparsalan Turkes Science and Technology University, Adana, Turkey. He recevied his PhD in Computer Science from University of Victoria, Canada in 2014. His research interests include software engineering, programming languages, and cyber security.

Ibrahim Sogukpinar received his PhD degree in Computer and Control Engineering from Technical University of Istanbul in 1995. Currently he is a Professor in the Department of Computer Engineering at Gebze Technical University, Gebze, Turkey. His main research areas are information security, computer networks, applications of information systems, and computer vision.

Received: March 30, 2020; Accepted: August 10, 2020.

Computer Science and Information Systems 18(1):93–114 https://doi.org/10.2298/CSIS200608039H

TrustRec: An Effective Approach to Exploit Implicit Trust and Distrust Relationships along with Explicit ones for Accurate Recommendations*

Masoud Reyhani Hamedani, Irfan Ali, Jiwon Hong, and Sang-Wook Kim*

Department of Computer and Software, Hanyang University,
Seoul, Korea, 04763
{masoud,irfan.ali,nowiz,wook}@hanyang.ac.kr

Abstract. Trust-aware recommendation approaches are widely used to mitigate the cold-start problem in recommender systems by utilizing trust networks. In this paper, we *point out* the problems of existing trust-aware recommendation approaches as follows: (P1) exploiting *sparse* explicit trust and distrust relationships; (P2) considering a *misleading* assumption that a user pair having a trust/distrust relationship *certainly* has a similar/dissimilar preference in practice; (P3) employing the *transitivity* of *distrust* relationships. Then, we propose *TrustRec*, a novel approach based on the matrix factorization that provides an effective *solution* to each of the aforementioned problems and incorporates all of them in a *single* matrix factorization model. Furthermore, TrustRec exploits *only* top-*k* most similar trustees and dissimilar distrustees of each user to improve both the computational cost and accuracy. The results of our extensive experiments demonstrate that TructRec outperforms existing approaches in terms of *both* effectiveness and efficiency.

Keywords: Recommender Systems, Collaborating Filtering, Trust Network, Matrix Factorization, Distrust Intransitivity

1. Introduction

Although collaborative filtering (CF) is a well-known technique in recommender systems [1, 2, 6], it performs *poorly* for users who have rated a *very small* number of items; this problem is called as a *cold-start user* problem. In the literature, several CF approaches have been proposed to solve this problem by exploiting the additional information such as social information [4–6, 8, 11, 16–20, 23–25], demographic information [21], crowd source information [14], uninteresting items [10], and location information [28] along with the ratings information. In particular, the *trust-aware* recommendation approaches [4, 5, 11, 16–19, 24, 25, 27] exploit a *trust network*, which is kind of social information.

In a trust network, users can establish two types of relationships: *trust* and *distrust*. Trust/distrust is a unidirectional relationship between two users, indicating that one *agrees/ disagrees* on the opinion of the other on some items where *explicit* trust/distrust relationships are established by users *themselves*, while the *implicit* ones are *inferred* based on

* The initial idea has been presented with some preliminary experimental results in ACM SAC 2017 [3]. The current paper is an extended version that contains some new ideas, formulations, and more extensive experimental results.
Corresponding author: Sang-Wook Kim (wook@hanyang.ac.kr)

some *evidence* such as the degree of similarity in users' preferences [16] or the transitivity of other trust/distrust relationships [12]. A user who trusts someone is called a *trustor* and a user who is being trusted by someone else is called a *trustee*; a user who distrusts someone is called a *distruster* and a user who is being distrusted by someone else is called a *distrustee*. Fig. 1 shows a sample trust network where thick and dotted thick arrows indicate explicit trust and distrust relationships, respectively (i.e., A is an explicit trustor of B while B is an explicit trustee of A). The thin arrow shows an *implicit* trust relationship between two users inferred by *transitivity*; F is an implicit trustor of B and B is an implicit trustee of F since F trusts A and A trusts B.

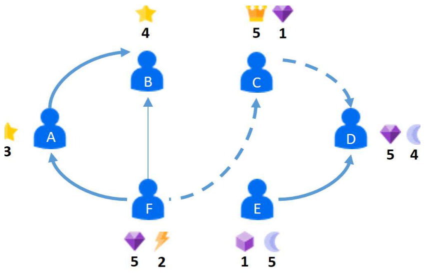


Fig. 1. A sample trust network

Some of the existing trust-aware recommendation approaches exploit *only* trust relationships [11,12,17,19] and some approaches exploit *both* trust and distrust relationships; the latter approaches achieve *better* accuracy than the former ones [4,5,16,18,24,25]. In this paper, we first point out the problems in the existing approaches as follows:

Problem 1: as shown in Table 1, which summarizes the statistics of three real-world datasets, *not* only ratings (i.e., R-density) but also explicit trust and distrust relationships (i.e., T/D-density) are *sparse*. Therefore, the approaches exploiting *only* explicit relationships may *fail* to infer the true preferences of users, in particular, in the case of cold-start users. To solve this problem, PushTrust [4], JNMF-SG [24], and Impl [16] infer implicit trust and distrust relationships and exploit them along with the explicit ones. Impl exploits implicit and explicit trust/distrust relationships *separately* in *two* different models, thus *unable* to solve the sparsity of explicit relationships. Although PushTrust and JNMF-SG exploit both implicit and explicit relationships in a same model, they still *suffer* from Problem 2, which will be explained below.

Problem 2: existing approaches assume that a user pair having an explicit trust/distrust relationship has a similar/dissimilar preference. However, they *neglect* the fact that some of the user pairs having explicit trust/distrust relationships may have dissimilar/similar preferences *in practice* as we show it in Section 3.2; likewise, the same situation may exist for implicit relationships. The similarity in preferences between users having explicit or implicit trust/distrust relationships need to be examined *before* being used in inferring users' preferences.

Problem 3: It has been shown that distrust relationships are intransitive, while trust relationships are transitive [7]. However, MF-TD [5], PushTrust [4], Impl [16], JNMF-SG [24], and RecSSN [25] consider distrust relationships as transitive ones, which may

Table 1. Statistics of some datasets

	Epinions	FilmTrust	Ciao
# Users	132,492	1,508	7,375
# Items	755,760	2,071	99,746
# Ratings	13,668,320	35,497	278,483
# R-density	0.015%	1.136%	0.037%
# Trusts	717,667	1,853	111,781
# Distrusts	123,705	0	0
# T/D-density	0.005%	0.081%	0.205%

cause *inaccurate inference* of a user's preferences. We clarify this problem with the following example. In Fig. 1, user *F* explicitly distrusts user *C* who explicitly distrusts user *D*. By transitivity of distrust relationships, user *F* should *implicitly* distrust user *D*, which means they have dissimilar preferences; however, as shown in Fig. 1, their ratings on the common item (i.e., diamond) are both 5 that indicates they have likely similar preferences.

In addition to these problems, to address the sparsity of explicit trust/distrust relationships, PushTrust [4] and JNMF-SG [24] exploit *all* implicit and explicit relationships of *each* user, which requires a *significantly high* computational cost (e.g., the complexity of PushTrust is $\mathcal{O}(n^2)$ where *n* denotes the number of users). Furthermore, as we show in Section 4, this problem also leads to inaccurate recommendations.

In this paper, we propose a novel trust-aware recommendation approach called *TrustRec*, which provides *effective solutions* to the three aforementioned problems as *inferring implicit relationships (IIR)*, *confirming trustees and distrustees (CTD)*, and *employing distrust's intransitivity (EDI)*, respectively. The IIR solution infers implicit trust/distrust relationships between a pair of users by computing the similarity between their rating, which increases the *density* of trust and distrust relationships a lot, thereby enabling us to infer users' preferences effectively. The CTD solution *confirms* the degree of similarity/dissimilarity of user pairs having explicit trust/distrust relationships in terms of their ratings. This allows us to infer the preferences of the target user by exploiting *only* her *real* similar trustees and dissimilar distrustees. The EDI solution considers the *intransitivity* of distrust relationships in the trust network. Also, TrustRec selects top-*k most* similar trustees and top-*k most* dissimilar distrustees of the target user from her *all* explicit as well as implicit trustees and distrustees to infer her preferences, which is expected to *not* only reduce the computational cost *but also* to improve the accuracy; we refer to it as the *selecting top-k relationships (STR)* solution. Furthermore, TrustRec utilizes a *novel* matrix factorization model that incorporates *all* the proposed solutions together into a *single* model. We evaluate our TrustRec and each of the four solutions by conducting extensive experiments with a real-world dataset. Our experimental results demonstrate that 1) *each* of our four solutions is *effective* in making accurate recommendations; 2) our TrustRec significantly outperforms existing approaches in terms of *both* effectiveness and efficiency. The contributions of this paper are summarized as follows:

- We point out the problems of existing trust-aware recommendation approaches.
- We propose an effective solution to each of the problems.
- We propose a novel matrix factorization model that incorporates *all* the aforementioned solutions together in a *single* model.

- We conduct extensive experiments with a real-world dataset, evaluating the effectiveness and efficiency of our TrustRec in comparison with existing approaches.

The rest of the paper is organized as follows. We discuss existing trust-aware recommendation approaches and point out their problems in Section 2. In Section 3, we present our solutions in detail and explain how to integrate them under a single matrix factorization model. In Section 4, we evaluate the effectiveness of our solutions in recommendation and also evaluate the effectiveness and efficiency of our TrustRec in comparison with existing approaches. In Section 5, we conclude our paper.

2. Background and Related Work

In this section, we explain CF by focusing on matrix factorization since our approach is built upon it. We also provide a brief overview of existing trust-aware recommendation approaches and point out their problems.

CF is a widely used technique in recommender systems because it is simple, effective, and efficient [2, 6, 22, 27]. Matrix factorization-based CF approaches have lately gained popularity since they scale *linearly* with the numbers of users and items [13]. Suppose $R \in \mathbb{R}^{n \times m}$ denotes a *user-item rating* matrix where each entry R_{ui} represents a rating given by user *u* on item *i*; *n* and *m* represent the total numbers of users and items, respectively; matrix factorization-based CF approaches try to obtain a latent *user* feature matrix *U* and a latent *item* feature matrix *V* such that $U \in \mathbb{R}^{n \times f}$ and $V \in \mathbb{R}^{f \times m}$ can effectively recuperate the rating matrix $R \cong \hat{R} = UV$ where *f* is the number of latent features. Matrix factorization tries to obtain *U* and *V* by minimizing the following objective function:

$$\mathcal{F}(U,V) = \sum_{u=1}^{n} \sum_{i=1}^{m} \left(r_{ui} - U_u^T V_i\right)^2 + \lambda_U \parallel U \parallel_F^2 + \lambda_V \parallel V \parallel_F^2 . \qquad (1)$$

where $\|.\|_F^2$ denotes the Frobenius norm. We refer to the first part of Eq. 1 as the *factorization part*, which minimizes the difference between real users' ratings and their corresponding predicted ratings. We refer to the second and third parts as the *regularization part*; λ_U and λ_V are regularization parameters to avoid overfitting [2, 13].

SoRec [19], RSTE [17], SocialMF [12], and TrustMF [27] exploit *only* trust relationships. SoRec [19] is based on a probabilistic graphical model that factorizes a rating matrix and a trust matrix simultaneously by sharing a common user latent feature matrix. RSTE [17] is a linear combination of a traditional matrix factorization-based CF method and a trust-aware recommendation approach. However, both SoRec and RSTE exploit trust relationships in the factorization part of the matrix factorization model. In contrast, SocialMF [12] exploits trust relationships in the regularization part of the matrix factorization model and employs the transitivity of trust relationships. TrustMF [27] maps users into two low dimensional spaces, truster space and trustee space, by factorizing the trust relationship matrix and proposes a matrix factorization model that incorporates the truster model along with the trustee model to address the cold-start user problem.

Later, it has been observed that distrust relationships are also important as trust ones since exploiting them enables to make more accurate recommendations [4,5,16,18,24,25]. The earlier attempts include an approach that exploits trust and distrust relationships *separately* into *two* different models [18] where it is shown that the matrix factorization model

Table 2. Comparison of TrustRec with existing approaches

	MF-TD	PushTrust	JNMF-SG	RecSSN	TrustRec
Problem 1	✗	✓	✓	✗	✓
Problem 2	✗	✗	✗	✗	✓
Problem 3	✗	✗	✗	✗	✓

exploiting only trust relationships performs better than the one that exploiting only distrust relationships. The true effectiveness of exploiting distrust relationships is shown by MF-TD [5], PushTrust [4], JNMF-SG [24], and RecSSN [25] where they exploit both trust and distrust relationships together in a *single* model. Let us compare the latter approaches with our TrustRec in regarding to the three problems explained in Section 1 as follows; Table 2 summarizes the result where a '✗' mark denotes that the approach does not solve the problem and a '✓' mark denotes that the approach solves the problem.

(1) PushTrust and JNMF-SG solve Problem 1. PushTrust infers implicit trust relationships between a user and *all* those users who have no any explicit relationship with her. The explicit distrustees of a user are used to filter out dissimilar users from her explicit and implicit trustees based on their similarity score, which is computed by exploiting their latent feature vectors. JNMF-SG infers implicit trust and distrust relationships by clustering explicit ones using the ratio-cut spectral clustering technique [15] where users in the same cluster are regarded to have implicit trust relationships, while users in different clusters are regarded to have implicit distrust relationships. On the contrary, both MF-TD and RecSSN suffer from Problem 1 since they exploit *only* explicit relationships. (2) MF-TD, PushTrust, JNMF-SG, and RecSSN do not consider the degree of similarity/dissimilarity between users having trust/distrust relationships, thus *all* suffering from Problem 2. (3) MF-TD, JNMF-SG, and RecSSN exploit distrust relationships in the regularization part of a matrix factorization model, thereby employing the transitivity of distrust relationships; they *all* suffer from Problem 3. Although PushTrust exploits only trust relationships in the regularization part, the degree of similarity of users having trust relationships is not confirmed; some of those users may have dissimilar preferences. Therefore, PushTrust may *indirectly* exploit some distrust relationships in the regularization part, thereby suffering from Problem 3. As shown in Table 2, on contrary to the existing approaches, TrustRec addresses *all* the three aforementioned problems by providing an effective solution to each of them. In addition, both PushTrust and JNMF-SG suffer from high computational cost since they exploit all the explicit as well as implicit trust and distrust relationships for every user, which increases their computational cost significantly and also could introduce noise in inferring users' preferences.

3. Proposed Approach

In this section, we present our proposed approach, TrustRec, and each of the IIR, CTD, STR, and EDI solutions in detail. Finally, we present our matrix factorization model.

As shown in Fig. 2, TrustRec provides an effective solution to each of the problems described in Sections 1 and 2. TrustRec infers implicit relationships between users to solve Problem 1, considers the degree of similarity/dissimilarity of users having trust/distrust

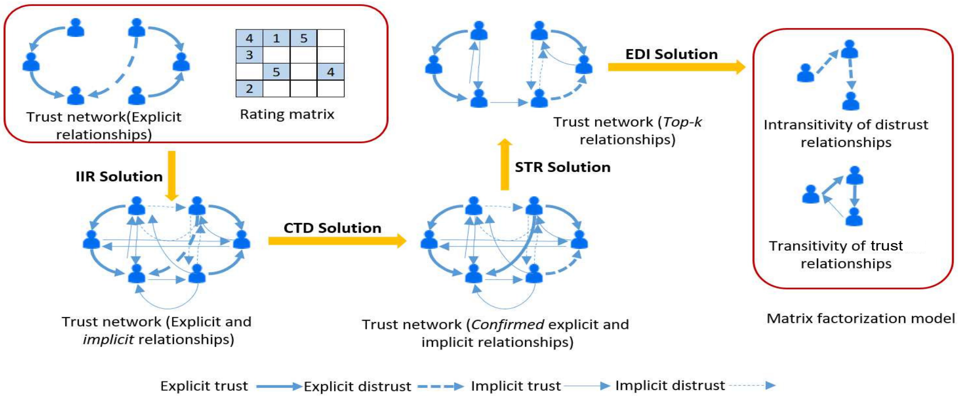

4 1 5
3
5 4
2
Trust network(Explicit relationships)
Rating matrix
IIR Solution
Trust network (Explicit and *implicit* relationships)
CTD Solution
Trust network (*Confirmed* explicit and implicit relationships)
STR Solution
Trust network (*Top-k* relationships)
EDI Solution
Intransitivity of distrust relationships
Transitivity of trust relationships
Matrix factorization model
Explicit trust
Explicit distrust
Implicit trust
Implicit distrust

Fig. 2. Overview of TrustRec

Table 3. Symbols and their meanings in TrustRec

Symbol	Meaning
$\mathbb{U} = \{u_1, ..., u_n\}$	Set of *n* users
$\mathbb{I} = \{i_1, ..., i_m\}$	Set of *m* items
$R \in \mathbb{R}^{n \times m}$	Sparse rating matrix
f	# of latent features in matrix factorization
$U \in \mathbb{R}^{n \times f}$	Latent features matrix for users
$V \in \mathbb{R}^{f \times m}$	Latent features matrix for items
$S \in \mathbb{R}^{n \times n}$	Similarity matrix for all user pairs
$T \in \{-1, +1\}^{n \times n}$	Explicit relationships matrix
$G \in \{-1, +1\}^{n \times n}$	Inferred implicit relationships matrix
X_u	Row *u* in matrix *X*
$\overline{X}_u$	Average of all values in X_u
$X_{u,v}$	Value at row *u* and column *v* in matrix *X*
$X\,(u)$	Set of column indexes for non-null values in X_u
$X_+(u)$	Set of trustees of user *u* in matrix *X*
$X_-(u)$	Set distrustees of user *u* in matrix *X*
$S_+(u,k)$	Set of top-*k* trustees of user *u*
$S_-(u,k)$	Set of top-*k* distrustees of user *u*

relationships to solve Problem 2, and employs a novel matrix factorization model incorporating distrust relationships into the factorization part which makes it exploit the intransitivity of distrust relationships to solve Problem 3. Furthermore, TrustRec selects only the top-*k* relationships of a user regardless of the relationship type (i.e., explicit or implicit) to improve both efficiency and effectiveness. Given a sparse rating matrix *R* and a trust network *T*, *the problem that our TrustRec tries to solve is to infer accurately the missing values in R*. Table 3 lists all the notations and their meanings used in this paper.

3.1. Inferring Implicit Relationships (IIR)

One of possible solutions to the sparsity problem of explicit relationships is to exploit additional information such as implicit relationships, which is performed by PushTrust, Impl, and JNMF-SG; however, all these approaches suffer from Problem 2 as discussed before. We also solve the sparsity problem by applying the same solution; however, to avoid Problem 2, we infer implicit relationships as follows. We compute the similarity scores between *all* possible users based on their ratings; if the similarity score between a pair of users is *higher/lower* than a given threshold and an explicit trust/distrust relationship does *not* exist between them, we add an implicit trust/distrust relationship between them. We note that Impl also infers implicit relationships based on the similarity score of the users' ratings. However, it selects *topmost* similar/dissimilar users to a user as her implicit trustees/distrustees. Thus, it is likely that some users having negative/positive similarity scores (i.e., having dissimilar/similar preferences) with the user could be selected as her implicit trustees/distrustees; Impl may be *unable* to accurately infer the implicit trustees/distrustees of a user. On the contrary, our IIR solution infers implicit trustees/distrustees of a user *only if* the similarity score between them is greater/smaller than a given threshold, thereby leading to the inference of implicit trustees/distrustees of a user more effectively.

We utilize the Pearson correlation coefficient (PCC) as a similarity measure to compute the similarity between users u and v based on their ratings as follows; we utilized PCC since it is the most commonly used similarity measure in recommender systems [2, 26].

$$P(u,v) = \frac{\sum_{i\in R(u)\cap R(v)} \left(R_{u,i}-\overline{R_u}\right)\cdot\left(R_{v,i}-\overline{R_v}\right)}{\sqrt{\sum_{i\in R(u)\cap R(v)} \left(R_{u,i}-\overline{R_u}\right)^2}\cdot\sqrt{\sum_{i\in R(u)\cap R(v)} \left(R_{v,i}-\overline{R_v}\right)^2}}\,. \tag{2}$$

By following [2] and [9], to make the similarity score more reliable, we compute the similarity between two users *only if* they have at least *h* co-rated items; if the number of co-rated items is less than *h*, their implicit relationship is *not* inferred. Let $G \in \{-1, +1\}^{n\times n}$ be a matrix that represents the inferred *implicit* relationships between users where $G_{u,v} = 1$ and $G_{u,v} = -1$ denote implicit trust and distrust relationships between users *u* and *v*, respectively. We infer the implicit relationships between *u* and *v* as follows:

$$G_{u,v} = \begin{cases} 1, & P(u,v) > \sigma_T \text{ and } T_{u,v} = \textit{null}\,. \\ -1, & P(u,v) < \sigma_D \text{ and } T_{u,v} = \textit{null}\,. \end{cases} \tag{3}$$

where σ_T/σ_D denotes a threshold to ensure that the similarity/dissimilarity between *u* and *v* is sufficient to have an implicit trust/distrust relationship and $T_{u,v} = null$ indicates that the explicit relationship does *not* exist between *u* and *v*.

3.2. Confirming Trustees and Distrustees (CTD)

In the Epinions dataset, we analyzed the similarity scores between all user pairs having explicit relationships, presented in Table 4. Surprisingly, $45,208$ (i.e., 6.3%) out of $717,667$ user pairs having an explicit trust relationship have *negative* similarity scores, which indicates that these users actually have *dissimilar* preferences. Also, $19,498$ (i.e.,

15.76%) out of $123,705$ user pairs having an explicit distrust relationship have *positive* similarity scores, which indicates that these users actually may have *similar* preferences. These results show that, even if users *explicitly* trust/distrust each other, they may have dissimilar/similar preferences in practice. Therefore, it seems *necessary* to *confirm* the degree of similarity/dissimilarity of users having explicit trust/distrust relationships.

Table 4. Analysis on explicit relationships in the Epinions dataset

# Trust relationships	# user-pairs with similarity scores	# user-pairs with negative similarity scores
717,66	523,983	45,208
# Distrust relationships	# user-pairs with similarity scores	# user-pairs with positive similarity scores
123,705	64,175	19,498

In our TrustRec, the similarity/dissimilarity of users having implicit trust/distrust relationships are confirmed *by default* since implicit relationships are inferred based on the similarity score between users' ratings. Therefore, we *only* need to confirm the degree of similarity/dissimilarity of users having explicit trust/distrust relationships by utilizing the similarity scores, which are *already* computed in the IIR solution: for a user pair having an explicit relationship, if their similarity score is greater than σ_T, we consider it as a *trust* relationship; if their similarity score is less than σ_D, we consider it as a *distrust* relationship. To incorporate this solution in our approach, we modify matrix *T* as follows:

$$T_{u,v} = \begin{cases} 1, & P(u,v) > \sigma_T \text{ and } T_{u,v} \neq \textit{null}\,. \\ -1, & P(u,v) < \sigma_D \text{ and } T_{u,v} \neq \textit{null}\,. \end{cases} \tag{4}$$

where $T_{u,v} = 1$ and $T_{u,v} = -1$ show explicit trust and distrust relationships between users *u* and *v,* respectively.

3.3. Selecting Top-*k* Relationships (STR)

PushTrust [4] and JNMF-SG [24] exploit *all* implicit and explicit relationships of every user, which increases the *computation cost* significantly and also may introduce *noise* in inferring users' preferences [11]. To solve this problem, we select *only* the top-*k* trust and distrust relationships of each user *regardless* of their relationship type (i.e., implicit or explicit). The top-*k* trust and distrust relationships of a target user are those relationships with *k* trustees and *k* distrustees who have the highest and lowest similarity scores with her, respectively. However, selecting top-*k* relationships is *challenging* since if two users involved in an explicit relationship have *less* than *h* co-rated items, their similarity score *cannot* be computed as explained in Section 3.1. Ignoring those user pairs may make the sparsity problem of explicit relationships more serious. Hereafter, we refer to a pair of users who have an explicit relationship with *less* than *h* co-rated items as an *unconfirmed user pair* and refer to the user in that pair as an *unconfirmed* trustor, distruster, trustee, or distrsutee depending on her role in the relationship. Also, we refer to a pair of users who have an explicit relationship with *more* than *h* co-rated items as a *confirmed user pair* and refer to the user in that pair as a *confirmed* trustor, distruster, trustee, or distrsutee. The similarity score between users in a confirmed user pair is computed easily by Eq. (2). We estimate the similarity score for unconfirmed user pairs as follows.

Let *u* and *v* be an unconfirmed user pair where *u* is an unconfirmed trustor and *v* is an unconfirmed trustee. We regard that the preference of *v* is similar to *u* as much as those of *u*'s confirmed trustees; in other words, the degree of similarity between a user and her confirmed trustees are exploited to estimate the similarity score between the user and her unconfirmed trustee. We consider *two* possible situations as follows: first, *u* has some confirmed trustees (i.e., $S\left(u\right)\bigcap T_+\left(u\right) \neq \emptyset$); second, *u* has *no* confirmed trustees and her explicit trustees are *all* unconfirmed ones (i.e.,$S\left(u\right)\bigcap T_+\left(u\right) = \emptyset$). Note that, according to Table 3, $S\left(u\right)\bigcap T_+\left(u\right)$ denotes the confirmed trustees of *u*. In the first situation, we estimate the similarity score between *u* and *v*, $S(u,v)$, as follows:

$$S(u,v) = \frac{\sum_{z\in S(u)\bigcap T_+(u)} P(u,z)}{\left|S\left(u\right)\bigcap T_+\left(u\right)\right|}\,. \tag{5}$$

where $P(u,z)$ is the similarity score between users *u* and *z* computed by Eq. (2); we regard the *average* of similarity scores between *u* and all her confirmed trustees as the estimated similarity score between *u* and *v*.

In the second situation, since the number of confirmed trustees of u is equal to zero, we estimate the similarity between *u* and *v* as follows:

$$S(u,v) = \frac{\sum_{y\in\mathbb{U}}\sum_{z\in S(y)\bigcap T_+(y)} P(y,z)}{\sum_{y\in\mathbb{U}}\left|S\left(y\right)\bigcap T_+\left(y\right)\right|}\,. \tag{6}$$

where we regard the *average* of the similarity scores between two users in *all* the existing confirmed user pairs having explicit trust relationship as the estimated similarity score between *u* and *v*.

Let *u* and *w* be an unconfirmed user pair where *u* is an unconfirmed distruster and *w* is an unconfirmed distrustee. We regard that the preference of *w* is dissimilar to *u* as much as those of *u*'s confirmed distrustees; in other words, the degree of dissimilarity between a user and her confirmed distrustees are exploited to estimate the similarity score between the user and her unconfirmed distrustees. We consider *two* possible situations as follows: first, *u* has some confirmed distrustees (i.e., $S\left(u\right)\bigcap T_-\left(u\right) \neq \emptyset$); second, *u* has *no* confirmed distrustees and her explicit distrustees are *all* unconfirmed ones (i.e., $S\left(u\right)\bigcap T_-\left(u\right) = \emptyset$). Note that, according to Table 3, $S\left(u\right)\bigcap T_-\left(u\right)$ denotes the confirmed distrustees of user *u*. In the first situation, $S(u,w)$ is estimated as follows:

$$S(u,w) = \frac{\sum_{z\in S(u)\bigcap T_-(u)} P(u,z)}{\left|S\left(u\right)\bigcap T_-\left(u\right)\right|}\,. \tag{7}$$

where we regard the *average* of similarity scores between *u* and *all* her confirmed distrustees as the estimated similarity score between *u* and *w*.

In the second situation (i.e., $S\left(u\right)\bigcap T_-\left(u\right) = \emptyset$), we estimate the similarity between *u* and *w* as follows:

$$S(u,w) = \frac{\sum_{y\in\mathbb{U}}\sum_{z\in S(y)\bigcap T_-(y)} P(y,z)}{\sum_{y\in\mathbb{U}}\left|S\left(y\right)\bigcap T_-\left(y\right)\right|}\,. \tag{8}$$

where we regard the *average* of the similarity scores between two users in *all* the existing confirmed user pairs having explicit distrust relationship as the estimated similarity score between *u* and *w*.

Once a similarity score is assigned to every user pair having a relationship, we select the top-*k* trust/distrust relationships of a user according to the similarity scores between the user and her trustees/distrustees sorted in *descending/ascending* order; we select *equal numbers* of trust and distrust relationships of a user by following previous work [4,5,24,25] where it has been shown that exploiting equal numbers of trust and distrust relationships provides the best accuracy. As shown in Section 4, applying the STR solution reduces the computational cost. In addition, it improves the accuracy since it somehow *refines* the trustees/distrustees to a target user and only exploits those trustees/distrustees who are highly similar/dissimilar to her to infer her preferences.

3.4. Employing Distrust's Intransitivity (EDI)

As discussed earlier, employing the transitivity of distrust relationships may cause to treat implicit trustees of a user as her implicit distrustees in a wrong way. As a result, a user's preferences may be inferred incorrectly, thereby leading to inaccurate recommendations. Moreover, it has been shown that trust is transitive while distrust is intransitive [7]. Therefore, it should be a natural choice to employ the transitivity of trust and intransitivity of distrust in recommendation. To employ the intransitivity of distrust relationships, we exploit them in the factorization part of Eq. (1) as follows:

$$\min_{U,V}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(R_{i,j}-U_i^TV_j-\lambda_D\left(\frac{\sum_{v\in S_-(u,k)}U_v}{|S_-(u,k)|}\right)V_j\right)^2+\lambda_U\|U_i\|_F^2+\lambda_V\|V_j\|_F^2\,. \quad (9)$$

where λ_D as a parameter controls the importance of distrustees' latent feature vectors.

By exploiting distrust relationships in the factorization part, latent feature vectors of distrustees are utilized *only* to predict the target user's original ratings, then those predicted ratings are used to infer users' latent feature vectors. Since the distrustees of a target user can *directly* affect only her original ratings rather than her latent feature vector, our proposed approach *employs* the intransitivity of distrust relationships successfully.

3.5. Unified Matrix Factorization Model

To incorporate the four aforementioned solutions together, we propose a novel matrix factorization model as follows:

$$\min_{U,V}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(R_{i,j}-U_i^TV_j-\lambda_D\left(\frac{\sum_{v\in S_-(u,k)}U_v}{|S_-(u,k)|}\right)V_j\right)^2+$$
$$\lambda_U\|U_i\|_F^2+\lambda_V\|V_j\|_F^2+\lambda_T\|U_i-\left(\frac{\sum_{v\in S_+(u,k)}U_v}{|S_+(u,k)|}\right)\|_F^2\,. \quad (10)$$

where λ_T denotes a parameter for controlling the importance of trustees' latent feature vectors. Following SocialMF [12], we add a term into the regularization part of Eq. (9), which minimizes the difference between the latent feature vector of a user and the average of her trustees' latent feature vectors. As a result, the latent feature vector of a user is learned by referring to her trustees, and the latent feature vectors of her trustees are learned by referring to their trustees recursively, thereby making our approach to employ the transitivity of trust relationships.

Moreover, as explained before, our model also employs the intransitivity of distrust relationships. We can find a local minimum of Eq. (10) by performing *gradient descent* on U_u and V_i for all users *u* and all items *i* as follows:

$$\begin{aligned}
\frac{\partial \mathcal{F}}{\partial U_u} =& 2\sum_{u=1}^{n}\left(R_{u,i}-\left(U_u-\lambda_D\left(\frac{\sum_{v\in S_-(u,k)} U_v}{|S_-(u,k)|}\right)\right)^T V_i\right)^2 \\
&\cdot\left(1-\lambda_D\left(\frac{\sum_{v\in S_-(u,k)} U_v}{|S_-(u,k)|}\right)^T V_i\right) \\
&+2\lambda_U U_u + 2\lambda_T\left(U_u-\frac{\sum_{v\in S_+(u,k)} U_v}{|S_+(u,k)|}\right) \\
&-\lambda_T \sum_{\{v|u\in S_+(v,k)\}}\left(U_v-\frac{\sum_{w\in S_+(v,k)} U_w}{|S_+(w,k)|}\right) \\
\frac{\partial \mathcal{F}}{\partial V_i} =& 2\sum_{i=1}^{m}\left(R_{u,i}-\left(U_u-\lambda_D\left(\frac{\sum_{v\in S_-(u,k)} U_v}{|S_-(u,k)|}\right)\right)^T V_i\right)^2 \\
&\cdot\left(U_u-\lambda_D\left(\frac{\sum_{v\in S_-(u,k)} U_v}{|S_-(u,k)|}\right)\right)+2\lambda_V V_i\,.
\end{aligned} \tag{11}$$

Our TrustRec infers implicit trust relationships between users *not* only by applying the IIR solution but *also* by employing the transitivity of trust; for simplicity, we call it as an *ETT* solution. However, IIR and ETT solutions solve the problems of *different* sets of cold-start users as summarized in Table 5. In the case of those users having almost a zero number of ratings but a few explicit trust relationships, the ETT solution solves their sparsity problem since it infers implicit trust relationships by employing the transitivity of explicit ones. In the case of those users who have rated a small number of items but do not have explicit relationships, the IIR solution solves their sparsity problem by inferring implicit trust and distrust relationships based on the similarity score between users' ratings. If a particular set of users have both ratings and explicit trust relationships, clearly they would take advantage of both solutions. The worst case could happen when users have neither ratings nor explicit trust relationships, which is out of the scope of this paper[1].

[1] One possible solution to this problem is to exploit content information associated with users such as demographic information [21] and location information [28].

Table 5. Applicability of TrustRec to various sets of cold-start users

# of explicit trust relationships	# of ratings	
	Low	Zero
Low	IIR & ETT	ETT
Zero	IIR	×

4. Evaluation

In this section, we evaluate the effectiveness and efficiency of our TrustRec by performing extensive experiments with a real-world dataset. The objective of our experimental study is to answer the following key questions:

Q_1: What are the best *values* of TrustRec's parameters to get the highest accuracy?
Q_2: Are all solutions (i.e., IIR, CTD, STR, and EDI) in TrustRec really effective to achieve better accuracy?
Q_3: How much accurate is TrustRec for *all users* in comparison with existing approaches?
Q_4: How much accurate is TrustRec for *cold-start users* in comparison with existing approaches?
Q_5: How much is the *training time* of TrustRec in comparison with those of existing approaches?

4.1. Experimental Setup

In our experiments, we employed Epinions since it is a real-world dataset, publicly available, and widely used to evaluate recommender systems in the literature as in references [4], [17], [16], [25], [27], and [12]. Epinions contains users' ratings on items and explicit trust/distrust relationships between users. We used 80% of total ratings as a training set and other 20% as a testing set. All required codes were implemented in Java and all the experiments were conducted on an Intel machine equipped with four Core i7-2600K CPUs, 24GB RAM, and a 64-bit Windows 10 operating system. In order to evaluate the effectiveness, we utilized the mean average error (MAE) [2] and the root mean squared error (RMSE) [2] as the two well-known accuracy metrics in the literature, which are defined as follows:

$$MAE = \frac{\sum_{(u,i)\in E} |\hat{R}_{u,i} - R_{u,i}|}{|E|} . \tag{12}$$

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in E} \left(\hat{R}_{u,i} - R_{u,i}\right)^2}{|E|}} . \tag{13}$$

where *E* denotes the set of ratings in the testing set and $\hat{R}_{u,i}$ does the predicted rating for user *u* on item *i*.

We compared the effectiveness and efficiency of TrustRec with those of the following approaches:

- *MF*: it is based on matrix factorization and exploits *only ratings* [13].

- *SocialMF*: it exploits ratings and explicit trust relationships [12].
- *RSTE*: it exploits ratings and explicit trust relationships [17]. The difference between SocialMF and RSTE is that SocialMF employs the transitivity of trust relationships while RSTE does *not*.
- *TrustMF*: it exploits ratings and explicit trust relationships [27]. However, it not only factorizes the rating matrix but also factorizes the trust matrix and incorporates both of them in the matrix factorization model.
- *Impl*: it exploits ratings and both implicit trust and distrust relationships [16].
- *RecSSN*: it exploits ratings and both explicit trust and distrust relationships [25].
- *PushTrust*: it exploits ratings along with *both* explicit and implicit trust/distrust relationships [4].

4.2. Experimental Results

In this section, we answer questions Q_1 to Q_5, one by one.

Q_1: Parameter Tuning There are five parameters in Eq. (10) as *f*, λ_D, λ_T, λ_U, and λ_V where *f* denotes the number of latent features, λ_D and λ_T control the importance of latent feature vectors of distrustees and trustees of a target user, respectively; λ_U and λ_V are the regularization parameters. Also, *h*, σ_T, σ_D, and *k* are other important parameters used in our proposed solutions; the similarity score between two users is computed if they have at least *h* co-rated items; in the IIR solution, σ_T and σ_D are utilized to infer implicit relationships; in the CTD solution, σ_T and σ_D are utilized to confirm the degree of similarity/dissimilarity of users having explicit trust/distrust relationships, respectively; also, in the STR solution, top-*k* trustees/distrustees are selected for each user. Most existing approaches set the values of both λ_U and λ_V as 0.001 to reduce the complexity [4, 5, 12, 16–19, 24, 25]; we follow the same practice. On the contrary, the best values of λ_T and λ_D in existing approaches are quite different and are heavily *dependent* on matrix factorization models; thus we decided to find their best values in our TrustRec. We set the value of *h* as 5 by following [2, 9]. Also, heuristically, we set the values of σ_T and σ_D as 0.1 and -0.1, respectively.

For the rest of parameters, *f*, *k*, λ_T, and λ_D, we employed a two-step approach to determine their best values since finding the best combination among all possible values for these parameters is computationally too expensive. In the first step, while we assigned an *identical* value to λ_T and λ_D (i.e., $\lambda_T = \lambda_D$), we tried to find the best values of *f*, *k*, and λ_T as follows: we set the value of *f* as 5 and 10; for each value of *f*, we set the value of λ_T from 0.1 to 1.0 in step of 0.1; we set the value of *k* as 5, 10, 25, 50, and number of *all* available implicit and explicit relationships. Finally, we measured RMSE and MAE of TrustRec when it was equipped with each of the aforementioned settings. Table 6 shows the results of this parameter where the boldface numbers represent the best accuracy; the best accuracy is observed when *f*, *k*, and λ_T are set as 10, 5, and 0.7, respectively.

In the second step, we tried to find the best *individual* values of λ_T and λ_D as follows: we set $f = 10$ and $k = 5$ based on the result of our parameter tuning in the first step and changed the values of λ_T and λ_D from 0.1 to 1.0 in step of 0.1; then, we measured RMSE and MAE of TrustRec when it was equipped with each of the aforementioned settings. Tables 7 shows the results where the boldface numbers show the best accuracy in the

Table 6. Results of parameter tuning (first step)

	$F = 5$									
	$k = 5$		$k = 10$		$k = 25$		$k = 50$		all	
λ_D=λ_T	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
0.1	0.878	1.090	1.071	1.268	1.111	1.305	0.582	0.782	1.270	1.469
0.2	0.607	0.806	1.121	1.311	0.956	1.162	0.582	0.784	1.099	1.203
0.3	1.046	1.247	1.039	1.242	0.610	0.818	0.640	0.858	1.153	1.371
0.4	0.605	0.803	0.640	0.844	0.676	0.893	0.733	0.963	0.944	1.130
0.5	0.621	0.821	0.657	0.861	0.705	0.920	0.750	0.964	0.863	1.078
0.6	0.613	0.811	0.642	0.843	0.686	0.888	0.732	0.927	0.745	0.941
0.7	0.584	0.782	0.609	0.810	0.650	0.846	0.697	0.878	0.718	0.930
0.8	0.560	0.757	0.568	0.773	0.607	0.809	0.645	0.828	0.708	0.894
0.9	**0.542**	**0.745**	0.543	0.747	0.556	0.766	0.573	0.781	0.655	0.845
1.0	0.544	0.765	0.541	0.763	0.559	0.807	0.548	0.775	0.668	0.887
	$F = 10$									
	$k = 5$		$k = 5$		$k = 25$		$k = 50$		all	
0.1	0.820	1.049	0.596	0.820	0.582	0.800	0.567	0.782	0.858	1.177
0.2	0.612	0.840	0.816	1.026	0.581	0.799	0.570	0.788	0.850	1.103
0.3	0.733	0.965	1.026	1.234	0.610	0.843	0.632	0.880	0.941	1.201
0.4	0.601	0.830	0.608	0.844	0.677	0.940	0.728	1.007	0.733	1.016
0.5	0.573	0.800	0.619	0.850	0.671	0.919	0.721	0.969	0.729	1.000
0.6	0.639	0.838	0.578	0.800	0.614	0.838	0.650	0.866	0.661	0.957
0.7	**0.522**	**0.743**	0.537	0.772	0.549	0.783	0.565	0.799	0.677	0.962
0.8	0.530	0.771	0.546	0.797	0.560	0.820	0.574	0.847	0.686	0.971
0.9	0.569	0.830	0.573	0.837	0.601	0.872	0.647	0.931	0.656	0.950
1.0	0.625	0.898	0.631	0.905	0.610	0.876	0.620	0.891	0.643	0.928

columns and the italic boldface numbers represent the best accuracy in the whole table. As observed, when the values of λ_T and λ_D are *identical* or *very close* to each other, TrustRec provides *high* accuracy; the best accuracy is observed when $\lambda_T = \lambda_D = 0.7$. More specifically, we can assign an identical value to λ_D and λ_T in the range 0.6 to 0.8 or even two values with 0.1 difference in the same range. Table 8 summarizes the final result of our parameter tuning.

Q_2: Effectiveness of Proposed Solutions To answer Q_2, we evaluate TrustRec by removing each of the proposed solutions *one* at a time as follows. To show the effectiveness of the STR solution, we exploit *all* implicit and explicit trustees/distrustees of users; we refer to this version of TrustRec as *TrustRec-S*. To show the effectiveness of the CTD solution, we employ the *top-k* trustees and distrustees for each user without confirming their degree of similarity and dissimilarity with her; we refer to this version as *TrustRec-C*. To show the effectiveness of the IIR solution, we employ *top-k* trustees and distrustees of users that are obtained *only* from their explicit relationships; we refer to it as *TrustRec-I*. To show the effectiveness of the EDI solution, we exploit distrust relationships in the regularization part of the matrix factorization model; we refer to it as *TrustRec-E*.

Table 9 shows the effectiveness of our original TrustRec and its aforementioned versions. The original TrustRec universally outperforms TrustRec-S since the latter one exploits *all* implicit and explicit trustees/ distrustees of a user, some of which may adversely affect the inference of her preferences when their preferences are not that similar/dissimilar to hers. This result also *coincides* with the observation found in Tables 6

Table 7. Results of parameter tuning (second step)

RMSE										
	λ_D									
λ_T	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	1.049	**0.784**	0.792	0.799	0.806	0.814	0.821	0.826	0.840	0.854
0.2	**0.782**	0.840	**0.783**	0.972	0.797	0.798	0.806	0.821	0.836	0.831
0.3	0.792	0.792	0.965	**0.783**	0.790	0.791	0.798	0.814	0.841	0.852
0.4	0.801	0.800	0.830	0.799	**0.782**	0.806	0.791	0.806	0.844	**0.809**
0.5	0.808	0.808	0.807	0.807	0.801	**0.782**	0.782	0.798	0.831	0.827
0.6	0.816	0.816	0.815	0.815	0.804	0.838	0.781	0.791	0.896	0.854
0.7	1.113	1.062	0.822	0.821	0.822	0.821	***0.743***	0.781	0.830	0.921
0.8	0.845	0.825	0.925	0.827	0.827	0.827	0.804	**0.771**	0.849	0.985
0.9	0.829	0.936	0.827	0.834	0.830	0.829	0.826	0.847	**0.830**	1.08
1.0	0.850	1.077	1.165	0.828	0.827	0.829	0.827	1.122	1.183	0.898
MAE										
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	0.820	**0.565**	0.571	0.574	0.578	0.580	0.598	0.602	0.611	0.629
0.2	**0.565**	0.612	**0.564**	0.571	0.580	0.570	0.590	0.597	0.600	0.616
0.3	0.571	0.571	0.732	**0.564**	0.680	0.563	0.580	0.589	0.596	0.630
0.4	0.580	0.580	0.705	0.601	0.583	0.563	0.570	0.580	0.596	**0.603**
0.5	0.588	0.588	0.588	0.589	**0.573**	**0.559**	0.563	0.571	0.589	0.637
0.6	0.594	0.594	0.595	0.595	0.578	0.639	0.550	0.563	0.641	0.712
0.7	0.870	0.813	0.597	0.600	0.603	0.574	***0.522***	0.563	0.580	0.805
0.8	0.618	0.596	0.690	0.598	0.605	0.607	0.563	**0.530**	0.601	0.893
0.9	0.604	0.702	0.599	0.609	0.600	0.605	0.607	0.601	**0.569**	0.895
1.0	0.616	0.846	0.940	0.651	0.700	0.763	0.800	0.890	0.908	0.625

Table 8. Final results of parameter tuning

	f	λ_D	λ_T	λ_U	λ_V	h	σ_T	σ_D	k
Value	10	0.7	0.7	0.001	0.001	5	0.1	−0.1	5

where as the value of *k increases*, the accuracy of TrustRec *decreases*; the *lowest* accuracy is observed when *all* the implicit and explicit trustees/distrustees are exploited. Also, the STR solution contributes to obtain higher *efficiency* since exploiting only top-*k* trustees/distrustees requires much less training time than exploiting all of them. TrustRec outperforms TrustRec-C since the latter exploits trustees/distrustees of each user *without* confirming the similarity scores between them and her; if the trustees/distrustees of a user have actually dissimilar/similar preferences with hers, it negatively affects the inference of her preferences. TrustRec shows better accuracy than TrustRec-I because the latter does *not* exploit implicit relationships. TrustRec outperforms TrustRec-E since the latter one employs the transitivity of distrust relationships that causes the implicit trustees of a user to be considered incorrectly as her implicit distrustees, thereby adversely affecting the inference of the user's true preferences. Also, TrustRec-I performs better than TrustRec-E even though both versions exploit only explicit relationships; the reason is that TrustRec-I employs the intransitivity of distrust relationships while TrustRec-E does not.

In summary, *each* of our solutions employed in TrustRec is *effective* in recommendation and contributes to achieve higher accuracy; the *best* accuracy is obtained when *all* the solutions are employed *together* (i.e., the original TrustRec).

Table 9. Effectiveness of TrustRec with/without each of the proposed solutions

	TrustRec	TrustRec-S	TrustRec-C	TrustRec-I	TrustRec-E
RMSE	0.743	0.889	0.779	0.819	0.852
MAE	0.522	0.702	0.555	0.590	0.612

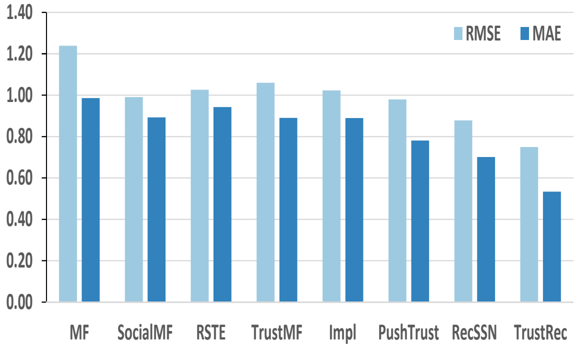


Fig. 3. Accuracy comparison for *all* users

Q_3**: Accuracy for All Users** To answer Q_3, we compared the accuracy of TrustRec with those of existing approaches explained in Section 4.1; Fig. 3 demonstrates the results. MF shows the *worse* accuracy since, on the contrary to trust-aware recommendation approaches, it exploits *only* ratings; this result clearly shows the power of employing trust networks in recommendation. The approaches exploiting both of trust and distrust relationships (i.e., Impl, RecSSN, PushTrust, and TrustRec) outperform those ones exploiting only trust relationships (i.e., SocialMF, RSTE, and TrustMF); this result shows that distrust relationships are also important as trust ones and exploiting them leads to make more accurate recommendations as already observed in previous work [4] [16] [25]. Among Impl, RecSSN, PushTrust, and TrustRec, Impl performs worst since it exploits *only* implicit relationships between users, which are obtained based on the similarity in their ratings. As already discussed (i.e., in Sections 1 and 2), most users usually give ratings on a very small number of items where the implicit trust and distrust relationships may not be inferred well for them. In addition, Impl employs the transitivity of distrust in recommendation; these two problems would cause its low accuracy.

PushTrust exploits implicit as well as explicit relationships, while RecSSN exploits only explicit relationships; however, PushTrust performs *worse* than RecSSN. The reason is that PushTrust exploits implicit trustees for a user without conforming their degree of similarity with her where some of them may have *dissimilar* preferences with the user (i.e., suffering from Problem 2). On the contrary, although RecSSN exploits only explicit relationships, it exploits a number of explicit trustees/distrustees of a user who may have actual dissimilar/similar preferences with her; therefore, RecSSN performs better than PushTrust. RecSSN shows *lower* accuracy than our TrustRec since RecSSN employs the transitivity of distrust relationships and thus may consider implicit trustees of a user as her implicit distrustees, failing to infer her true preferences. Moreover, it cannot make

Table 10. Accuracy improvement(%) obtained by TrustRec over other methods with all users

	MF	SocialMF	RSTE	TrustMF	Impl	PushTrust	RecSSN
RMSE	39.4	24.3	26.9	29.2	26.6	23.4	15.6
MAE	45.8	40.2	43.3	40.0	39.9	31.5	23.8

accurate recommendations for cold-start users since it only considers the explicit relationships.

In summary, TrustRec *significantly* outperforms *all* existing approaches and shows higher accuracy in terms of both RMSE and MAE due to the following reasons: (1) the availability of a sufficient number of implicit trust and distrust relationships due to the IIR solution; (2) the confirmation of similarity/dissimilarity degree among users having trust/distrust relationships due to the CTD solution; (3) the employment of the transitivity of trust relationships and the intransitivity of distrust ones due to the EDI solution. Furthermore, by applying the STR solution, TrustRec somehow *refines* the similar/dissimilar trustees/distrustees to a target user and only exploits the trustees who are highly similar and the distrustees who are highly dissimilar to her. In other words, when we consider all the trustees/distrustees to a target user, some of them may adversely affect the inference of her preferences since their preferences are not that similar/dissimilar to hers. Table 10 represents the *percentage* of *improvement* in accuracy obtained by TrustRec over other approaches; the average improvement in terms of RMSE and MAE over existing approaches are 26.4% and 37.7%, respectively.

Q_4: Accuracy for Cold-start Users To answer Q_4, we compared the accuracy of TrustRec with those of other approaches in the case of considering *only* cold-start users; by following [4], we chose cold-start users as those having rated at most 20 item. Fig. 4 shows the results; as we expected, by comparing Figs 3 and 4, it is observed that the accuracy of *all* approaches decrease when only cold-start users are considered. However, our findings here *coincide* with the ones observed when all users are considered (i.e., shown in Fig. 3); even when considering only cold-start users, all the approaches outperform MF since it exploits only ratings. The approaches exploiting both of trust and distrust relationships (i.e., Impl, RecSSN, PushTrust, and TrustRec) outperform those ones exploiting only trust relationships (i.e., SocialMF, RSTE, and TrustMF), which means distrust relationships are important as trust ones even when only cold-start users are considered. Among Impl, RecSSN, PushTrust, and TrustRec, again Impl performs worst since it exploits only implicit relationships and considers distrust relationships to be transitive. With the same reasons explained on Fig. 3, although PushTrust exploits both implicit and explicit relationships, its accuracy is less that RecSSN where only explicit relationships are exploited. RecSSN shows lower accuracy than TrustRec since it employs the transitivity of distrust relationships and also it basically cannot make accurate recommendations for cold-start users.

When considering only the cold-start users, our TrustRec again significantly outperforms all existing approaches since, on the contrary to other approaches, it employs the four effective solutions IIR, CTD, EDI, and STR in recommendation process. Table 11 represents the percentage of improvement in accuracy obtained by TrustRec over other approaches; the average improvement in terms of RMSE and MAE over existing approaches are 28.3% and 28.0%, respectively.

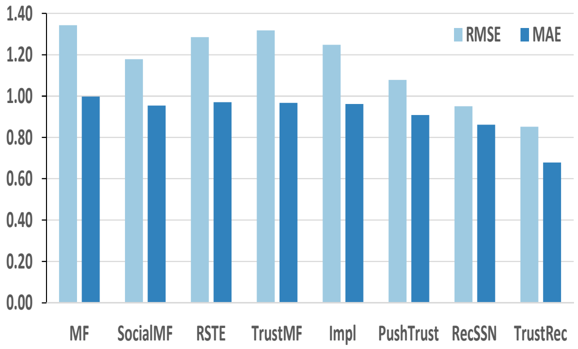


Fig. 4. Accuracy comparison for *cold-start* users

Table 11. Accuracy improvement(%) obtained by TrustRec over other methods with cold-start users

	MF	SocialMF	RSTE	TrustMF	Impl	PushTrust	RecSSN
RMSE	36.5	27.7	33.7	35.3	31.7	21.0	12.5
MAE	31.9	28.8	30.0	29.7	29.3	25.2	21.1

Q_5**: Efficiency** To answer Q_5, first, we measured the *training time* (i.e., efficiency) of TrustRec *with* different values of parameter *k*, which is shown in Table 12. Then, we compared its efficiency with those of existing approaches, which is shown in Fig. 5. We define the training time as the total amount of time spent on learning user and item latent feature matrices. We do not consider the time spent on predicting ratings on users' unrated items since once the user and item latent feature matrices are obtained, *all* the approaches will require the same time to make predictions. As observed in Table 12, the training time of TrustRec increases smoothly as the value of *k* increases, which means TrustRec is *scalable*. The reason is that it exploits relatively a small number of trust and distrust relationships, thanks to the STR solution.

Table 12. Training time of TrustRec with different *k*

	$k = 5$	$k = 10$	$k = 25$	$k = 50$	$k = all$
Time (hour)	0.5	1	2	5	10

In Fig. 5, we set the value of k for TrustRec as 5 since TrustRec shows its best accuracy (i.e., Table 8) and efficiency (i.e., Table 12) when $k = 5$. Although RSTE exploits only explicit trust relationships, its training time is more than RecSSN that exploits both explicit trust and distrust relationships. This is because RSTE exploits trust relationships in the factorization part of the model, thereby using a user u's trustees $|R_u|$ (i.e., the number of ratings given by *u* in a training set) times in a *single* iteration. On the contrary, RecSSN exploits trust and distrust relationships in the regularization part of the model; as a result, it exploits trustees and distrustees of a user *only once* in each iteration. TrustRec *significantly* outperforms *all* other approaches in terms of efficiency due to the STR solution where only top-k similar trustees and dissimilar distrustees of each user are exploited.

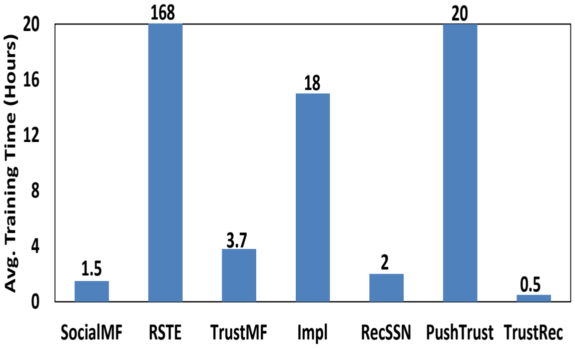


Fig. 5. Comparison of training time

5. Conclusions

In this paper, we analyzed some real-world datasets and observed that explicit trust/distrust relationships are very sparse and some users despite having trust/distrust relationships could have dissimilar/similar preferences in real life. Also, we pointed out that existing trust-aware recommendation approaches require a high computational cost and employing the transitivity of distrust relationships misleads us to considering implicit trustees of a user as her implicit distrustees. We proposed TrustRec that provides an effective solution to each of the aforementioned problems, incorporates all of them together in a single matrix factorization model, and effectively exploits both implicit and explicit relationships. TrustRec exploits more trust/distrust relationships between users by inferring implicit trust/distrust relationships, confirms the degree of similarity/dissimilarity of users having trust/distrust relationships, exploits only top-k most similar/dissimilar trustees/distrustees of each user, and employs the transitivity of trust relationships and the intransitivity of distrust ones. The results of our extensive experiments with a real-world data showed that: 1) each of the proposed solutions is really effective and contributes to achieve better accuracy; 2) TrustRec outperforms all other existing approaches in terms of both accuracy and efficiency when considering all users as well as cold-start users only.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1G1A1007598 and No. NRF-2020R1A2B5B03001960), the research fund of Hanyang University (NO. HY-201800000003218), and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01373, Artificial Intelligence Graduate School Program, Hanyang University).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Aggarwal, C.C.: Recommender Systems: The Textbook. Springer, 1st edn. (2016)
3. Ali, I., Hong, J., Kim, S.: Exploiting implicit and explicit signed trust relationships for effective recommendations. In: ACM SAC. pp. 804–810 (2017)
4. Forsati, R., Barjasteh, I., Masrour, F., Esfahanian, A., Radha, H.: Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations. In: ACM RecSys. pp. 51–58 (2015)
5. Forsati, R., Mahdavi, M., Shamsfard, M., Sarwat, M.: Matrix factorization with explicit trust and distrust side information for improved social recommendation. ACM Trans. on Information Systems 32(4), 17:1–17:38 (2014)
6. Gohari, F., Aliee, F.S., Haghighi, H.: A new confidence-based recommendation approach: Combining trust and certainty. Information Sciences 422, 21–50 (2018)
7. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: WWW. pp. 403–412 (2004)
8. Guo, G., Zhang, J., Smith, N.: A novel recommendation model regularized with user trust and item ratings. IEEE Trans. on Knowledge and Data Engineering 28(7), 1607–1620 (July 2016)
9. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: ACM SIGIR. pp. 230–237 (1999)
10. Hwang, W., Parc, J., Kim, S., Lee, J., Lee, D.: Told you i didn't like it! exploiting uninteresting items for effective collaborative filtering. In: IEEE ICDE. pp. 349–360 (May 2016)
11. Jamali, M., Ester, M.: Trustwalker: A random walk model for combining trust-based and item-based recommendation. In: ACM SIGKDD. pp. 397–406 (2009)
12. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: ACM RecSys. pp. 135–142 (2010)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
14. Lee, J., Jang, M., Lee, D., Hwang, W., J.Hong, Kim, S.: Alleviating the sparsity in collaborative filtering using crowdsourcing. In: CrowdRec, ACM RecSys. pp. 1–6 (2013)
15. Luxburg, U.V.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (Dec 2007)
16. Ma, H.: An experimental study on implicit social recommendation. In: ACM SIGIR. pp. 73–82 (2013)
17. Ma, H., King, I., Lyu, M.: Learning to recommend with social trust ensemble. In: ACM SIGIR. pp. 203–210 (2009)
18. Ma, H., Lyu, M., King, I.: Learning to recommend with trust and distrust relationships. In: ACM RecSys. pp. 189–196 (2009)
19. Ma, H., Yang, H., Lyu, M., King, I.: Sorec: Social recommendation using probabilistic matrix factorization. In: ACM CIKM. pp. 931–940 (2008)

20. Park, C., Kim, D., Oh, J., Yu, H.: Improving top-k recommendation with truster and trustee relationship in user trust network. Information Sciences 374, 100–114 (2016)
21. Park, S., Chu, W.: Pairwise preference regression for cold-start recommendation. In: ACM RecSys. pp. 21–28 (2009)
22. Pham, M., Cao, Y., Klamma, R., Jarke, M.: A clustering approach for collaborative filtering recommendation using social network analysis. Journal of Universal Computer Science 17(4), 583–604 (feb 2011)
23. Punam, B., Pooja, V.: Empowering recommender systems using trust and argumentation. Information Sciences 279, 569–586 (2014)
24. Rafailidis, D.: Modeling trust and distrust information in recommender systems via joint matrix factorization with signed graphs. In: ACM SAC. pp. 1060–1065 (2016)
25. Tang, J., Aggarwal, C., Liu, H.: Recommendations in signed social networks. In: WWW. pp. 31–40 (2016)
26. Wang, Y., Deng, J., Gao, J., Zhang, P.: A hybrid user similarity model for collaborative filtering. Information Sciences 418-419, 102–118 (2017)
27. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE Trans. on Pattern Analysis and Machine Intelligence 39(8), 1633–1647 (Aug 2017)
28. Yin, H., Cui, B., Chen, L., Hu, Z., Zhang, C.: Modeling location-based user rating profiles for personalized recommendation. ACM Trans. Knowledge Discovery from Data 9(3), 1–41 (2015)

Masoud Reyhani Hamedani received the B.S. degree in computer science from Shahid Bahonar University, Kerman, Iran, in 2004, and the M.S. degree in software engineering from Payame Nour University, Tehran, Iran, in 2009, and the PhD degree in computer science from Hanyang University, Seoul, Korea in 2016. He worked as a postdoc researcher in Dankook University, Yongin, Korea until February 2018. In March 2018, he joint Hanyang University and currently is working as an assistant research professor in the Computational Social Science Research Center, Department of Computer and Software. His current research interests include data science, feature representation learning, similarity computation in social network, and deep learning.

Irfan Ali received B.S. and M.S. degrees in computer science from Mehran University of Engineering and Technology Hyderabad, Pakistan in 2012, and the PhD degree in computer science from Hanyang University, Seoul, Korea in 2018.

Jiwon Hong received his BS in computer science from Hanyang University, Seoul, Korea, in 2009. He is currently completing the PhD degree in computer and software at Hanyang University. His research interests include data mining, database, social network analysis, and recommender system.

Sang-Wook Kim received the B.S. degree in computer engineering from Seoul National University, in 1989, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), in 1991 and 1994, respectively. In 2003, he joined Hanyang University, Seoul, Korea, where he currently is a professor at the Department of Computer Science and Engineering and the director of the Brain-Korea21-Plus research program. He is also leading a National Research Lab (NRL) Project funded by the National Research Foundation since 2015. From 2009 to 2010, he visited the Computer Science Department, Carnegie Mellon University, as a visiting professor. From 1999 to 2000, he worked with the IBM T. J. Watson Research Center, USA, as a postdoc. He also visited the Computer Science Department at Stanford University as a visiting researcher in 1991. He is an author of more than 200 papers in refereed international journals and international conference proceedings. His research interests include databases, data mining, multimedia information retrieval, social network analysis, recommendation, and web data analysis. He is a member of the ACM and the IEEE.

Received: June 08, 2020; Accepted: October 10, 2020.

Computer Science and Information Systems 18(1):115–128 https://doi.org/10.2298/CSIS200515046P

A Dual Hybrid Recommender System based on SCoR and the Random Forest

Costas Panagiotakis[1], Harris Papadakis[2], and Paraskevi Fragopoulou[2]

[1] Department of Management Science and Technology
Hellenic Mediterranean University
72100 Agios Nikolaos, Crete, Greece
Tel.: +30-28410-91203
cpanag@hmu.gr

[2] Department of Electrical and Computer Engineering
Hellenic Mediterranean University
71004 Heraklion, Crete, Greece
Tel.: +30-2810-379119
adanar@hmu.gr, fragopou@ics.forth.gr [⋆]

Abstract. We propose a Dual Hybrid Recommender System based on SCoR, the Synthetic Coordinate Recommendation system, and the Random Forest method. By combining user ratings and user/item features, SCoR is initially employed to provide a recommendation which is fed into the Random Forest. The two systems are initially combined by splitting the training set into two "equivalent" parts, one of which is used to train SCoR while the other is used to train the Random Forest. This initial approach does not exhibit good performance due to reduced training. The resulted drawback is alleviated by the proposed dual training system which, using an innovative splitting method, exploits the entire training set for SCoR and the Random Forest, resulting to two recommender systems that are subsequently efficiently combined. Experimental results demonstrate the high performance of the proposed system on the Movielens datasets.

Keywords: recommender systems, synthetic coordinates, random forest.

1. Introduction

The explosive growth and variety of information available on the Web frequently overwhelm users and lead them to make poor choices. This problem is addressed by Recommender Systems (RS), that have become increasingly popular in guiding users to make more wise decisions [28]. Recommender Systems provide the degree of preference of a user for an item for a variety of entities such as e-shop items, web pages, news, articles, movies, music, hotels, television shows, books, restaurants, friends, etc.

A variety of techniques have emerged in the field of recommender systems. One of the main techniques is Similarity-based Collaborative Filtering (CF) [1,3], classified into user-based Collaborative Filtering and item-based Collaborative Filtering. CF is based on a similarity function that takes into account user preferences and outputs similarities for

[⋆] C. Panagiotakis and P. Fragopoulou are also with the Foundation for Research and Technology-Hellas (FORTH), Institute of Computer Science, 70013 Heraklion, Crete, Greece.

pairs of users. More specifically, the basic idea of user-based CF approaches is to detect a set of users who have similar favorite patterns to a given user (i.e., “neighbor” set of the user) and recommend to the user those items that others in its “neighbor” set like. While, item-based CF approaches recommend an item to a user based on other items with high correlations (i.e., “neighbor” set of the item).

In Dimensionality Reduction methods, each user or item is represented by a vector, where a user’s vector is the set of his ratings for all items in the system. The sparsity of these vectors renders it difficult to identify correlations between user-item pairs. For this reason, Dimensionality Reduction techniques are employed, such as Singular Value Decomposition [6], Principal Component Analysis, Probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation [17]. The Matrix Factorization method [13,11] that characterizes both items and users by vectors of latent factors inferred from item rating patterns, is also a Dimensionality Reduction technique. High correlation between item and user factors lead to recommendations.

In [23], the SCoR Recommender System was proposed. SCoR assigns synthetic coordinates (vectors) to users and items (nodes) as proposed in [6], but instead of using the dot product, SCoR uses the Euclidean distance between a user and an item in the Euclidean space, so that, when the system converges, the distance between a user-item pair provides an accurate prediction of that user’s preference for the item. SCoR has several benefits, it is parameter free, thus does not require parameter tuning to achieve high performance, and is more resistant to the cold-start problem compared to other algorithms from the literature. The Vivaldi synthetic network coordinates algorithm, which lies at the back-end of SCoR, has been successfully applied to movie recommendation [23], personalized video summarization [19], detection of abnormal profiles in RS [18,20], community detection [22], and to the interactive image segmentation problem [21] providing high performance results compared to other state-of-the-art methods on public datasets.

Different architectures of artificial neural networks and deep learning methods have been found to be effective in several domains including computer vision, pattern recognition [27,7] and also in recommender systems [9]. These methods are powerful in processing unstructured multimedia data for feature representation learning like audio, text, image and videos. Convolutional Neural Networks (CNNs) [9] have been applied on the results of a preprocessing step e.g. the outer product between user and item ratings to obtain the 2D interaction map, in order to model the user item interaction patterns and to capture high-order correlations. Deep Hybrid Models for Recommendation integrates neural building blocks to formalize more powerful and expressive models. In [30] and [5], a CNN and an RNN based hybrid models for hashtag and citation recommendation are proposed, respectively.

In [24], the extended LSTM model with a higher-order interaction layer, proposed in [24], is able to handle data sparsity, includes a novel attention mechanism to reduce the burden of encoding the entire user history into a cell vector, and time aware input and forget gates to handle irregular time gaps between input interactions [24]. The three hidden layer-system proposed in [24], has been evaluated on the users from Twitter, Google Plus and YouTube. The authors used Tweets and Google Plus posts and YouTube videos either liked or added to playlists.

Random Forest algorithms [2] have been successfully employed in recommender systems [29,26]. In [29], the authors propose a framework that integrates three-way decision

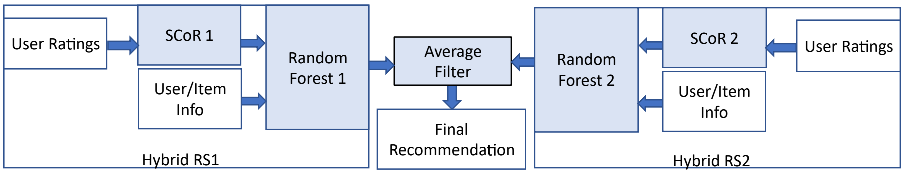


Fig. 1. The schema of the proposed dual hybrid system architecture.

and Random Forests to build recommender systems. Three way decision was introduced to map user recommendations for items, to "recommend", "not recommend", or "consult the user" actively for his/her preference. In [26], the authors propose a framework that employs reinforcement learning to derive good policies for Personalized Ad Recommendation (PAR) systems. Random Forest regression is used to efficiently learn a PAR policy.

There also exist hybrid methods that combine more than one approaches in order to improve recommendation accuracy. The system proposed in this paper belongs to this category. In [28], the UO-CRBMF model is combined with the IC-CRBMF model [14] to improve recommendation accuracy. In [25], a hybrid approach is proposed and applied to learning material. This hybrid system consists of attribute-based filtering and a genetic-based recommender system in order to improve the quality of recommendation in an e-learning environment. In [15], a Personalized Context-Aware Hybrid Travel Recommender System (PCAHTRS) is proposed, providing personalized tourist recommendations based on user ratings and their preferences. The hybrid recommendation algorithm employs user-based similarity, user's point-of-interest similarity, implicit user profiles and user's point-of-interest opinion similarity to predict users' ratings for tourist attractions.

In this paper, we propose a Dual Hybrid Recommender System by combining SCoR and the Random Forest approaches. SCoR receives user ratings and provides an initial recommendation to the Random Forest that gets as input user and item features to provide the final recommendation. A problem in such approaches is the use of smaller training sets in order to train both systems, which reduces the performance of each individual system. This problem is alleviated by the final proposed dual training system, resulting to two "equivileant" recommender systems that are efficiently combined. In addition, better results are obtained thanks to a novel method, proposed in this paper, that splits the training set into two "equivalent" parts for the training purposes of SCoR and the Random Forest, respectively. Figure 1 depicts the schema of the proposed dual hybrid system architecture. According to the proposed architecture, we have trained two SCoR systems (SCOR 1 and SCoR 2) and two Random Forest (Random Forest 1 and Random Forest 2) getting two recommendations. The final recommendation is given by the average of recommendations of Random Forest 1 and Random Forest 2.

The main contribution of this work is the improvement of the results of SCoR by efficiently combining context features and user ratings, while taking advantage of the Random Forest integration. The proposed approach, based on a dual process, also provides interesting directions towards alleviating two well-know problems in the field of recommender systems, namely, the cold-start and the data sparsity problems [12]. The user cold start problem appears in model-based methods, like SCoR [23] and Matrix Factorization

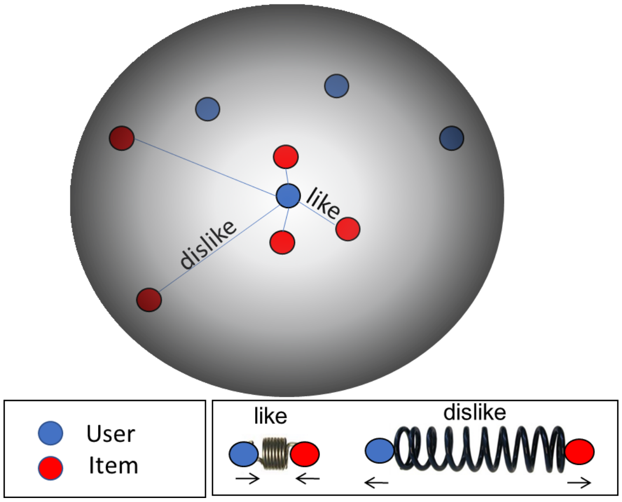


Fig. 2. A synthetic example following the execution of SCoR that shows the position of nodes (users and items) in $\mathbb{R}^2$. The item preferences for the user located in the center of the graph is indicated by the brightness of the graph background - from light grey (like) to darker grey (dislike).

[13], when a new user arrives and the system does not have user's historical behavior data. Data sparsity appears when several users have rated only a small subset of the items. The proposed system is highly flexible since any model-based method can be easily integrated by replacing SCoR. The selection of the remaining input features of Random Forest is also flexible, making the applicability of this work possible to any context where user ratings and user/item information are available.

The rest of the paper is organized as follows: Section 2 presents in detail the proposed dual hybrid recommend system. Section 3 describes the experimental setup along with the obtained results. Conclusions are provided in Section 4.

2. The Proposed Recommender System

2.1. SCoR

The proposed system (see Fig. 1) is based on SCoR, a novel personalized recommendation algorithm [23]. SCoR uses a Model-based Collaborating Filtering approach, which is dependent on a known set of user-to-item ratings, in order to train a preference prediction model. Thus, a number of preferences (ratings) of each user for some items must be already known. These are provided in the form of triplets (u, i, r), where r is the scalar rating of user u for item i.

In the core of SCoR lies the spring metaphor which inspired the Vivaldi synthetic network coordinate algorithm [4]. Essentially, the basis of SCoR is a Synthetic Euclidean Coordinate system, which randomly assigns a position in an N-dimensional Euclidean space to each element in the user U and the item I sets. The algorithm iteratively updates the positions of all elements (users and items) until, for every known rating (u, i, r), the Euclidean distance between user u and item i corresponds to the value r. The positions are updated using (1), as follows:

$$p(x) = p(y) + \delta \cdot (dd(x, y) - d(x, y)) \cdot b(x, y) \tag{1}$$

$$b(x, y) = \frac{p(x) - p(y)}{d(x, y)} \tag{2}$$

where $p(x)$, $p(y)$ are the positions of a user-item pair, $d(x, y)$ is their current Euclidean distance, $dd(x, y)$ is their desired distance (based on the rating value r). The unit vector $b(x, y)$ provides the direction towards which node x should move, and δ controls the method's convergence, since it is the fraction of distance node x is allowed to move toward its ideal position. Upon algorithm conversion, the Euclidean distance between user u and an unrated (by user u) item i provides a prediction for the preference of user u for item i. Thus, after the training phase, SCoR is able to provide a recommendation $\hat{r}(u, i)$ for any given user-item pair (u, i) in $O(1)$ based on the Euclidean distance between u and i. More details about SCoR can be found in [23].

Algorithm 1 shows the pseudo-code of the SCoR system. The input to the system is the set of users U, the set of items I and the values $minR$ (smallest rating), $maxR$ (highest rating). The training set TS and the test set VS consist of the given recommendations $(u, i, r(u, i)) \in TS$ and the predicted recommendations $\widehat{r}(u, i)$, with $(u, i) \in VS$ (produced by SCoR), respectively. $MSE(u)$ is the Mean Square Error of node u and its neighbors, while the procedure $getWeightedRandomSample$ selects nodes with smaller error more often for position updates. More details about SCoR can be found in [23].

Figure 2 shows a synthetic example after the execution of SCoR that shows the position of nodes (users and items) in $\mathbb{R}^2$. The distance between user u and item i corresponds to the predicted preference of u for item i. The item preferences for the user located in the center of the graph is indicated by the brightness of the graph background - from light grey (like) to darker grey (dislike).

2.2. Hybrid Recommender System

The proposed Hybrid Recommender System is based on the Random Forest approach, where the goal is to learn the recommendation for a given pair (u, i) taking as input: user's u and item's i information and the prediction $\hat{r}(u, i)$ of SCoR. In order to train the system, we have to use different training sets for SCoR and the Random Forest, otherwise low performance results are obtained (see *Hybrid RF* method in Section 3). The low performance is due to the fact that SCoR yields very low error for the instances of its training set, which is not the case for the test set. If we use the same training set for Random Forest, then it will give very high confidence to the input features provided by SCoR, making incorrect predictions on the test set.

In the proposed approach, the training set of ratings T is efficiently split into two "equivalent" parts (T_1 and T_2) to train SCoR and the Random Forest. Algorithm 2 presents the proposed splitting method described hereafter. Let G be the graph that shows the connections (ratings) between users and items in the training set (see Fig. 3(a)). The nodes of the graph are the union of users and items, and the edges of the graph correspond to the

Algorithm 1: The *SCoR* algorithm.

```
input  : $i \in I, u \in U, (u, i, r(u, i)) \in TS, minR, maxR.$
output: $\widehat{r}(u, i)$
foreach $u \in U,$ do
    $p(u) =$ random position in $\mathbb{R}^n$
end
foreach $i \in I$ do
    $p(i) =$ random position in $\mathbb{R}^n$
end
repeat
    $(u, i) = getRandomSample(TS)$
    $[p(u), p(i)] = Vivaldi(p(u), p(i), r(u, i))$
until $\forall x \in I \cup U$ $p(x)$ is stable
foreach $(u, i) \in TS$ do
    $W(u, i) = e^{-0.2 \cdot MSE(u)} \cdot (dd(u, i) - d(u, i))^2$
end
repeat
    $(u, i) = getWeightedRandomSample(TS, W)$
    $[p(u), p(i)] = Vivaldi(p(u), p(i), r(u, i))$
until $\forall x \in I \cup U$ $p(x)$ is stable
foreach $(u, i) \in VS$ do
    $\widehat{r}(u, i) = maxR - (maxR - minR) \cdot \frac{||p(u)-p(i)||_2}{100}$
    $\widehat{r}(u, i) = min(max(\widehat{r}(u, i), minR), maxR)$
end
```

ratings in the training set. The splitting method tries to minimize the sum of the relative differences between the degree of a node (user or item) $x \in G$ in T_1 and T_2 (see Fig. 3(b)). The following function $f_{T_1,T_2}(x)$ shows the relative difference between the degrees of node x in T_1 $(deg_{T_1}(x))$ and T_2 $(deg_{T_2}(x))$:

$$f_{T_1,T_2}(x) = \frac{|deg_{T_1}(x) - deg_{T_2}(x)|}{\varepsilon + \min(deg_{T_1}(x), deg_{T_2}(x))}, \tag{3}$$

where ε is a small constant to prevent the zero in the denominator, e.g. $\varepsilon = 1$. When $f_{T_1,T_2}(x)$ is minimized, then $deg_{T_1}(x) = deg_{T_2}(x)$ or $|deg_{T_1}(x) - deg_{T_2}(x)| = 1$, which means that node x has almost the same number of edges in T_1 and T_2.

The proposed method is iterative and is based on sequential minimization. It starts from an initial "random" solution (see line 1 of Algorithm 2) and in every step it identifies the best pair of ratings (r_1 and r_2) that should be exchanged between T_1 and T_2 in order to minimize the following objective function $F(T_1, T_2)$: (see lines 3-16 of Algorithm 2).

$$F(T_1, T_2) = \sum_{x \in G} f_{T_1,T_2}(x) \tag{4}$$

The method terminates when there is no pair of ratings (r_1 and r_2) that can be exchanged between T_1 and T_2 which can further reduce the objective function (see line 17 of Algorithm 2). Following the execution of the proposed splitting method, a user u or an item v has almost the same number of connections (number of ratings) in T_1 and in T_2, while the total number of ratings in T_1 equals those in T_2.

Figure 3 depicts a synthetic example of a graph G and a splitting of its edges into sets T_1 and T_2. In this example, graph G consists of 8 nodes (3 users and 5 items) with 10

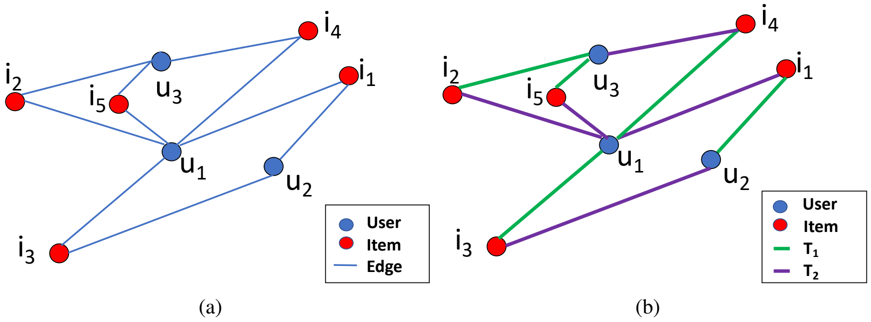


Fig. 3. A synthetic example of **(a)** a graph G and **(b)** the splitting of its edges into sets T_1 and T_2.

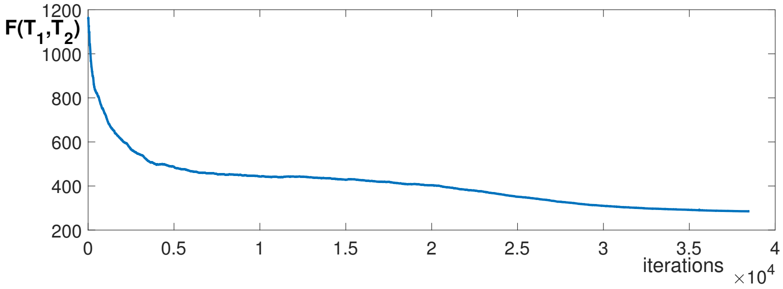


Fig. 4. The evolution of $F(T_1, T_2)$ on the *ML-100k* dataset.

edges (ratings). The splitting shown in 3(b) consists of two equivalent sets T_1 and T_2, each with 5 edges. In addition, each node has almost the same degree in both sets T_1 and T_2 according to the proposed splitting method. Figure 4 depicts the evolution of $F(T_1, T_2)$ on the *ML-100k* dataset. At convergence, it holds that $F(T_1, T_2)$ is not zero due to the existence of nodes with odd degree.

2.3. Dual Hybrid Recommender System

The two modules (SCoR and Random Forest) of the single Hybrid Recommender System presented in the previous subsection do not take advantage of the entire training set, since SCoR is only trained by T_1 and the Random Forest is only trained by T_2. The problem of reduced training for each individual module is alleviated by the proposed Dual Hybrid Recommender System.

The dual hybrid system is based on the dual training of two single Hybrid Recommender Systems (*Hybrid RS1* and *Hybrid RS2*). The training set of *Hybrid RS1* is used to train *Hybrid RS2* and vice versa, thus allowing each module to exploit the entire training set. Figure 5 depicts the proposed training schema. In *Hybrid* $RS1$, SCoR 1 is trained by T_1 and the Random Forest 1 by T_2. In *Hybrid* $RS2$, SCoR 2 is trained by T_2 and Ran-

Algorithm 2: The proposed splitting method.

```
input : $T, G$
output: $T_1, T_2$
$[T_1, T_2] = randomSplit(T)$
repeat
    foreach $r \in T_1$ do
        $[u, v] = getUserItem(G, r)$
        $c_1(r) = f_{T_1-\{r\},T_2\cup\{r\}}(u) - f_{T_1,T_2}(u) + f_{T_1-\{r\},T_2\cup\{r\}}(v) - f_{T_1,T_2}(v)$
    end
    foreach $r \in T_2$ do
        $[u, v] = getUserItem(G, r)$
        $c_2(r) = f_{T_1\cup\{r\},T_2-\{r\}}(u) - f_{T_1,T_2}(u) + f_{T_1\cup\{r\},T_2-\{r\}}(v) - f_{T_1,T_2}(v)$
    end
    $r_1 = argmin(c_1)$
    $r_2 = argmin(c_2)$
    if $c_1(r_1) + c_2(r_2) < 0$ then
        $T_1 = T_1 - \{r_1\}, \ T_2 = T_2 \cup \{r_1\}$
        $T_1 = T_1 \cup \{r_2\}, \ T_2 = T_2 - \{r_2\}$
    end
until $c_1(r_1) + c_2(r_2) < 0$
```

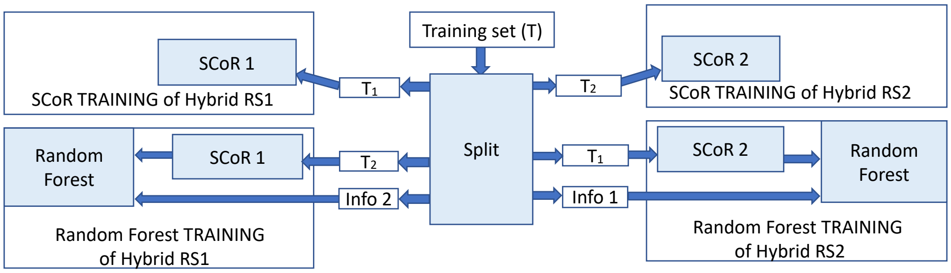

Training set (T)
SCoR 1
T₁
Split
T₂
SCoR 2
SCoR TRAINING of Hybrid RS1
SCoR TRAINING of Hybrid RS2
Random Forest
SCoR 1
T₂
T₁
SCoR 2
Random Forest
Info 2
Info 1
Random Forest TRAINING of Hybrid RS1
Random Forest TRAINING of Hybrid RS2

Fig. 5. The schema of the dual training system.

dom Forest 2 by T_1. Both systems have almost equal performance due to the "equivelant" training sets provided by the novel splitting method. Therefore, averaging the recommendations of the two equivalent systems to provide the final recommendation comes as a natural choice. This technique is illustrated in Figure 1.

3. Experimental Results

The experiments on the proposed *Dual Hybrid RS* method, are compared to the following three baseline methods

- *CF*: The user-based Collaborative Filtering approach with cosine similarity [1].
- *SCOR*: The method based exclusively on SCoR [23].
- *RF*: The method based exclusively on the Random Forest [2].

and the following two variants of the proposed method:

- *Hybrid RS*: The proposed single Hybrid Recommender System that uses SCoR and the Random Forest, both trained by the same entire training set.

- *Hybrid RS1 or Hybrid RS2*: The proposed single Hybrid Recommender System that uses SCoR and the Random Forest, each with half the training set as provided by the splitting method (T_1 for SCoR and T_2 for the Random Forest in $RS1$, and T_2 for SCoR and T_1 for the Random Forest in $RS2$).

3.1. Datasets and Features

The experiments are performed on the two well-known MovieLens datasets [8,23,29] with 5 rating-gradations (1-5) [3]:

- *ML-100k* consisting of 943 users and 1682 movies with 100,000 ratings.
- *ML-1M* consisting of 6040 users and 3883 movies with 1,000,000 ratings.

For the Random Forest, we use the following input features for each user and movie entry:

- *User entry*: average rating, number of ratings, age, gender and profession.
- *Movie entry*: average rating, number of ratings and genre.

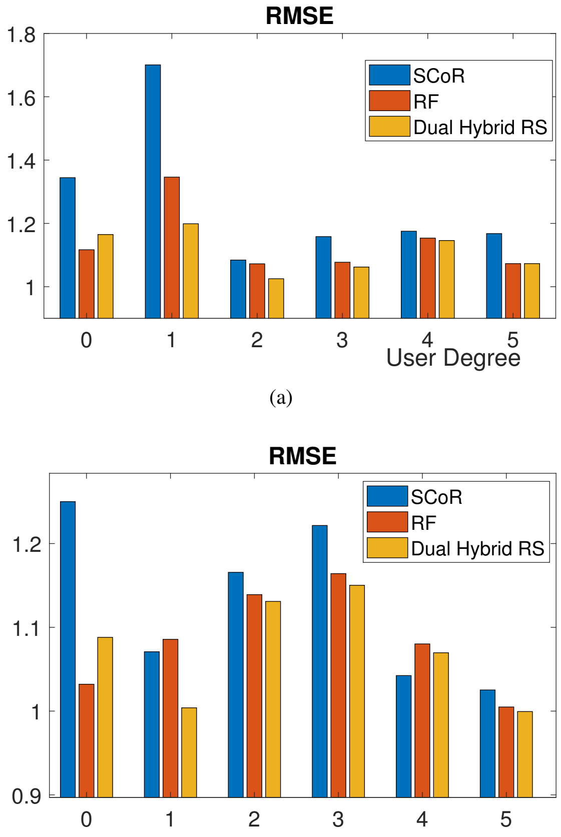
3.2. Performance Evaluation

Table 1. The $RMSE$ values for the proposed method and its variations on the *ML-100K* (left) and *ML-1M* (right) datasets.

ML-100K		**ML-1M**	
METHOD	**RMSE**	**METHOD**	**RMSE**
CF	0.9522	CF	0.9508
SCOR	0.9474	SCOR	0.9576
RF	0.9450	RF	0.9551
Hybrid RS	0.9994	Hybrid RS	0.9853
Hybrid RS1	0.9517	Hybrid RS1	0.9765
Hybrid RS2	0.9569	Hybrid RS2	0.9782
Dual Hybrid RS	**0.9393**	**Dual Hybrid RS**	**0.9474**

The original dataset is randomly divided into training set (80%) and test set (20%). To evaluate the performance of the proposed method and its variations, we report the Root Mean Squared Error ($RMSE$) [10,1,23] for the test set. Table 1 presents the $RMSE$ values of the proposed method and its variations on the *ML-100K* and the *ML-1M* datasets. It is evident that the proposed method (*Dual Hybrid RS*) clearly outperforms all the remaining methods for both datasets. The second and third methods in performance are RF and $SCOR$, respectively, while the single hybrid (*Hybrid RS1* or *Hybrid RS2*) is the fourth method in performance due to reduced training. The worst results are obtained for method *Hybrid RS*, that uses the same training set for SCoR and the Random Forest.

[3] After the publication of this work, our intention is to make publicly available the code implementing the proposed method and the experiments performed.



(a)

(b)

Fig. 6. The $RMSE$ values for the proposed method and two baselines methods (SCoR and RF) on the **(a)** *ML-100K* and the **(b)** *ML-1M* datasets, as a function of the user degree.

3.3. Computational performance

The proposed system has been implemented using MATLAB, apart from SCoR which was implemented in Java. All experiments were performed on an Intel I7 CPU processor at 2.4 GHz. The processing time for training the *Dual Hybrid RS* for the *ML-100K* and *ML-1M* datasets are 70 secs and 25 mins, respectively. For the *ML-100K* dataset, it holds that 4%, 8% and 88% of the total processing time is consumed by the splitting method, SCoR and random forest, respectively. For *ML-1M*, it holds that 15%, 4% and 79% of the total processing time is consumed by the splitting method, SCoR and random forest, respectively. These results can be explained by the fact that the computational complexity of SCoR is $O(N)$, the complexity of the average case random forest training is $O(N \cdot$

$log^2 N)$ [16], while that of the splitting method is $O(N^2)$, where N denotes the number of training samples.

The processing time required for the execution of the *Dual Hybrid RS*, in order to predict the ratings of the test set for *ML-100K* and *ML-1M*, are 1.2 and 72 secs, respectively. This can be explained by the fact that the computational complexity of the pre-trained *Dual Hybrid RS* increases linearly with the size of the dataset, since it only uses two pre-trained random forests.

3.4. Cold Start and Sparsity Problems

This section examines the stability and efficiency of the proposed dual system with respect to the Cold Start and the Sparsity Problems [12], two well-known issues in recommender systems. The cold start problem occurs, when a new user or item enters the system. Sparsity appears for users that have rated only a small subset of the items, or items that have been rated by few users.

In order to examine the behavior of the proposed system with respect to the aforementioned problems, ratings from the training set are moved to the test set to ensure that there exists a minimum of users (e.g. 200 users) with zero or low degree (≤ 5) in the training set. Subsequently, we train the recommendation system and measure the $RMSE$ values in the test set, for the users as a function of the number of their ratings in the training set (user degree) as shown in Figure 6. It holds that for the cold start problem (zero degree users), as expected, SCoR fails to provide good recommendations, while the *RF* method and the proposed *Dual Hybrid RS* both yield satisfactory results. For the sparsity problem (users with small number of ratings), the proposed dual hybrid system yields slightly better results than RF and SCoR. This experiment demonstrates that the proposed method *Dual Hybrid RS* is a good combination of SCoR and the Random Forest that exploits the advantages of both systems and performs well on the Cold Start and the Sparsity Problems.

4. Conclusions

We presented a Dual Hybrid Recommender System based on the SCoR Recommender System and the Random Forest approaches. The proposed system efficiently combines context features and user ratings taking advantage of the Random Forest integration. In order to train the system, the training set is split into two "equivelant" parts, each one used to train one of the modules (SCoR or Random Forest) resulting to reduced training for both modules.

The proposed Dual Hybrid Recommender System improves the single training approach and it outperforms all the baseline methods and their variations as presented in our experimental results on the Movielens datasets. Furthermore, it performs well on the cold start and the sparsity problems. As future work, we plan to apply the proposed methodology to other datasets with richer context based features.

Acknowledgements. This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T1EDK-02147).

References

1. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering 24(5), 896–911 (2012)
2. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
3. Cai, Y., Leung, H.f., Li, Q., Min, H., Tang, J., Li, J.: Typicality-based collaborative filtering recommendation. IEEE Transactions on Knowledge and Data Engineering 26(3), 766–779 (2013)
4. Dabek, F., Cox, R., Kaashoek, F., Morris, R.: Vivaldi: A decentralized network coordinate system. In: ACM SIGCOMM Computer Communication Review. vol. 34, pp. 15–26. ACM (2004)
5. Ebesu, T., Fang, Y.: Neural citation network for context-aware citation recommendation. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 1093–1096. ACM (2017)
6. Gorrell, G.: Generalized hebbian algorithm for incremental singular value decomposition in natural language processing. In: 11st Conference of the European Chapter of the Association for Computational Linguistics (2006)
7. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: A review. Neurocomputing 187, 27–48 (2016)
8. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) 5(4), 19 (2016)
9. He, X., Du, X., Wang, X., Tian, F., Tang, J., Chua, T.S.: Outer product-based neural collaborative filtering. arXiv preprint arXiv:1808.03912 (2018)
10. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22(1), 5–53 (2004)
11. Hwang, W.S., Li, S., Kim, S.W., Lee, K.: Data imputation using a trust network for recommendation via matrix factorization. Computer Science and Information Systems 15(2), 347–368 (2018)
12. Jain, S., Grover, A., Thakur, P.S., Choudhary, S.K.: Trends, problems and solutions of recommender system. In: International Conference on Computing, Communication & Automation. pp. 955–958. IEEE (2015)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
14. Liu, X., Ouyang, Y., Rong, W., Xiong, Z.: Item category aware conditional restricted boltzmann machine based recommendation. In: International Conference on Neural Information Processing. pp. 609–616. Springer (2015)
15. Logesh, R., Subramaniyaswamy, V.: Exploring hybrid recommender systems for personalized travel applications. In: Cognitive informatics and soft computing, pp. 535–544. Springer (2019)
16. Louppe, G.: Understanding random forests: From theory to practice. arXiv preprint arXiv:1407.7502 (2014)
17. Mobasher, B., Burke, R.D., Sandvig, J.J.: Model-based collaborative filtering as a defense against profil injection attacks. In: The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA. pp. 1388–1393 (2006)
18. Panagiotakis, C., Papadakis, H., Fragopoulou, P.: Detection of hurriedly created abnormal profiles in recommender systems. In: International Conference on Intelligent Systems (2018)
19. Panagiotakis, C., Papadakis, H., Fragopoulou, P.: Personalized video summarization based exclusively on user preferences. In: European Conference on Information Retrieval (2020)
20. Panagiotakis, C., Papadakis, H., Fragopoulou, P.: Unsupervised and supervised methods for the detection of hurriedly created profiles in recommender systems. Machine Learning and Cybernetics (2020)

21. Panagiotakis, C., Papadakis, H., Grinias, E., Komodakis, N., Fragopoulou, P., Tziritas, G.: Interactive image segmentation based on synthetic graph coordinates. Pattern Recognition 46(11), 2940–2952 (2013)
22. Papadakis, H., Panagiotakis, C., Fragopoulou, P.: Distributed detection of communities in complex networks using synthetic coordinates. Journal of Statistical Mechanics: Theory and Experiment 2014(3), P03013 (2014)
23. Papadakis, H., Panagiotakis, C., Fragopoulou, P.: Scor: A synthetic coordinate based system for recommendations. Expert Systems with Applications 79, 8–19 (2017)
24. Perera, D., Zimmermann, R.: Lstm networks for online cross-network recommendations. In: IJCAI. pp. 3825–3833 (2018)
25. Salehi, M., Kamalabadi, I.N.: Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree. Knowledge-Based Systems 48, 57–69 (2013)
26. Theocharous, G., Thomas, P.S., Ghavamzadeh, M.: Personalized ad recommendation systems for life-time value optimization with guarantees. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
27. Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E.: Deep learning for computer vision: A brief review. Computational intelligence and neuroscience 2018 (2018)
28. Xie, W., Ouyang, Y., Ouyang, J., Rong, W., Xiong, Z.: User occupation aware conditional restricted boltzmann machine based recommendation. In: Internet of Things (iThings), 2016 IEEE International Conference on. pp. 454–461. IEEE (2016)
29. Zhang, H.R., Min, F.: Three-way recommender systems based on random forests. Knowledge-Based Systems 91, 275–286 (2016)
30. Zhang, Q., Wang, J., Huang, H., Huang, X., Gong, Y.: Hashtag recommendation for multimodal microblog using co-attention network. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia. pp. 3420–3426 (2017)

Costas Panagiotakis received the B.A, M.Sc., and Ph.D. degrees from the Dep. of Computer Science, University of Crete, Greece, in 2001, 2003, and 2007, respectively. Currently, he is Associate Professor and Head in Department of Management Science and Technology, Hellenic Mediterranean University and a researcher at the Institute of Computer Science (ICS), Foundation for Research and Technology-Hellas (FORTH) in Heraklion, Crete, Greece. He is the author of one book and more than 70 articles in international journals and conferences. His interests include image analysis, pattern recognition, multimedia and signal processing. For more information, please visit
https://sites.google.com/site/costaspanagiotakis/

Harris Papadakis received his B.Sc. and PhD in Computer Science from the University of Crete in 2001 and 2011 respectively, and the M.Sc. in Computer Science from the University of Patras in 2005. He is an Assistant Professor (under appointment) at the Department of Electrical and Informatics Engineering of the Hellenic Mediateranean University. He has worked in Large Scale Distributed Systems as well as Information Analysis. He has published over 40 papers related to the field of Data mining, Graph analysis, Community detection and Recommender systems as well as Distributed and P2P Systems.

Paraskevi Fragopoulou received her B.Sc. in Computer Science from the University of Crete in 1989, and the M.S. and Ph.D. degrees in Computer Science from Queen's University, Ontario, Canada in 1990 and 1995, respectively. Currently, she is a Professor of

Computer Science at the Department of Electrical and Computer Engineering, Hellenic Mediterranean University, Crete, Greece, and an Associated Researcher at the Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH-ICS) as member of the Distributed Computing Systems and Cybersecurity (DiSCS) Laboratory. She has co-authored more than 60 conference/journal papers and book chapters. Her primary research interests are in the areas of Distributed Algorithms, Internet Technology, Recommender Systems and Online Social Networks.

Received: May 15, 2020; Accepted: September 9, 2020.

Computer Science and Information Systems 18(1):129–154 https://doi.org/10.2298/CSIS200221032J

A method of assessing rework for implementing software requirements changes

Shalinka Jayatilleke[1] and Richard Lai[2]

[1] Department of Management, Sports and Tourism
La Trobe University, Victoria. 3086, Australia
s.jayatilleke@latrobe.edu.au
[2] Department of Computer Science and Computer Engineering
La Trobe University, Victoria. 3086, Australia
R.Lai@latrobe.edu.au

Abstract - Software development is often affected by user/system requirements changes. To implement requirements changes, a system which is being developed needs to be reworked. However, the term "Rework" has not been clearly defined in the literature. Depending on the complexity of the changes, the amount of rework required varies from some software module modifications to a non-trivial alteration to the software design of a system. The effort associated with such a rework obviously will vary too. To date, there has been scant research on rework assessment, and the relationship between it and change effort estimation is hardly understood. In this paper, we present a definition for rework, and describe a method of assessing rework for implementing software requirements changes. Our method consists of three stages: namely (i) change identification; (ii) change analysis; and (iii) rework assessment. To demonstrate the practicality that it enables developers to compare the rework between the different options available for implementing a requirements change and to identify the one which is less invasive and requires lesser amount of modifications to the software system design, we explain our concept with the use of a running example.

Keywords - Rework, Rework assessment, Requirements changes, Requirements change management, Software System Design Document

1. Introduction

Software development is often affected by changes in user/system requirements. Rapid changes in requirements are found to be one of the main cost drivers [1]; and they have a significant impact on development efforts and project duration [2, 3]. To implement requirements changes, a system in design phase or later (but not yet deployed) needs to be reworked on. However in the literature, the term "*Rework*" has not been uniformly and clearly defined as past practitioners and researchers considered terms like "reconsideration", "re-instantiation", "redoing" and "revision" as synonymous with rework, while the Oxford Dictionary defines "*Rework*" as "making changes to the original version of something". Depending on the complexity of the changes, the amount of rework required varies from some software module modifications to a non-trivial alteration to software design of a system. The cost associated with such a rework obviously will vary too.

According to our systematic review on Requirements Change Management (RCM) [4], there are three main components in managing requirements changes: change identification, change impact analysis and change cost/effort estimation. Effort estimation is about calculating and predicting the effort of a set of activities before they are actually performed [5, 6]; and effort is a value usually expressed in terms of time and/or dollars. Subsequently, change effort estimations are to predict the cost and time required for implementing a change. Such estimations are important as underestimation can result in budget overrun, poor quality and delay in project completion; whereas overestimation may result in the allocation of too many resources which will cause inefficiency [6]. Accurate estimation can also help assess the feasibility of implementing a change, prioritize the implementation of the requested changes and determine the cost of the implementation of a change.

Prior to conducting a change effort estimation, we need to have a better understanding of the extent to which and how a system would be reworked as it is possible to have more than one option for implementing a change and different options require different amounts of rework to be made to a system. In such a situation, estimation might need to be done for each option in order to determine its suitability. It should be noted that with the complexity of the changes requested and the number of implementation options available, change effort estimation can be a tedious and time-consuming task. It would therefore be beneficial to have a method which can identify the implementation option which involves the lesser amount of rework, before any estimation is carried out; and a lot of time will be saved by not having to conduct the unnecessary estimations. Given the importance of rework for estimation, the relationship between them is hardly understood.

In our systematic review [4], we explained how existing estimation methods and models can be applied to effort estimation related to implementing requirements changes and pointed out the fact that general effort estimation models may not be suitable for estimating the effort of implementing a requirements change. There are a few models that deal specifically with requirements change effort/cost estimation as discussed in the related work section of our systematic review paper [4]. Most existing methods use expert judgment which is based on the experience of the estimator which is not a consistent component and expert judgement also relies on past project data which may not be applicable to a particular case where there is no historical data. It is therefore important that the interrelations and dependencies between systems functions are identified for estimating the effort/cost of changes as the dependencies will have an impact on an implementation. An inherent drawback in most existing estimation methods is that these dependencies are not well understood [6].

To date, there is limited research on the concept of rework for software development and assessing rework for implementing requirements changes. In this paper, we first present our definition of rework and then describe a method of assessing rework for implementing software requirements changes in the context of its definition. Our method consists of three stages: (i) identification of the change; (ii) identification of the activities within the software system design which are affected by the change; and (iii) assessing the rework required. Stages 1 and 2 are based on the concepts and ideas described in our two previously published papers and the results of applying Stages 1 and 2 enable Stage 3 to be carried out. Stage 3 involves the computations of: (i) Interaction Comparison (IC); (ii) Interaction Weight (IW); and (iii) Rework which is based on IC and IW. To demonstrate the practicality that it enables developers to compare the rework between the different options available for implementing a requirements change and to identify the one which is less invasive and requires lesser amount of modifications to the software system design, we have applied our method to a running example for a reader's better comprehension of it.

2. The concept of rework and our proposed definition

The concept of rework exists in fields outside software development. In the field of medicine, doctors may need to rework treatment plans for patients who have developed unexpected reactions; in the building industry, civil engineers may need to rework plans for the load bearing of a bridge depending on future traffic conditions; academics will need to rework course and/or subject material depending on assessment outcomes or feedback by students. Several studies in civil engineering defined rework as "the unnecessary effort of redoing a process or activity that was incorrectly implemented the first time" [7, 8].

Rework is common in software development due to changes emanating from clients, development environment, and laws of the government and society. We discussed the causes of these changes extensively in our systematic review [4]. A key activity in RCM is to identify the amount of rework required for the proposed changes, as this will have a significant impact on the time and cost of a project. Studies show that normally rework leads to additional effort and cost [9-14] of a project. However, a clear relationship between rework and effort estimation has not been understood/established. Some studies proposed methods for reducing the amount of rework [10, 15], yet the fact remains that there will still be a considerable amount of rework to deal with. In the agile software development environment, it encourages rework instead of attempting to eliminate it [4, 16]. Rework is often unavoidable as the understanding of a problem and its possible solutions evolve over time.

Rework is a central activity in the development of software. The cost of rework is said to reach or even exceed 50% of the total project cost [9-11, 17]. These costs are one of the main concerns in software development since it is an important parameter defining the success of software projects [12, 13]. According to Charrette [14], software developers spend 40-50% of their time on rework activities. Based on the above facts, rework is generally considered as an important software development activity. In software development, Zhao and Osterweil [15] define rework as "the re-instantiation of tasks previously carried out in earlier development phases in a richer context that is provided by the activities and artifacts that had been performed and created during subsequent phases". In a simpler manner, Ghezzi et al. [17] suggest that rework consists of "going back to a previous phase" of software development to redo decisions made or work carried out in that previous phase". It is clear that the concept of rework has been subject to different interpretations. In short, rework has not been uniformly and well defined [10, 15, 18].

Based on the discussion above and our systematic review findings, we are of the opinion that the concept of rework needs to be more narrowly focussed on the following items:

- Requirements changes are the reasons for doing it;
- Instead of being broadly considered as a software development activity, it is one which falls in the area of RCM; and
- It is closely related to change cost/effort estimation which is also a RCM activity.

Our proposed definition of *rework* is therefore as follows:

"Rework in the field of software engineering is an activity within the area of Requirements Change Management (RCM), which makes modifications/alternations to a system which has a software design document and is being developed (pre-delivery) for implementing certain requirements changes, with the alternations/modifications normally introducing extra work and increasing the total amount of cost/effort for completing the software project; and assessing rework, a preliminary step to change cost/effort estimation which is another RCM activity, is about studying how a system needs to be modified/altered for implementing the changes."

According to this definition, we establish that rework is an activity conducted prior to the delivery of a system. Given that a software design document is necessary, rework assessment can be applied to any stage of software development as long as a software system design document is available; and it is independent of the type of software development methodology (be it waterfall or agile). With Agile Software Development, a design document becomes available as the development progresses and therefore rework assessment becomes plausible.

Another noteworthy point is that there is a key difference between our definition of rework and maintenance. According to IEEE standard 1219, software maintenance is defined as "the process of modifying a software system or component after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment"[19, 20]. The post-delivery nature of maintenance is also emphasised similarly in the ISO/IEC [ISO95] definition [20, 21]. The modifications to a system during the maintenance phase will always preserve the integrity of the software product [20, 22]. If the software design of a system needs to be altered substantially, the alternation will not be done as a piece of maintenance work but rework which will lead to new version of the software product. An example is that Microsoft usually release a newer version of its Windows operating system every period of say 3-4 years, or sometimes shorter.

3. Overview of the method of assessing rework

We anticipate that our method of assessing rework enable us to understand to what extent a system needs to be altered for implementing the required changes. Based on our previously developed methods of specification and classification [23, 24], we have identified that some requirements changes can be implemented in more than one way, which we refer to as change implementation possibilities/options. We aim to realize the following:

1. A numerical representation of the assessment of rework required to implement a requirement change for all possible implementation options.
2. Selection of the option which requires a lesser invasive to the software design of the system, ergo is of lesser rework.

3. Comparison of the assessment of rework between multiple requirements changes.

This method is a continuation of the findings of the specification and classifications methods [23, 24] and change analysis methods [25] previously established by the authors. The use of these methods in the rework method is detailed in Figure 1.

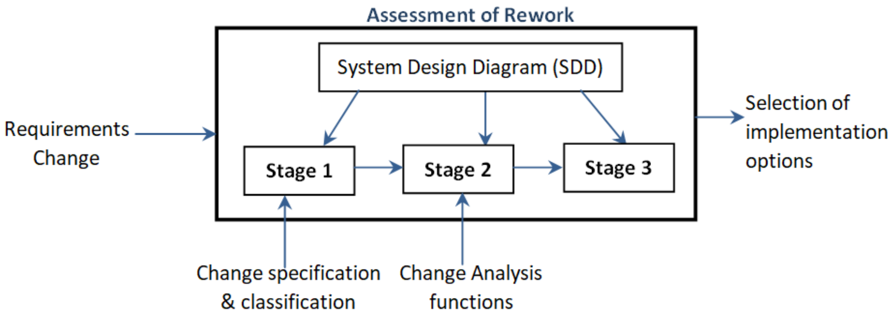

Assessment of Rework
System Design Diagram (SDD)
Requirements Change
Stage 1
Stage 2
Stage 3
Selection of implementation options
Change specification & classification
Change Analysis functions

Fig. 1. Overview of the method

According to Figure 1, the method will use as input the requirements changes and the system design diagram (SDD). The output of the method is executed in three stages:

Stage 1: Identification of the change

The change is identified and categorised using the methods which we have developed and reported in [23, 24].

Stage 2: Identification of the system activities affected due to the change

Once the change is identified, we apply the change analysis functions of the change analysis method which was developed and reported in [25]. As a result, the change is further exposed, enabling us to identify the activities that are directly affected by the change. Using the SDD, we then map the directly affected activities (DAA) to identify the indirectly affected activities (IdAA). The IdAA are the activities that are connected to DAA through input and/or output. IdAAs are considered in this assessment as modifications to a DAA which may have a direct impact on the activities associated with the DAA via the input-output links. The SDD can be any design diagram that shows the relationships between different objects and activities. Typically various forms of UML [26] diagrams such as activity diagrams, class diagrams, etc. can be used for this purpose.

Stage 3: Assessing the rework required

Once all the activities related to the change are identified, we can assess the rework. In order to do this, we adopt the methods that are introduced in [27-29]. From [27] and [28], the concept we adopt is referred to as interaction frequency. This frequency refers to the ratio of the number of interactions (input-output) performed by the affected operations (of a change) and the number of interactions performed by all operations of the interface. A similar concept is used in [29] where instead, the number of interfaces are used. Given that the interactions between the activities are identified and indicated in the

SDD, we can use this concept to assess the rework and thereafter make a selection of the implementation option with a lesser rework.

4. The details of the method

In this section, we describe in detail, the stages presented above.

4.1. Identification of the changes – Stage 1

To identify a requested requirements change, we use the change specification and classification methods which we have developed and reported in [23, 24]; and a summary of them can be found in Appendix 1. Change specification denotes a way of specifying a change so that communication ambiguities between business and IT staff can be avoided. Once a requirements change has been initiated from the client side, this method will use the SDD as input to map the location of the change. In order to create the specification template, we use two established methods, i.e. Goal Question Metrics (GQM) [30] and the Resource Description Framework (RDF) [31]. We also use a set of additional questions to enable better identification when using the specification template output.

The change classification method uses the outcomes of the specification template to expand on the type of change along with preliminary guidance on the action to be taken in managing the change. The classification itself is based on the concepts of the change taxonomy found in the existing change management literature [32-35] and is refined using the unstructured interviews of practitioners in the field of change management. The outcomes of the change classification will provide software developers with a better understanding of what the change is and offers preliminary guidance on how the change implementation can be carried out. The detailed change classification is shown in Table 1. The term link mentioned in Table 1 refers to the input-output connection between the activities. The term activity refers to the process activities in a SDD.

At implementation time, the key elements of the two methods (specification and classification) are incorporated into a single table (see Table 2). In the table, change number refers to the number given to each change as they are requested. The object, purpose and focus in Table 2 correspond to the specification method (please refer to Appendix 1).

Table 1. Detailed change description

Change focus	Answer to Additional Question	**Change type**	**Action**
Add	No	Matched links	Add new activity without changing the current activity or any connected links
	Yes	Mismatched links	Add new activity by changing the activity and/or connected links
Modification	No	Inner property modification	Modify the implementation of an activity without changing the connected links
	Yes	Input data modification	Modify the input link and internal properties of an activity
	Yes	Output data modification	Modify the output link and internal properties of an activity
Delete	No	Matched links	Delete activity without changing connected activities
	Yes	Mismatched links	Delete activity by changing connected activities and links
Activity Relocation	No	Relocation with matched links	Relocate existing activity without changing the activity or connected links
	Yes	Relocation with mismatched links	Relocate new activity by changing the activity and/or connected links

Object → the activity name according to the SDD (this is the activity affected by the change)
Purpose → the reason for the change
Focus → the activity selected from the list - Add, Delete, Modification or Activity relocation

Change type and action can be sourced from Table 1 based on the information provided for the object, focus and additional question, respectively. The option columns represent how each change may be described using different foci. This may not apply to all changes. This feature was added to the implementation template to provide more diversity and flexibility for communicating a change. Having multiple options also provides flexibility as to how the change can be implemented.

Table 2. Template for implementation

	Change No.	**Option 01**	**Option 02**	**Option n**
Specification Method	**Object**			
	Purpose			
	Focus			
	Additional Question			
	RESULT			
Classification Method	**Change type**			
	Action			

Running example

In order to explain this stage and the following stages, we will consider the following running example.

Diskwiz is a company which sells DVDs by mail order. Customer orders are received by the sales team, which checks that the customer details have been completed properly on the order form (for example, delivery address and method of payment). If they are not, a member of the sales team contacts the customer to obtain the correct details. Once the correct details are confirmed, the sales team passes a copy of the order to the warehouse team to pick and pack, and a copy to the Finance team to raise an invoice. Finance raises an invoice and sends it to the customer within 48 hours of the order being received. When a member of the warehouse team receives the order, they check the real-time inventory system to make sure the discs ordered are in stock. If they are, they are collected from the shelves, packed and sent to the customer within 48 hours of the order being received, so that the customer receives the goods at the same time as the invoice. If the goods are not in stock, the order is held in a pending file in the warehouse until the stock is replenished, whereupon the order is filled. This process is illustrated in the following system design diagram.

The change scenario:

The management is not satisfied with some parts of the process and points out that the following issue should be rectified: “It is identified, due to a design error, there is no communication between Finance and the Warehouse to confirm discs are in stock so that the order can be shipped. Therefore, Finance could be raising invoices when the order has not been sent.”

One of the reasons for having no communication between Finance and Warehouse is because there is no communication between A_4 and A_5, where A_4 represent one activity of the Warehouse and A_5 represents Finance. Another way to view this would be that there is no communication between A_5 and A_6, where A_6 is another activity of the Warehouse.

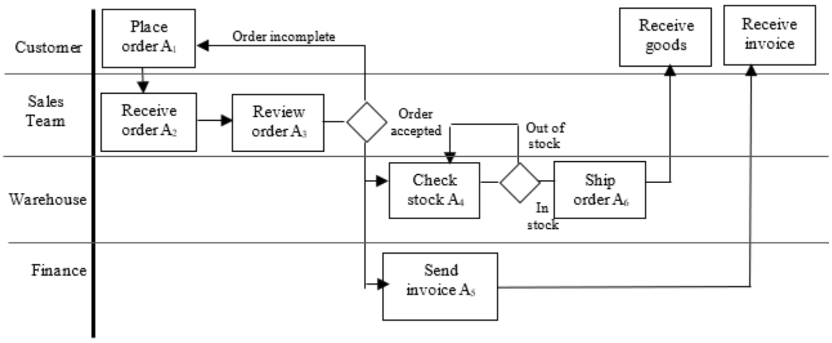


Fig. 2. Diskwiz customer order fulfilment process diagram

Therefore, the objects to be considered are A_4, A_5 and A_6. The purpose of this change is to resolve an existing design issue (according to the change scenario). A software engineer may use different focus based on the different combinations of objects which are A_4, A_5 and A_6 (based on the views taken in the above paragraph).

Table 3 is the complete application of Stage 1 of the rework method, i.e. this is a populated form of Table 2. If we consider option 01 in Table 3, the objects selected are A_4 and A_5, with a purpose of resolving a design error. Based on the rationale given between the non-existence of communication between the two objects, it would be feasible to add this communication between the objects. Therefore, the "Add" focus was chosen as the focus. Now this focus can be mapped to Table 1. From this point onwards, the rest of the fields in Table 2 will follow the directions related to the change focus "Add" in Table 1. The same rationale can be applied to options 02 and 03.

Table 3. Change classification outcome

Change 01	**Option 01**	**Option 02**	**Option 03**
Object	A_4 and A_5	A_4 and A_5	A_5 and A_6
Purpose	Resolution of design error	Resolution of design error	Resolution of design error
Focus	Add	Modify	Modify
Additional Question	Need addition input/output? Y	Input/output modification? Y	Input/output modification? Y
Result			
Change Type	Add new function between A_4 and A_5 (Mismatched links)	Inner property modification and output data modification A_4 and input data modification of A_5	Inner property modification and output data modification A_6 and input data modification of A_5
Action	Add new function by changing the function and/or connected links of A_4 & A_5	Modify A_4 to send message to A_5	Modify A_6 to send message to A_5

4.2. Identification of the system activities affected by the change(s) – Stage 2

We use a part of the change analysis method which we have developed and reported in [25] for expanding further the change identified; and a summary of this analysis method can be found in Appendix 2. Using this expansion, both DAAs and IdAAs are identified using the system design diagram. The change analysis functions are based on the change foci identified in [23, 24]: add, delete, modify and relocation. We use the category of primary change analysis functions to expand the changes. The category of primary functions can be used for building a block of more complex functions. The need to do this is due to the fact that it is hard to project every possible way of implementing the changes so practitioners can use this type of block to help them facilitate the changes.

The following terminologies are used for the functions:

The term activity in this method is used to represent process activities in the SDD.

A_N – New activity, A_O – Old activity, A_T - Target activity, Pt – Pointer, A_R – Relocating activity, A_C – Connected activity

V – Value: the value passed onto the function for data manipulation

L – Link: the connection between two activities

The primary category consists of the following set of functions:

1. Function to create a new activity
   CreateFunc(String, V) $\rightarrow A_N$
2. Function to link a new activity with existing activities
   CreateLink(A_N, A_O, V)
3. Function to link existing activities
   CreateLink($A_{X\text{-}O}$, $A_{Y\text{-}O}$, V)
4. Function to delete an activity
   DeleteFunc(A_O)
5. Function to delete links between activities
   DeleteLink($A_{X\text{-}O}$, $A_{Y\text{-}O}$)
6. Function to modify inner property of an activity
   ModifyInner(A_T,V)
7. Function to modify input data of an activity
   ModifyIn(A_S, A_T, V)
8. Function to modify output data of an activity
   ModifyOut(A_S, A_T, V)
9. Function to create a pointer to an existing activity
   CreatePointer(Pt, A_T)
10. Function to delete a pointer
    DeletePointer(Pt)

Once the change has been expanded, the activities identified in the functions are mapped to the SDD. These are the DAAs. In the SDD, any activity connected as the input and/or output of a DAA is considered an IdAA.

Running example stage 2

In accordance with this example and Table 3, the change can be implemented using one of the three options. In stage 2, we apply the preliminary functions from the change analysis method for the 3 options and we generate the following expansions of the change:

Table 4. Expansion of change options

Option 1	Option 2	Option 3

CreateFunc(String, V) →A_N
CreateLink(A_N, A_4, V)
{
ModifyInner(A_4,V)
ModifyIn(A_N, A_4, V)
ModifyOut(A_N, A_4, V)
}
CreateLink(A_N, A_5, V)
{
ModifyInner(A_5,V)
ModifyIn(A_N, A_5, V)
ModifyOut(A_N, A_5, V)
}
	ModifyInner(A_4,V)
	CreateLink(A_4, A_5, V)
	ModifyOut(A_4, A_5, V)
	ModifyIn(A_4, A_5, V)
		ModifyInner(A_6,V)
		CreateLink(A_5, A_6, V)
		ModifyOut(A_6, A_5, V)
		ModifyIn(A_6, A_5, V)

Based on Table 4, we are able to identify the DAAs for each option. Then by mapping the DAAs to the SDD, we are able to identify the IdAAs for each DAA. In this paper when selecting the IdAAs, we consider only the first impact level. Investigation of further levels can be considered as a future enhancement, which is outside the scope of this paper.

Table 5. Identification of DAAs and IdAAs

Options	DAAs	IdAAs
1	A_4	$A_{3,}$ A_6
	A_5	A_3
2	A_4	$A_{3,}$ A_6
	A_5	A_3
3	A_5	A_3
	A_6	A_4

4.3. Assessing the rework required – Stage 3

Through the numerical values generated, we are able to assess the rework to be carried out as a result of the change. In order to ensure the assessment of the rework is based on both the total interactions of the activities to be reworked as well as the difficulty level of implementing the change action, we use the number of affected interactions as well as the change weights introduced in the change analysis method [25]. The values for the weights are adopted from [36]. It has been established that in the change analysis method, each change action / type has a different difficulty level. Therefore, this difficulty level needs to be represented in the rework.

The assessment of the work required to implement a change involves the following calculations:

1. The interaction comparison (IC) of the affected activities (direct and indirect)

2. The interaction weight (IW) using the change weights of the affected activities (direct)
3. The rework based on IC and IW

As a result of the values generated from IC and IW, developers will have a numerical view of the assessment of the rework for implementing a change. If there are more than one option of implementation, then based on the combination of IC and IW, the developer can choose the lesser invasive option, which would result in the option with lesser rework.

When choosing the lesser invasive option, first preference is given to the lesser value of IC as this denotes lesser number of connections in the software design of the system will need to be altered. In the event that the IC value is the same for two or more options, IW will be considered. Use of IW is explained in the following sections.

Interaction comparison (IC) Calculation

Interaction comparison is the identification of the percentage of interactions that need to be altered in order to accomplish the required change. An interaction is a connection between two or more process activities (input-output links) in a SDD. This is in comparison to the total number of interactions identified in the SDD. Using the SDD, the following steps are used to calculate IC:

- For each activity (DAAs and IDAAs) involved in the change, identify the number of interactions. These interactions will be the number of connections each activity has with the other activities of the system.
- Identify the total number of interactions in the entire system.
- Calculate IC.

Running example stage 3 (IC calculation)

Using the above example, we show how the value of IC is calculated for all the options.

IC calculation for option 1:

The number of interactions for each identified activity based on Table 5 is as follows:
A_4 – has 2 interactions (Connected to A_3 and A_6)
A_5 – has 1 interaction (Connected to A_3)
A_3 – has 4 interactions (Connected to A_1, A_2, A_4 and A_5)
A_6 – has 1 interaction (Connected to A_4)

Considering all the interactions, the system design contains six activities. The interaction count for each activity is as follows:
A_1 – has 2 interactions (Connected to A_2 and A_3)
A_2 – has 2 interactions (Connected to A_1 and A_3)
A_3 – has 4 interactions (Connected to A_1, A_2, A_4 and A_5)
A_4 – has 2 interactions (Connected to A_3 and A_6)
A_5 – has 1 interaction (Connected to A_3)
A_6 – has 1 interaction (Connected to A_4)
The way of calculating the value of IC is adopted from [27].

$$IC_{CO} = \frac{N_I}{N_{TI}}$$

Formula 1: IC caclulation

Where CO is the Change Option number, N_I is the number of interactions per change action and N_{TI} is the total number of interactions for the system according to the SDD.

$$N_I = \sum_{x=1}^{n} N_{Ix}$$

Formula 2:No. of interactions affected by change

where x is the number of activities affected by the change action and N_{Ix} is the interactions for each affected activity.

$$N_{TI} = \sum_{x=1}^{n} N_{TIx}$$

Formula 3: Total no. of interactions in the system

where x is the total number of activities of the system and N_{TIx} is the interactions for each activity.

Applying to the example option 1:

When calculating N_I we consider the interaction of all the activities (DAAs and IdAAs) of option 1 which include: A_4, A_5, A_3 and A_6 (extracted from Table5). Based on the interactions identified for these activities, N_I is;

$$N_I = 2 + 1 + 4 + 1 = 8 \qquad (1)$$

When calculating N_{TI} interactions of all the activities are considered. Based on the interactions identified for all activities, N_{TI} is;

$$N_{TI} = 2+2+4+2+1+1 = 12 \qquad (2)$$

$$IC_1 = \frac{8}{12} = 67\% \qquad (3)$$

According to this value, when considering option 1 for change implementation, 67% of all the interactions have to be altered in order to implement the required change.

IC calculation for option 2:

The number of interactions for each identified activity based on Table 5 is as follows:
A_4 – has 2 interactions (Connected to A_3 and A_6)
A_5 – has 1 interaction (Connected to A_3)
A_3 – has 4 interactions (Connected to A_1, A_2, A_4 and A_5)
A_6 – has 1 interaction (Connected to A_4)

The total number of interactions is the same as that of option 1
Therefore;

$$IC_2 = \frac{N_I}{N_{TI}} \qquad (4)$$

Applying the same principles as option 1;

$$N_I = 2 + 1 + 4 + 1 = 8 \qquad (5)$$

$$N_{TI} = 2+2+4+2+1+1 = 12 \qquad (6)$$

$IC_2 = \frac{8}{12} = 67\%$

According to this value, when considering option 2 for change implementation, 67% of all the interactions have to be altered for implementing the required change.

IC calculation for option 3:

The number of interactions for each identified activity based on Table 5 is as follows:
A_5 – has 1 interaction (Connected to A_3)
A_6 – has 1 interaction (Connected to A_4)
A_4 – has 2 interactions (Connected to A_3 and A_6)

The total number of interactions is the same as that of option 1
Therefore;

$$IC_3 = \frac{N_I}{N_{TI}} \qquad (7)$$

Applying the same principles as option 1;

$$N_I = 1 + 1 + 2 = 4 \qquad (8)$$

$$N_{TI} = 2+2+4+2+1+1 = 12 \qquad (9)$$

$$IC_3 = \frac{4}{12} = 33\% \qquad (10)$$

According to this value, when considering option 3 for change implementation, 33% of all the interactions have to be altered for implementing the required change.

Interaction weight (IW) Calculation

The interaction weight is the change weight corresponding to the directly affected interactions due to the requirements change. The change weight concept was established in the change analysis method [25]. The weights for the change categories are assigned, using the principles described in [36] and [37] and based on the knowledge they have gained in working in the industry as well as extensive research on requirements change management. In both studies the change weights are incorporated in mathematical formulas which compute a change complexity. IW adds depth to the IC value by providing a numerical representation of the difficulty level of implementing the change and how this relates to the interactions. The value of IW becomes further important in assessment and selection, when the value for IC can be the same for different options of a given change, as we demonstrated in the running example. We establish that the lower the IW, the less difficult it would be to implement a change. In order to calculate IW, the following steps are used:

- Identify the change types using the expanded change action steps (Stage 2).
- Calculate the total change weight based on the change analysis method.
- Use the interactions and the total change weight to calculate IW.

In order to calculate IW, we consider only the activities directly affected by the change. This is because the identification of change types are acquired from stage 2 where it only contains DAAs.

From the change expansion in stage 2, we consider the change functions *Create*, *Modify* and *Delete* when calculating IW.

Running example stage 3 (IW calculation)

Using the same running example, we use the outcome of Table 4 to identify the change types as follows:

Table 6. Change weight identification

Option 1	Option 2	Option 3

CreateFunc(String, V) →A_N
CreateLink(A_N, A_4, V)
{
ModifyInner(A_4,V)
ModifyIn(A_N, A_4, V)
ModifyOut(A_N, A_4, V)
}
CreateLink(A_N, A_5, V)
{
ModifyInner(A_5,V)
ModifyIn(A_N, A_5, V)
ModifyOut(A_N, A_5, V)
}
	ModifyInner(A_4,V)
	CreateLink(A_4, A_5, V)
	ModifyOut(A_4, A_5, V)
	ModifyIn(A_4, A_5, V)
		ModifyInner(A_6,V)
		CreateLink(A_5, A_6, V)
		ModifyOut(A_6, A_5, V)
		ModifyIn(A_6, A_5, V)

Create Functions – 3 Modify Functions – 6 Delete Functions – 0	Create Functions – 1 Modify Functions – 3 Delete Functions – 0	Create Functions – 1 Modify Functions – 3 Delete Functions – 0

Using the weighting system introduced in the change analysis method, we develop Table 7 to calculate the change weight (CW):

- All create functions will have the Add weight of 3
- All modify functions will have the Modify weight of 2
- All delete functions will have the Delete weight of 1
- All other functions are a combination of the main three functions i.e. create, modify and delete

Table 7. Change weight calculation

Change Type	Option 1	Option 2	Option n
Add	No. of functions × CW Add	No. of functions × CW Add	…. × ….
Modify	No. of functions × CW Mod	No. of functions × CW Mod	…. × ….
Delete	No. of functions × CW Del	No. of functions × CW Del	…. × ….
Total CW			

Applying the findings of the running example of Table 6:

Table 8. Calculated change weights

Change Type	Option 1	Option 2	Option 3
Add	3 × 3 = 9	1 × 3 = 3	1 × 3 = 3
Modify	6 × 2 = 12	3 × 2 = 6	3 × 2 = 6
Delete	N/A	N/A	N/A
Total CW	21	9	9

$$IW_{CO} = \left(\sum_{X=1}^{n} N_{CO}\right) \times \sum CW_{CO}$$

Formula 4: IW calculation

where CO is the Change Option number and N_{CO} is the number of interactions per change action where only interactions of the DAAs are considered. We reiterate the reason for only considering DAAs is they are directly attached to the change actions (as seen in Table 4) and IdAAs are not. The number of interactions for the DAAs was identified when calculating the IC value. CW_{CO} is the total change weight for that option as shown in Table 8.

Applying the equation to the running example:

For option 1:
The directly affected activities are A_4 and A_5. Therefore,

$$N_1 = 2+1 \quad (11)$$

$$CW_1 = 21 \quad (12)$$

$$IW_1 = (2 + 1) \times 21 = 63 \quad (13)$$

For option 2:
The directly affected activities are A_4 and A_5. Therefore,

$$N_2 = 2+1 \quad (14)$$

$$CW_2 = 9 \quad (15)$$

$$IW_2 = (2 + 1) \times 9 = 27 \quad (16)$$

For option 3:
The directly affected activities are A_5 and A_6. Therefore,

$$N_3 = 1+1 \quad (17)$$

$$CW_3 = 9 \quad (18)$$

$$IW_3 = (1 + 1) \times 9 = 18 \quad (19)$$

Rework calculation based on IC and IW

In section IC Calculation, IC was established to be the percentage of interactions that need to be altered in order to facilitate the required change and in section Running example stage 3, IW was established to be the change weight corresponding to the directly affected interactions due to the requirements change. Based on these two values, the assessment of rework is a combined look at both the interactions that need to be altered in comparison to the full system depicted in the SDD and the difficulty of implementing the change action on those interactions. In order to display the comparison between the rework required for the changes requested and their multiple options, we use Table 9 as a template.

Table 9. Template of comparison between rework

	Change 1			Change 2	Change n
	Opt 1	Opt 2	Opt n		
IC					
IW					

Running example stage 3 (rework calculation)

To better understand this template, we populate it with the outcome of the running example:

Table 10. Outcome of comparison

	Change 1		
	Opt 1	Opt 2	Opt 3
IC	67%	67%	33%
IW	63	27	18

According to this example, one change was requested with three possible actions that can be taken to implement it. According to the above table, the value of IC is the same for options 1 and 2. Option 3 has a lower IC value than that of options 1 and 2. This is a good indication that option 3 is the lesser invasive option for implementing the change as a lesser number of interactions has to be altered. This fact is further validated by the IW value where option 3 has the lowest IW value corresponding to a lower difficulty level of implementing the change.

Based on the above results, it can be said that:

- option 1 and 2 require 67% of the interactions to be altered while option 3 requires only 33% alterations;
- based on IW, option 3 has a lesser difficulty level of implementation as compared to the other options; and
- therefore, the lesser invasive change implementation is option 3, based on both the IC and IW values.

5. Comparison with the related work

To the best of our knowledge, in the literature there has been no paper published on assessment of rework in the area of RCM. However, we are able to find two papers in the literature which focus on effort estimation related to implementation of requirements changes. Although these methods do not assess rework, they use requirements changes and their impact in the calculation process in a similar manner to our method. We shall discuss below a comparison with these two pieces of work.

Requirements changes can occur at any phase of the development process and even after deployment. There are few estimation methods dedicated to change effort/cost estimation and the importance of such methods were established in the introduction. The following discussion elaborates on two methods that deal specifically with change effort/cost estimation that use a similar rationale to the method introduced in this paper.

The estimation method introduced by Jeziorek [29] attempts to estimate the cost of the impact of a design change to development. The author emphasises the importance of identifying the functional requirements and design parameters that are impacted by the change, before attempting to estimate the cost of change. He uses this identification in the form of a matrix to detect the physical interactions between components. These physical interactions are used to determine how the change propagates through the system. The model developed in [29] outputs the affected components, how they are affected and what the cost of impact will be. In this particular method, the use of interactions between components and the mapping of the propagation of the change through the system are similar activities as used in our method.

In the method established by Lavazza and Valetto [38], several different artifacts are used to calculate the change costs. The key feature of this method is the use of requirements instead of lines of code to calculate the cost. Therefore, the method utilizes the design document and traceability techniques for estimation. The estimation is carried out in two stages: 1) characteristics such as the size and the complexity of the code are estimated on the basis of the size of the complexity of the requirements and the skill and experience of the implementation team; 2) effort is estimated based on the knowledge of the relations that link the inputs, outputs and the resources required. Most parts of the estimation are based on historic data. The use of requirements to establish the complexity and the linking of inputs and outputs resonate with the rework method introduced in this paper.

We use the aforementioned work to describe the limitations of the existing work and compare our methods to define what has been achieved. The limitations focus only on the techniques comparable with our method.

Table 11. Comparison with related work

Technique	**Limitations**	**What our method addresses**
Jeziorek [29]	Initially, a lot of time needs to be spent in developing the matrices needed to identify the impact. These matrices are non-transferable and therefore for every project, new matrices need to be established.	New diagrams are not needed. The method uses the system diagram which a software project would usually have.
Lavazza and Valetto [38]	The use of historical data which may not be available for some projects and is therefore limited to systems development that has such data. The use of traceability methods that have inherent limitations such as informal development methods, insufficient resources, time and cost for traceability, lack of coordination between people responsible for different traceable artifacts, imbalance between benefits obtained and effort spent implementing traceability practices, and construction and maintenance of a traceability scheme proves to be costly [39-46]	The method uses data only from the current project. The change identification and analysis techniques used in this method do not use traceability techniques and therefore do not have the drawbacks associated with traceability techniques.

6. Conclusions and future work

In this paper, we have presented a definition of rework – "*Rework in the field of software engineering is an activity within the area of Requirements Change Management (RCM), which makes modifications/alternations to a system which has a software design document and is being developed (pre-delivery) for implementing certain requirements changes, with the alternations/modifications normally introducing extra work and increasing the total amount of cost/effort for completing the software project; and assessing rework, a preliminary step to change cost/effort estimation which is another RCM activity, is about studying how a system needs to be modified/altered for implementing the changes.*" We have also described a method of assessing rework for implementing software requirements changes. Once a change has been proposed, our method identifies the paths of implementation, which lead to the identification of the impacted activities of the system through the SDD. Using these activities, two values (IC and IW) are computed to help assess the rework required for all the possible options. Based on the IC and IW values, a developer can choose the lesser invasive option which requires lesser rework.

To demonstrate the viability of our method, we have applied it to the Diskwiz customer order fulfilment process as a running example. For the requested requirements change, we generated multiple implementation options and for each option, IC and IW were calculated. We have shown that when multiple options of implementation exist for one change, IC alone is not sufficient to make an assessment and selection. In the example, the change resulted in two options, which have the same IC value for implementations. In such scenarios, IW plays an important role in the assessment process. Based on the values of IC and IW, the rework was assessed, and comparisons were then made between the implementation options of a change and we were able to identify which option requires a lesser amount of rework.

The results of applying our method to this running example indicates that it is useful in the area of RCM. It enables developers to have a better understanding of the rework required by different options for implementing change. Given the fact that the implementation path is extracted from the SDD, our method can be applied during any phase of the software development, provided that the design document is available.

We can thus conclude that it can serve as a precursor to change effort estimation, whereby it is not necessary to carry out estimation for all the possible implementation options but the one which has been assessed to involve a lesser amount of rework. Hence, a related future work would be to develop a change effort method for estimating the time and the cost required for implementing a change.

References

1. B. W. Boehm, "Software engineering economics," in *Pioneers and Their Contributions to Software Engineering*: Springer, 2001, pp. 99-150.
2. S. Ferreira, J. Collofello, D. Shunk, G. Mackulak, and P. Wolfe, "Utilization of process modeling and simulation in understanding the effects of requirements volatility in software development," in *International Workshop on Software Process Simulation and Modeling, Portland, Oregon*, 2003.
3. D. Pfahl and K. Lebsanft, "Using simulation to analyse the impact of software requirement volatility on project performance," *Information and Software Technology,* vol. 42, no. 14, pp. 1001-1008, 2000.
4. S. Jayatilleke and R. Lai, "A systematic review on Requirement Change Management," *Information and Software Technology,* vol. 93, pp. 163-185, 2018. DOI: 10.1016/j.infsof.2017.09.004., doi: 10.1016/j.infsof.2017.09.004.
5. D. Kiritsis, K.-P. Neuendorf, and P. Xirouchakis, "Petri net techniques for process planning cost estimation," *Advances in Engineering Software,* vol. 30, no. 6, pp. 375-387, 1999.
6. H. Leung and Z. Fan, "Software cost estimation," *Handbook of Software Engineering, Hong Kong Polytechnic University,* pp. 1-14, 2002.
7. P. E. D. Love, D. J. Edwards, H. Watson, and P. Davis, "Rework in Civil Infrastructure Projects: Determination of Cost Predictors," *Journal of Construction Engineering and Management,* vol. 136, no. 3, pp. 275-282, 2010, doi: doi:10.1061/(ASCE)CO.1943-7862.0000136.
8. P. E. D. Love, "Influence of Project Type and Procurement Method on Rework Costs in Building Construction Projects," *Journal of Construction Engineering and Management,* vol. 128, no. 1, pp. 18-29, 2002, doi: doi:10.1061/(ASCE)0733-9364(2002)128:1(18).
9. K. Butler and W. Lipke, "Software process achievement at tinker air force base," Technical Report CMU/SEI-2000-TR-014, Carnegie-Mellon Software Engineering Institute (September 2000), 2000.

10. A. G. Cass, S. M. Sutton, and L. J. Osterweil, "Formalizing rework in software processes," in EWSPT, 2003, vol. 2786: Springer, pp. 16-31.
11. F. CeBASE eWorkshop, "Focusing on the cost and effort due to software defects," NSF Center for Empirically Based Software Engineering, 2001.
12. V. R. Basili, S. E. Condon, K. E. Emam, R. B. Hendrick, and W. Melo, "Characterizing and modeling the cost of rework in a library of reusable software components," presented at the Proceedings of the 19th international conference on Software engineering, Boston, Massachusetts, USA, 1997.
13. U. T. Raja, M.J., "Defining and Evaluating a Measure of Open Source Project Survivability," IEEE Transactions on Software Engineering, vol. 38, no. 1, pp. 169-174, 2012.
14. R. N. Charette, "Why software fails [software failure]," IEEE Spectrum, vol. 42, no. 9, pp. 42-49, 2005.
15. X. O. Zhao, L.J., "An approach to modeling and supporting the rework process in refactoring," in International Conference on Software and System Process (ICSSP), 2012, pp. 110-119.
16. J. Highsmith and A. Cockburn, "Agile software development: The business of innovation," Computer, vol. 34, no. 9, pp. 120-127, 2001.
17. C. Ghezzi, M. Jazayeri, and D. Mandrioli, Fundamentals of software engineering. Prentice Hall PTR, 2002.
18. P. E. Love and J. Smith, "Benchmarking, benchaction, and benchlearning: rework mitigation in projects," Journal of Management in Engineering, vol. 19, no. 4, pp. 147-159, 2003.
19. J. Radatz, A. Geraci, and F. Katki, "IEEE standard glossary of software engineering terminology," IEEE Std, vol. 610121990, no. 121990, p. 3, 1990.
20. K. H. Bennett and V. T. Rajlich, "Software maintenance and evolution: a roadmap," in Proceedings of the Conference on the Future of Software Engineering, 2000: ACM, pp. 73-87.
21. ISO12207 Information technology - Software life cycle processes, I. I. S. Organisation, Geneva, Switzerland, 1995.
22. G. Canfora and A. Cimitile, "Software maintenance," in Handbook of Software Engineering and Knowledge Engineering: Volume I: Fundamentals: World Scientific, 2001, pp. 91-120.
23. S. Jayatilleke and R. Lai, "A method of specifying and classifying requirements change," in Software Engineering Conference (ASWEC), 2013 22nd Australian, 2013: IEEE, pp. 175-180.
24. S. Jayatilleke, R. Lai, and K. Reed, "Managing Software Requirements Changes through Change Specification and Classification," Computer Science and Information Systems, vol. 15, no. 2, pp. 321-346, 2018, doi: 10.2298/CSIS161130041J.
25. S. Jayatilleke, R. Lai, and K. Reed, "A method of requirements change analysis," Requirements Engineering, pp. 1-16, 2017. DOI: 10.1007/s00766-017-0277-7., doi: 10.1007/s00766-017-0277-7.
26. P. Selonen, K. Koskimies, and M. Sakkinen, "Transformations between UML diagrams," Journal of Database Management, vol. 14, no. 3, p. 37, 2003.
27. T. Wijayasiriwardhane and R. Lai, "Component Point: A system-level size measure for component-based software systems," Journal of Systems and Software, vol. 83, no. 12, pp. 2456-2470, 2010.
28. S. Mahmood and R. Lai, "A complexity measure for UML component-based system specification," Software: Practice and Experience, vol. 38, no. 2, pp. 117-134, 2008.
29. P. N. Jeziorek, "Cost estimation of functional and physical changes made to complex systems," Massachusetts Institute of Technology, 2005.
30. R. Van Solingen, V. Basili, G. Caldiera, and H. D. Rombach, "Goal question metric (gqm) approach," Encyclopedia of Software Engineering, 2002.
31. M. Weiss, "Resource description framework," in Encyclopedia of Database Systems: Springer, 2009, pp. 2423-2425.

32. N. Nurmuliani, D. Zowghi, and S. P. Williams, "Requirements volatility and its impact on change effort: Evidence-based research in software development projects," in Proceedings of the Eleventh Australian Workshop on Requirements Engineering, 2006.
33. S. McGee and D. Greer, "A software requirements change source taxonomy," in Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on, 2009: IEEE, pp. 51-58.
34. N. Nurmuliani, D. Zowghi, and S. P. Williams, "Using card sorting technique to classify requirements change," in Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International, 2004: IEEE, pp. 240-248.
35. H. Xiao, J. Quo, and Y. Zou, "Supporting change impact analysis for service oriented business applications," in Systems Development in SOA Environments, 2007. SDSOA'07: ICSE Workshops 2007. International Workshop on, 2007: IEEE, pp. 6-6.
36. Y. Li, J. Li, Y. Yang, and M. Li, "Requirement-centric traceability for change impact analysis: a case study," in Making Globally Distributed Software Development a Success Story: Springer, 2008, pp. 100-111.
37. S. Maadawy and A. Salah, "Measuring Change Complexity from Requirements: A Proposed Methodology," ed: IMACST, 2012.
38. L. Lavazza and G. Valetto, "Requirements-based estimation of change costs," Empirical Software Engineering, vol. 5, no. 3, pp. 229-243, 2000.
39. J. Cleland-Huang, C. K. Chang, and M. Christensen, "Event-based traceability for managing evolutionary change," Software Engineering, IEEE Transactions on, vol. 29, no. 9, pp. 796-810, 2003, doi: 10.1109/TSE.2003.1232285.
40. D. Zowghi and R. Offen, "A logical framework for modeling and reasoning about the evolution of requirements," in Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on, 1997: IEEE, pp. 247-257.
41. R. Sugden and M. Strens, "Strategies, tactics and methods for handling change," in Engineering of Computer-Based Systems, 1996. Proceedings., IEEE Symposium and Workshop on, 1996: IEEE, pp. 457-463.
42. M. Strens and R. Sugden, "Change analysis: a step towards meeting the challenge of changing requirements," in Engineering of Computer-Based Systems, 1996. Proceedings., IEEE Symposium and Workshop on, 1996: IEEE, pp. 278-283.
43. O. C. Gotel and A. C. Finkelstein, "An analysis of the requirements traceability problem," in Requirements Engineering, 1994., Proceedings of the First International Conference on, 1994: IEEE, pp. 94-101.
44. R. Torkar, T. Gorschek, R. Feldt, M. Svahnberg, U. A. Raja, and K. Kamran, "Requirements traceability: a systematic review and industry case study," International Journal of Software Engineering and Knowledge Engineering, vol. 22, no. 03, pp. 385-433, 2012.
45. J. Cleland-Huang, R. Settimi, C. Duan, and X. Zou, "Utilizing supporting evidence to improve dynamic requirements traceability," in Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on, 2005: IEEE, pp. 135-144.
46. M. Heindl and S. Biffl, "A case study on value-based requirements tracing," in Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering, 2005: ACM, pp. 60-69.

Shalinka Jayatilleke holds a BSc (Hons) from Institute of Technological Studies (Affiliated to Troy University, USA), a MSc from Sri Lanka Institute of Information Technology and a PhD (in computer science) from La Trobe University, Australia. She is currently a lecturer at La Trobe University with an academic career, which started in 2004. Her current research interests are requirements engineering, change management, digital disruption and learning analytics.

Richard Lai holds a BE (Hons) and a MEngSc from the University of New South Wales and a PhD from La Trobe University, Australia. He has spent about 10 years in the computer industry prior to joining La Trobe University in1989. His current research interests include component-based software system, software measurement, requirements engineering, and global software development.

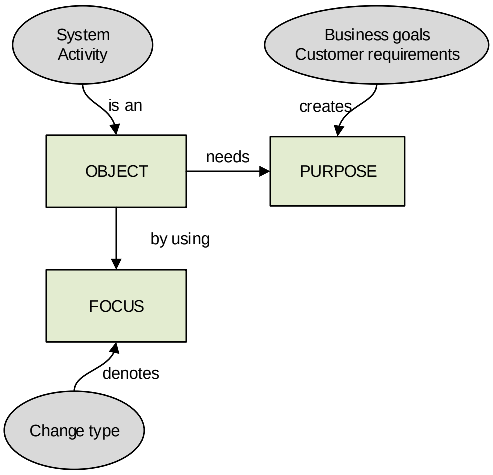
Received: February 21, 2020; Accepted: August 28, 2020

Appendix 1

The change specification method (Refer reference no. 24):

The specification method is made up of GQM and RDF. The GQM-RDF combination is a result of amalgamating ontology and terminology which in this paper, we refer to as onto-terminology. The method has both linguistic and logical principles. To ensure the correct combination of logic and terminology, we have selected two well-known methods where GQM represents terminology and the other RDF ontology. Three terms are extracted from GQM that can best describe a requirement change; Object, Purpose and Focus (of change). The terms extracted from RDF are Object, Attribute and Value, which is referred to as the RDF triplet. The logical relationship of the RDF triplet can be stated as Object O has an Attribute A with a Value V (Professor; Reads; a Book). The rationale behind the correspondence between RDF triplet and to the GQM terms is due to the similarity and the meanings of the terms, which is described in table below.

RDF term	GQM term	Correspondence	Rationale
Object	*Object*	One-to-one	Same concept
Attribute	*Purpose*	One-to-one	Both terms are activities. *Purpose* is an activity that is generated due to various business requirements.
Value	*Focus*	One-to-one	*Value* of RDF creates the significance for *Attribute* (of RDF). *Focus* of GQM creates the significance for *Object* (of GQM) by activating the term *Purpose* of GQM.



Onto-terminology Framework

The template designed for the change specification based on the framework above is given in the table below. By selecting the object of change using the system design diagram, designers and decision makers can accurately locate the main target of change, resulting in a clarification of the location of change. Knowing the reason for the change through the purpose ensures that change implementers are able to clarify the need for the change. The focus of change acts as advice on the basic implementation needed to execute the change, resulting in the clarification of the action of change. It indicates to the designers what to do instead of how to do the change. We believe that clearly describing the location, need and action of a change request using this template will resolve much of the existing miscommunication issues.

	Description
OBJECT	The activity name according to the system design diagram
PURPOSE	The reason for the change (can be descriptive)
FOCUS	Select from Add, Delete, Modify or Activity Relocation

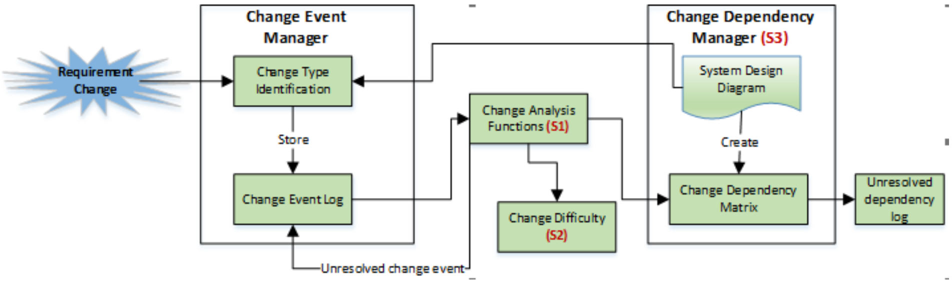
The change classification method (Refer reference no. 24):

The main purpose of change classification method is to ensure that change implementers are able to identify and understand unambiguously the requirement change. The classification is based on previous literature on the same and unstructured interviews of 15 practitioners in the field of change management. The result of this investigation is given in section 4.1 Table 1.

Appendix 2

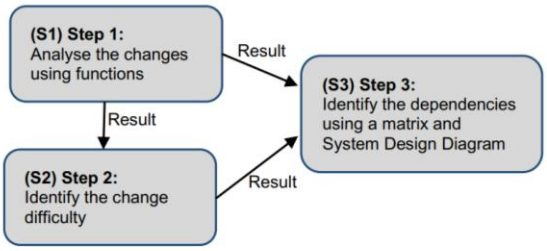
The Method of Requirements Change Analysis (Refer reference no. 25)

The method consists of three steps: namely, (1) analyzing the change using functions, (2) identifying the change difficulty; and (3) identifying the dependencies using a matrix. We have used step 1 in the rework method introduced in this paper.



Change analysis method

Once a change has been identified through the Change Event Manager (CEM), the method follows three steps:


(S1) Step 1:
Analyse the changes using functions
Result
Result
(S2) Step 2:
Identify the change difficulty
Result
(S3) Step 3:
Identify the dependencies using a matrix and System Design Diagram

Three step analysis process

- Step 1 (S1) is for expanding the identified changes and for discovering the more detailed information for the implementation as a result of the changes. As shown in Figure 2, the two categories of change analysis functions (herein after referred to as functions) described in section 3.2 are employed for carrying out this step.
- Step 2 (S2) identifies the difficulty of implementing the change. The result of this will be used later for assigning a priority to each of the requested changes.
- Step 3 (S3) identifies the conflicts and/or dependencies between the required changes. As shown in in Figure 2, the key elements involved are the Change Dependency Matrix (CDM) and the System Design Diagram (SDD). The conflicts and/or dependencies between the changes are identified once the changes have been mapped to the matrix.

Computer Science and Information Systems 18(1):155–168 https://doi.org/10.2298/CSIS200515038G

Double-Layer Affective Visual Question Answering Network

Zihan Guo[1], Dezhi Han[1], Francisco Isidro Massetto[2], and Kuan-Ching Li[3]*

[1] College of Information Engineering, Shanghai Maritime University
Shanghai 201306, China
guo_zihan11@163.com, dzhan@shmtu.edu.cn
[2] Center for Cognition and Complex Systems, Universidade Federal do ABC (UFABC)
Santo André, SP 09210-580, Brazil
francisco.massetto@ufabc.edu.br
[3] Dept. of Computer Science and Information Engineering, Providence University
Taichung 43301, Taiwan
kuancli@pu.edu.tw

Abstract. Visual Question Answering (VQA) has attracted much attention recently in both natural language processing and computer vision communities, as it offers insight into the relationships between two relevant sources of information. Tremendous advances are seen in the field of VQA due to the success of deep learning. Based upon advances and improvements, the Affective Visual Question Answering Network (AVQAN) enriches the understanding and analysis of VQA models by making use of the emotional information contained in the images to produce sensitive answers, while maintaining the same level of accuracy as ordinary VQA baseline models. It is a reasonably new task to integrate the emotional information contained in the images into VQA. However, it is challenging to separate question-guided-attention from mood-guided-attention due to the concatenation of the question words and the mood labels in AVQAN. Also, it is believed that this type of concatenation is harmful to the performance of the model. To mitigate such an effect, we propose the Double-Layer Affective Visual Question Answering Network (DAVQAN) that divides the task of generating emotional answers in VQA into two simpler subtasks: the generation of non-emotional responses and the production of mood labels, and two independent layers are utilized to tackle these subtasks. Comparative experimentation conducted on a preprocessed dataset to performance comparison shows that the overall performance of DAVQAN is 7.6% higher than AVQAN, demonstrating the effectiveness of the proposed model. We also introduce more advanced word embedding method and more fine-grained image feature extractor into AVQAN and DAVQAN to further improve their performance and obtain better results than their original models, which proves that VQA integrated with affective computing can improve the performance of the whole model by improving these two modules just like the general VQA.

Keywords. deep learning, natural language processing, computer vision, visual question answering, affective computing.

* Corresponding author

1. Introduction

In recent years, multimodal learning for natural language processing (NLP) and Computer Vision (CV) has gained broad interest, such as Visual Question Answering (VQA) [1], image captioning [41] and image-text matching [10], among several others [24]. Compared to other multimodal learning tasks, VQA is more challenging, since it requires a fine-grained understanding of both textual questions and visual images, and it may also involve complex reasoning and require common sense knowledge to answer the questions correctly. Therefore, VQA is regarded as a test of the deep visual and textual understanding ability of a model, as well as a benchmark for general artificial intelligence (AI). An instance of VQA consists of typical tasks that connect an image and a related question, so the task of the machine is to produce the correct answer. There are many potential applications for VQA, such as a personal assistant or robotics designed to assist individuals with physical disabilities.

In early VQA models, the conventional approach is to train a deep neural network with supervision, which maps the given question and the given image to a relative scoring of candidate answers. Specifically, the input question is first tokenized into words, so then the model utilizes the word embedding method to transform the terms into single vectors. Next, the model inputs the word vectors into a Recurrent Neural Network (RNN) to obtain the question features and inputs the given image into a Convolutional Neural Network (CNN) pre-trained on object recognition to capture the image features. Finally, the model fuses the question features and the image features through linear pooling (such as element-wise multiplication) and then feeds the joint embedding into a classification layer to predict the correct answer. With the emergence of advanced word embedding methods, fine-grained feature extractors, cognitive fusion mechanisms and various attention mechanisms, the performance of VQA models is also improved.

It is noteworthy that most of the existing VQA models do not further understand nor analyze the emotional information contained in the input images. Part of the reason is due to the fact that, there is no VQA dataset that includes rich emotional information to the images it contains so far. Till recently, the Affective Visual Question Answering Network (AVQAN) [34] enriches the model's understanding and analysis of VQA by making use of the emotional information contained in the images to produce sensitive answers, while maintaining the same level of accuracy as ordinary VQA baseline models. It is a reasonably new task to integrate the emotional information contained in the images into VQA.

However, it is challenging to separate question-guided-attention from mood-guided-attention in AVQAN, due to the concatenation of question words and mood labels. It is believed that this type of concatenation is hazardous to the performance of the model. To mitigate this effect, we propose the Double-Layer Affective Visual Question Answering Network (DAVQAN), which divides the task of generating emotional answers in VQA into two relatively simple subtasks, i.e., the generation of non-emotional responses and the production of mood labels, and utilizing two independent layers to tackle the two subtasks respectively. In such studies, the emotional information contained in the images refers to human facial expressions. Since there is no publicly available dataset suitable for VQA integrated with affective computing, we use the same method as AVQAN to construct a preprocessed dataset to complete the proposed research. We conduct a comparative experiment on the preprocessed dataset to compare the performance of AVQAN and

DAVQAN, and the experimental results show that the overall performance of DAVQAN is 7.6% higher than that of AVQAN, showing the effectiveness of the proposed model. We also introduce more advanced word embedding method and more fine-grained image feature extractor into AVQAN and DAVQAN to further improve their performance and obtain better results than their original models, what shows that VQA integrated with affective computing can improve the performance of the entire model by improving these two modules just like the general VQA.

The remainder of this article is organized as follows. Section 2 reviews the works related to VQA, while in section 3 is provided the details of DAVQAN. Next, details on how we construct the preprocessed dataset for the experiments and experimental evaluation are presented in Section 4, and finally, we present the conclusions and future directions of this work in Section 5.

2. Related work

Text-based Question Answering. Text-based question answering is a longstanding problem that has been studied for decades in natural language processing. The model needs to fully understand the textual questions and requires a wide range of knowledge to answer the questions correctly [26] [23]. The early text-based question answering system [39] uses information retrieval to find out the text containing the answer as the output of the model. Recently, advanced methods, e.g. [3], have improved the accuracy of their answers by constructing large-scale knowledge bases. Through all the efforts, text-based question answering has been successfully applied to search engines, mobile devices, and other fields. Various methods and models for text-based question answering inspire VQA techniques. Nevertheless, unlike text-based question answering, VQA is naturally grounded in images – requiring the understanding of both visual images and textual questions. The information contained in the visual images is more abundant and noisier than that contained in the textual questions. Therefore, VQA is more challenging to deal with than text-based question answering. Meanwhile, the interactions between the visual images and the textual questions are also essential to VQA. Furthermore, the questions are generated by humans, making the need for complex reasoning and common sense knowledge more essential.

Describing Visual Content. For many years, many researches have been devoted to the study of joint learning which combines the visual and textual information [4], [5], [30], [32], [37], [46], [40], [44]. Related to VQA are the tasks of image tagging [21], [15], video captioning [33], [13] and image captioning [22], [8], [27], [41], where the models are used to generate sentences or words to describe visual content. Automatically describing the content of an image is a fundamental problem of artificial intelligence [16]. In the early stages, the researches on describing visual content mainly included the object classification task and the task of assigning descriptions. While these tasks require both semantic and visual knowledge, captions can often be non-specific. Even the more advanced methods and models for generating generic image captions are of little use for VQA, since the questions in VQA require detailed specific information about the images. Therefore, compared with captioning tasks, VQA is more complex, more interactive, and has a broader range of applications.

Visual Question Answering. In the past few years, VQA has attracted more and more attention. The first VQA dataset developed as a benchmark is Data Set for Question Answering on Real World (DAQUAR) [25]. With the continuous development, the most popular modern datasets use images sourced from Microsoft Common Objects in Context (COCO) [38], a dataset initially designed for image recognition. Those images constitute a diverse collection of photographs. Some of the latest versions of these datasets, such as VQA v2.0 [11], have been proposed to address issues of dataset biases and other issues. Based on these datasets, many models have been proposed to deal with VQA tasks. Most of these models learn the joint embedding of the image features and the question features and then input them into a classification layer to predict the correct answer.

From the above description, we can see that in most VQA models, the first step is to use the word embedding method to transform the question words into single vectors. Initially, common word embedding methods included the one-hot representation of words and the GloVe word embeddings [28] pre-trained on a large-scale corpus. ELMo [29], a later proposed method, improves the performance of word embedding methods by concatenating the left-to-right and the right-to-left word features extracted from the text. However, models like ELMo are feature-based and not profoundly bidirectional. At present, the more advanced method BERT [7] can pre-train a deep bidirectional Transformer and can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of natural languages processing tasks, such as word embedding and sentence classification.

After transforming the question words into single vectors, the VQA models need to extract the question features and the image features. The Long Short-Term Memory (LSTM) [14] and the Gated Recurrent Unit [17] are the most common methods to extract question features. And most of the original VQA models use the VGGNet [36] to extract image features. Now the more advanced methods for extracting image features are the ResNet [18] and the bottom-up attention network [2] derived from Faster R-CNN [35]. The question features and the image features obtained by early VQA models are at the global-level and contain noisy information. In many cases, keywords of the questions and the local areas of the images are the key to answer the questions correctly. As a result, various attention mechanisms have been proposed and have become an integral part of VQA models (e.g., [2]). The core idea of attention mechanism is to assign different weights to local features so that the model can focus on the essential local features rather than the global features. Furthermore, the multimodal feature fusion mechanisms of question features and image features are also fundamental to VQA models because of the requirements of the models for understanding and analyzing the content of the input questions and the input images and the relationships between them. The element-wise addition and the element-wise multiplication are the earliest multimodal feature fusion mechanisms used for VQA. To obtain higher-level interactions between question features and image features, several methods based on bilinear pooling have been proposed, such as MCB [9]. With the development of the above technologies, the performance of VQA models also improved.

3. Double-Layer Affective Visual Question Answering Network

Although AVQAN enriches the model's understanding and analysis of VQA, it is difficult for AVQAN to separate question-guided-attention from mood-guided-attention due to the concatenation of the question words and the mood labels. Different from AVQAN, DAVQAN divides the task of generating emotional answers into two relatively simple subtasks, i.e., the generation of non-emotional answers and the generation of mood labels, and uses two independent layers to tackle the two subtasks respectively. The non-emotional layer takes the images, and the questions as input to predict non-emotional answers, and the emotional layer deals with the emotional information contained in the input images to predict mood labels for the images. Finally, we combine the non-emotional answers and the mood labels to compose the emotional answers. In this section, we first introduce the non-emotional layer. The emotional layer will be detailed in the second part.

3.1. Non-emotional layer

In the early stages, the image features used in VQA models were global features and contained irrelevant and noisy information. In many cases, the local areas of the image are the key to answer the question correctly. Thus, the attention mechanisms based on visual attention were proposed and have become an integral part of VQA models. With the development of attention mechanisms, researchers have successfully proposed the co-attention mechanisms [6], [45] that can focus on both the keywords of the questions and the local areas of the images to improve the performance of VQA models. For a fair comparison, DAVQAN uses the same attention mechanism as AVQAN, that is, we use the input questions to guide the model to focus on the local areas of the input images.

We introduce the spatial attention [12], [42] into the standard LSTM to construct our non-emotional layer. The non-emotional layer takes the images and the questions as input to predict the non-emotional answers. It learns to attend to the pertinent regions of the input image as it reads the input question tokens in a sequence. Specifically, the input textual question $Q = (q_1, q_2, \dots, q_n)$ is first tokenized into words and these words are then transformed into one-hot representations by function $OH(\cdot)$. And the input visual image I is represented as a set of regional image features extracted from a pre-trained CNN model. Now there are many advanced image feature extractors such as the ResNet [18] and the bottom-up attention network [2] derived from Faster R-CNN [35]. For fair comparison, we choose the same image feature extractor as AVQAN, i.e., the VGGNet [36]. In the experimental part, we also replace the VGGNet with ResNet to extract more fine-grained image features to further improve the performance of the models.

The embeddings of the image and the question tokens can be given as follows:

$$v_0 = W_i[F(I)] + b_i\,. \tag{1}$$

$$v_i = W_w[OH(t_i)], i = 1, ..., n\,. \tag{2}$$

where $F(\cdot)$ represents the CNN extractor which transforms the visual image I from the pixel space to a 4096-dimensional feature representation. The W_i matrix and the W_w matrix are used to embed the image feature and the question word embeddings into the same dimension. Thus, we can concatenate the image feature and the question word embeddings and input them into the LSTM model one by one to infuse our attention mechanism. In AVQAN, the embedding of the mood label is also added to the concatenation. We think that in this kind of concatenation, the question-guided-attention and the mood-guided-attention will interfere with each other. The update rules of our non-emotional layer can be defined as follows:

$$\mathrm{i}_t = \sigma(W_{vi}v_t + W_{hi}\mathrm{h}_{t-1} + W_{ri}\mathrm{r}_t + b_i)\,. \tag{3}$$

$$\mathrm{f}_t = \sigma(W_{vf}v_t + W_{hf}\mathrm{h}_{t-1} + W_{rf}\mathrm{r}_t + b_f)\,. \tag{4}$$

$$\mathrm{o}_t = \sigma(W_{vo}v_t + W_{ho}\mathrm{h}_{t-1} + W_{ro}\mathrm{r}_t + b_o)\,. \tag{5}$$

$$\mathrm{g}_t = \tanh(W_{vg}v_t + W_{hg}\mathrm{h}_{t-1} + W_{rg}\mathrm{r}_t + b_g)\,. \tag{6}$$

$$\mathrm{c}_t = \mathrm{f}_t \odot \mathrm{c}_{t-1} + \mathrm{i}_t \odot \mathrm{g}_t\,. \tag{7}$$

$$\mathrm{h}_t = \mathrm{o}_t \odot \tanh(c_t)\,. \tag{8}$$

where $\sigma(\cdot)$ represents the sigmoid function and $\bigodot$ is the element-wise multiplication operator. The convolutional features and the previous hidden state determine the attention term $\mathbf{r}_t$, which is the weighted average of the convolutional features and can be calculated by the following formula:

$$\mathrm{e}_t = w_a^T \tanh(W_{he}\mathrm{h}_{t-1} + W_{ce}C(I)) + b_a\,. \tag{9}$$

$$\mathrm{a}_t = \mathrm{softmax}(\mathrm{e}_t)\,. \tag{10}$$

$$\mathrm{r}_t = \mathrm{a}_t^T C(I)\,. \tag{11}$$

where the pre-trained model VGGNet extracts the 14×14 512-dimensional convolutional image features which are represented by *C(I)*, $\mathbf{e}_t$ represents the embedding of the previous hidden state $\mathbf{h}_{t-1}$, and $\mathbf{a}_t$ stands for a 196-dimensional vector of the image attention weights. The dimension of the regional image features is 512 and each image has 14×14 regions. All the weight matrices *W*s, biases *b*s and the attention terms in our non-emotional layer are learnable parameters. Finally, we relay the last LSTM hidden state to the Softmax classifier to predict the non-emotional answers. Figure 1 shows the structure of our non-emotional layer.

3.2. Emotional layer

Recent works have studied on utilizing CNN for visual attribute detection. In this paper, we follow the study from [31] to build our emotional layer. The emotional layer takes the visual images as input to predict mood labels for the images. In our settings, there are two tasks: the prediction of non-emotional answers and the prediction of mood labels. Both of these tasks share the same lower layers of the pre-trained CNN model VGGNet. The pre-trained CNN model VGGNet takes a square pixel RGB image as input and is composed of five successive convolutional layers C1... C5. After C5, there are three fully connected layers FC6... FC8. These three fully connected layers compute $\mathbf{Y}_6$=$\sigma(W_6\mathbf{Y}_5 + B_6)$, $\mathbf{Y}_7$=$\sigma(W_7\mathbf{Y}_6 + B_7)$ and $\mathbf{Y}_8$=$\psi(W_8\mathbf{Y}_7 + B_8)$, where $\mathbf{Y}_k$ denotes the output of the *k*-th layer, W_k, B_k are the learnable parameters of the *k*-th layer, and $\sigma(\mathbf{X})[i]$=max$(0, \mathbf{X}[i])$ and $\psi(\mathbf{X})[i]$=$e^{\mathbf{X}[i]}/\Sigma_j e^{\mathbf{X}[j]}$ are the "ReLU" and "SoftMax" non-linear activation functions.

Although the emotional layer and the pre-trained CNN model VGGNet are both designed to tackle the image classification task, the object labels of the two are quite different. To solve this problem, we remove the output layer FC8 of the VGGNet and add an adaptation layer formed by two fully connected layers FCA and FCB. FCA and FCB take

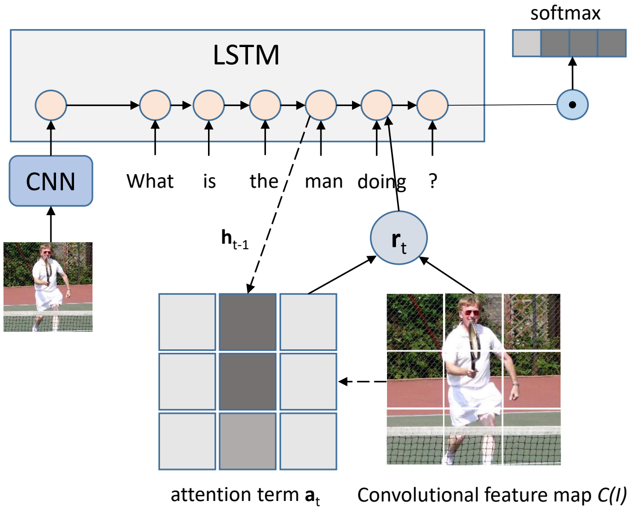


Fig. 1. The structure of the non-emotional layer

the output vector $\mathbf{Y}_7$ of the layer FC7 as input to predict a mood label for the given image. The calculation formula is as follows:

$$\mathrm{Y}_a = \sigma(W_a \mathrm{Y}_7 + B_a)\,. \tag{12}$$

$$\mathrm{Y}_b = \psi(W_b \mathrm{Y}_\mathrm{a} + B_b)\,. \tag{13}$$

where W_a, B_a, W_b, B_b are learnable parameters. In our emotional layer, FC6 and FC7 have the same size 4096, FCA has size 2048 and FCB has a size equal to the number of mood categories.

The layers C1. . . C5, FC6, and FC7 are pre-trained on the ImageNet and then transferred to our mood classification task and kept fixed. The two fully connected layers FCA and FCB are trained on the preprocessed dataset. Figure 2 shows the architecture of the emotional layer.

4. Experiments and results

In this section, we will describe how we construct the preprocessed dataset for our experiments and perform the experimental evaluation.

4.1. The preprocessed dataset

Since there is no publicly available dataset suitable for VQA integrated with affective computing, we use the same method as AVQAN to construct a preprocessed dataset, which is composed of images of people, questions, answers and mood labels, based on the

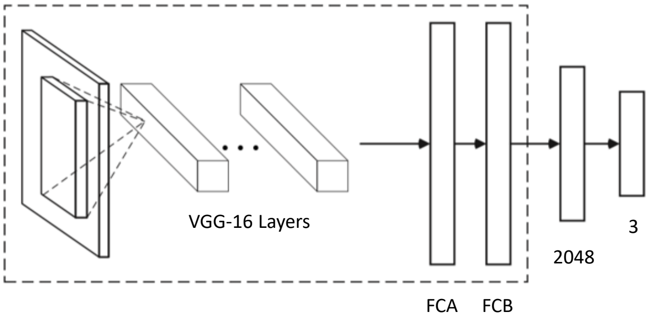


Fig. 2. Deep architecture of the emotional layer

Visual7w dataset [43] to complete our research. The Visual7w dataset is a subset of the Visual Genome QA dataset [20], which is one of the largest datasets designed for VQA with 1.7 million question/answer pairs. Besides, the Visual7w dataset uses the seven questions (What, Where, When, Who, Why, How and Which) to systematically check the visual and textual comprehension capabilities of a model. Note that, the 7th Which question category is used to extend existing VQA setups to accommodate visual answers, which is irrelevant to our study.

Specifically, we remove the images that are irrelevant to our task from the Visual7w dataset, leaving only the images bearing at least one person, and label each image with a mood label. It is worth noting that the Visual7w dataset is not a dataset dedicated to mood classification tasks, and many images of it contain little or no emotional information. Thus, we only obtain a limited set of samples. Considering that the corpus of the question words is tiny, we set the questions in our preprocessed dataset relatively simple to prevent the accuracy of the models from being too low. The 3 mood labels used are happy, surprised, and neutral, for there are too few samples of other mood labels such as sad. Among the total number of instances in the preprocessed dataset, 50% for training, 20% for validation, and 30% for testing. The ratios remain as they are in the AVQAN paper to ensure a fair comparison.

4.2. Experiment setup

During the experiment, the non-emotional layer takes the visual images, and the textual questions as input to predict non-emotional answers, and the emotional layer deals with the emotional information contained in the input images to predict mood labels for the images. If the mood label of an image is neutral, the emotional aspect is ignored in the answer. We use backpropagation to train our model and choose cross-entropy as our loss function. During validating, we use the validating split of the preprocessed dataset for hyper-parameter selection and early stopping. During testing, the model takes the visual images and textual questions as input, and we say the model is correct on a question if it manages to output the correct mood label and the correct non-emotional answer. The dimensions of the LSTM gates and memory cells are 512 in all the experiments, and the model is trained with Adam update rule [19]. In this paper, we evaluate the generated answers in the open-ended setting. An alternative method to evaluate is to let the model pick the correct mood label and the correct non-emotional answer among the candidates.

4.3. Answer categories

The answers generated by our proposed DAVQAN model can be classified as partially wrong (A: having a wrong mood but the rest of the answer is correct or B: having a correct mood but the rest of the answer is wrong), C: completely wrong, or D: completely correct. Figure 3 shows several examples of the four categories and Table 1 shows the accuracy of the four categories during testing.

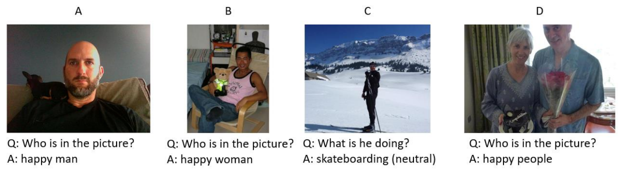

A
B
C
D
Q: Who is in the picture?
A: happy man
Q: Who is in the picture?
A: happy woman
Q: What is he doing?
A: skateboarding (neutral)
Q: Who is in the picture?
A: happy people

Fig. 3. The four answer categories from DAVQAN

Indeed, visual attribute detection is one of the most challenging problems in computer vision. As shown in Table 1, the performance of the emotional layer is not satisfactory. Although there are only three types of mood labels, the accuracy can only reach 79.89%. We think the reason that limits the performance of the emotional layer is that the preprocessed dataset is not large enough. In future studies, we will construct a larger and more suitable dataset for VQA integrated with affective computing to give full play to the advantages of deep learning.

4.4. Comparison with original AVQAN

Table 1. Analyzing the overall percentage of answers in each category for the DAVQAN model

category	accuracy
A	12.50%
B	24.46%
C	7.61%
D	55.43%

Nelson et al. [34] indicate that the integration of affective computing in AVQAN has no significant impact on the performance of ordinary VQA baseline models but rather enriches the model's understanding and analysis of images. In AVQAN, however, it is difficult to separate question-guided-attention from mood-guided-attention due to the concatenation of the question words and the mood labels. To solve this problem, we propose DAVQAN and conduct a comparative experiment on our preprocessed dataset to compare the performance of AVQAN and DAVQAN. To carry out the comparative experiment, we set the emotional answers as supervision for AVQAN, and the non-emotional answers and the mood labels as supervision for DAVQAN. To compare their structures fairly, the two models not only use the same word embedding method, the same feature extractors and the same attention mechanism, but also have the same parameter settings. Table 2 shows the results of AVQAN and DAVQAN on our preprocessed dataset. As observed in Table 2, the overall performance of DAVQAN is 7.6% higher than AVQAN, which shows the effectiveness of the proposed model. Due to the size limitation of the preprocessed dataset, the corpus of question words is very limited. The question-guided-attention is not effective enough to help the model answer questions correctly. Thus, we only count the overall performance of the two models, and the accuracy of AVQAN is slightly poorer than in Nelson et al. [34]. Besides, the imbalance of the preprocessed dataset may be another factor. In future researches, we will expand the size of the dataset and add more emotional information to the images of it to better train and evaluate the VQA models integrated with affective computing.

Table 2. The results of AVQAN and DAVQAN on our preprocessed dataset

Model	**Accuracy**
AVQAN	47.83%
DAVQAN	55.43%

4.5. AVQAN and DAVQAN with GloVe

Feature representation plays an important role in improving VQA performance. The AVQAN and DAVQAN described above use the one-hot representation of words to embed the question words. Now there are more advanced methods for word embedding, such as GloVe [28], ELMo [29] and BERT [7]. In order to study whether the more advanced word embedding methods can improve the performance of the two models, we use the GloVe word embeddings to replace the one-hot representation of the question words and carry out experimental verification. GloVe is a global log-bilinear regression model for the unsupervised learning of word representations, which can directly obtain the global corpus statistics. Instead of using individual context windows in a large corpus and the entire sparse matrix, the GloVe model uses the nonzero elements in a word-word co-occurrence matrix to train and construct a vector space with meaningful sub-structure thus efficiently leverages statistical information.

Specifically, instead of using the one-hot encoding, we use the 300-D GloVe word embeddings pre-trained on a large-scale corpus to transform the question words into 300-dimensional word vectors. The following operations are the same as those in the original AVQAN and DAVQAN models, that is, we embed the question word vectors and the image features into the same dimension and then take them as input to the LSTM model to complete the subsequent experiments. Table 3 shows the results of AVQAN and DAVQAN with GloVe. By comparing with Table 2, we can see that the accuracy of AVQAN and DAVQAN models after using GloVe is improved by 3.8% and 2.72% respectively, which proves that the improvement on the word embedding method can improve the performance of VQA models integrated with affective computing.

Table 3. The results of AVQAN and DAVQAN with GloVe on our preprocessed dataset

Model	**Accuracy**
AVQAN with GloVe	51.63%
DAVQAN with GloVe	58.15%

4.6. AVQAN and DAVQAN with ResNet

The AVQAN and DAVQAN described above use the VGGNet [36] to extract image features. Now there are more advanced image feature extractors, such as the ResNet [18] and the bottom-up attention network [2]. In this section, we use the ResNet to replace the VGGNet to extract more fine-grained image features to study whether the more advanced image feature extractors can improve the performance of the two models. The depth of image representation is crucial to many visual tasks, but the deeper neural networks are more difficult to train. The ResNet uses a residual learning framework to simplify the training of networks that are substantially deeper than those used previously. Instead of learning unreferenced functions, the ResNet explicitly reformulates the layers as learning residual functions with reference to the layer inputs. Empirical evidence shows that these residual networks are easier to optimize and can obtain accuracy from significantly increased depth to produce better results than previous networks.

The original AVQAN and DAVQAN models use VGGNet to extract image features to infuse attention mechanism and complete mood detection. For simple and convincing comparison, we only use the ResNet to replace the VGGNet used in the mood detector to complete our experiments. The rest of the two models are the same as the corresponding original model and the results are shown in Table 4. By comparing with Table 2, we can see that the accuracy of AVQAN and DAVQAN models after using ResNet-50 is improved by 5.97% and 1.09% respectively, which proves that better image feature extractors can improve the performance of VQA models integrated with affective computing. We have also explored the deeper network ResNet-101 and use it to improve the accuracy of AVQAN model to 54.35%. For DAVQAN, using ResNet-50 and ResNet-101 yielded similar results.

5. Conclusion and future work

The Affective Visual Question Answering Network (AVQAN) enriches the model's understanding and analysis of VQA by making use of the emotional information contained in the images while maintaining the same level of accuracy as ordinary VQA baseline

Table 4. The results of AVQAN and DAVQAN with ResNet-50 on our preprocessed dataset

Model	**Accuracy**
AVQAN with ResNet-50	53.80%
DAVQAN with ResNet-50	56.52%

models. It is a fairly new task to integrate the emotional information contained in the images into VQA. In AVQAN, however, it is difficult to separate question-guided-attention from mood-guided-attention due to the concatenation of the question words and the mood labels. We think that this kind of concatenation harms the performance of the model. To mitigate this effect, we propose the Double-Layer Affective Visual Question Answering Network (DAVQAN), which divides the task of generating emotional answers in VQA into two relatively simple subtasks, i.e., the generation of non-emotional answers and the generation of mood labels, and uses two independent layers to tackle the two subtasks respectively. Although the word embedding method, the feature extractors, and the attention mechanism used by the two models are the same, the overall performance of DAVQAN is 7.6% higher than that of AVQAN. We also introduce more advanced word embedding method and more fine-grained image feature extractor into AVQAN and DAVQAN to further improve their performance and obtain better results than their original models, which proves that VQA integrated with affective computing can improve the performance of the whole model by improving these two modules just like the general VQA. Furthermore, the performance of the models is limited because the dataset used is not large enough to give full play to the advantages of deep learning, and the emotional information contained in the images of the dataset is not rich enough. In future work, we will construct a larger, more specialized, and more balanced dataset to promote VQA tasks integrated with affective computing.

References

1. Agrawal, A., Jiasen, L., Antol, S., Mitchell, M., Zitnick, C.L., Batra, D., Parikh, D.: Vqa: Visual question answering. ICCV (2015)
2. Anderson, P., Xiaodong, H., Buehler, C., Teney, D., Johnson, M., Gould, S., Lei, Z.: Bottom-up and top-down attention for image captioning and visual question answering. CVPR (2018)
3. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. IJCAI pp. 2670–2676 (2007)
4. Barnard, K., Duygulu, P., Forsyth, D., Freitas, N.D., Blei, D.M., Jordan, M.I.: Matching words and pictures. The Journal of Machine Learning Research pp. 1107–1135 (2003)
5. Chen, K., Dahua, L., Bansal, M., Urtasun, R., Fidler, S.: What are you talking about? text-to-image coreference. CVPR pp. 3558–3565 (2014)
6. Chowdhury, M.I.H., Sridharan, S., Fookes, C., Nguyen, K.: Hierarchical relational attention for video question answering. ICIP (2018)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018)
8. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: Generating sentences for images. ECCV pp. 15–29 (2010)
9. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. EMNLP (2016)

10. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. ECCV pp. 241–257 (2016)
11. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. CVPR (2016)
12. Gregor, K., Danihelka, I., Graves, A., Rezende, D.J., Wierstra, D.: Draw: A recurrent neural network for image generation. ICML 37, 1462–1471 (2015)
13. Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., Saenko, K.: Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zeroshot recognition. ICCV pp. 2712–2719 (2013)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation (1997)
15. Jia, D., Berg, A.C., Fei-Fei, L.: Hierarchical semantic indexing for large scale image retrieval. CVPR pp. 785–792 (2011)
16. Jiang, Y., Liang, W., Tang, J., Zhou, H., Li, K.C., Gaudiot, J.L.: A novel data representation framework based on nonnegative manifold regularisation. Connection Science (forthcoming)
17. Junyoung, C., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS (2014)
18. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015)
19. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. The 3rd International Conference for Learning Representations (2015)
20. Krishna, R., Yuke, Z., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li-Jia, L., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. CoRR abs/1602.07332 (2016)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. NIPS 1, 1097–1105 (2012)
22. Kulkarni, G., Premraj, V., Dhar, S., Siming, L., Yejin, C., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating simple image descriptions. CVPR pp. 1601–1608 (2011)
23. Li, K., Jiang, H., Zomaya, A.: Big data: Management and processing. Taylor & Francis (2017)
24. Li, K., Martino, B.D., Yang, L.T., Zhang, Q.: Smart data: State-of-the-art perspectives in computing and applications. Taylor & Francis (2019)
25. Malinowski, M., Fritz, M.: A multi-world approach to question answering about real-world scenes based on uncertain input. NIPS 1, 1682–1690 (2014)
26. Martino, B.D., Li, K., Yang, L., Esposito, A.: Internet of everything: Algorithms, methodologies, technologies and perspectives. Springer (2018)
27. Mitchell, M., Dodge, J., et al., A.G.: Midge: Generating image descriptions from computer vision detections. EACL pp. 747–756 (2012)
28. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. EMNLP pp. 1532–1543 (2014)
29. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. NAACL (2018)
30. Pirsiavash, H., Vondrick, C., Torralba, A.: Inferring the why in images. CoRR abs/1406.5472 (2014)
31. Quanzeng, Y., Hailin, J., Jiebo, L.: Visual sentiment analysis by attending on local image regions. AAAI (2017)
32. Ramanathan, V., Joulin, A., Liang, P., Fei-Fei, L.: Linking people with ”their” names using coreference resolution. ECCV pp. 95–110 (2014)
33. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. ICCV pp. 433–440 (2013)
34. Ruwa, N., Qirong, M., Liangjun, W., Ming, D.: Affective visual question answering network. MIPR (2018)
35. Shaoqing, R., Kaiming, H., Girshick, R., Jian, S.: Faster r-cnn: Towards real-time object detection with region proposal networks. NIPS 1, 91–99 (2015)

36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2014)
37. Socher, R., Karpathy, A., Le, Q.V., Manning, C.D., Ng., A.Y.: Grounded compositional semantics for finding and describing images with sentences. Transactions of the Association for Computational Linguistics (2014)
38. Tsung-Yi, L., Maire, M., et al., S.B.: Microsoft coco: Common objects in context. ECCV (2014)
39. Voorhees, E.M., Tice, D.M.: Building a question answering test collection. SIGIR pp. 200–207 (2000)
40. 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 (forthcoming)
41. Xinlei, C., Hao, F., Tsung-Yi, L., Vedantam, R., Gupta, S., Dollar, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. CoRR abs/1504.00325 (2015)
42. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. ICML (2015)
43. Yuke, Z., Groth, O., Bernstein, M., Fei-Fei, L.: Visual7w: Grounded question answering in images. CVPR (2016)
44. 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 (forthcoming)
45. Zhou, Y., Jun, Y., Chenchao, X., Jianping, F., Dacheng, T.: Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. IEEE Transactions on Neural Networks and Learning Systems pp. 5947–5959 (2018)
46. Zitnick, C.L., Parikh, D., Vanderwende, L.: Learning the visual interpretation of sentences. ICCV pp. 1681–1688 (2013)

Zihan Guo is currently pursuing the Ph.D. degree in Shanghai Maritime University. His research interests include computer vision and natural language processing methods related to visual question answering.

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 network security, cloud computing, mobile networking, wireless communication, and cloud security.

Kuan-Ching Li is a Professor in the Dept of Computer Science and Information Engineering (CSIE) at Providence University, Taiwan, where he also serves as the Director of the High-Performance Computing and Networking Center. He published more than 300 scientific papers and articles and is co-author or co-editor of more than 25 books published by Taylor & Francis, Springer, and McGraw-Hill. Professor Li is the Editor in Chief of the Connection Science and serves as an associate editor for several leading journals. Also, he has been actively involved in many major conferences and workshops in program/general/steering conference chairman positions and has organized numerous conferences related to computational science and engineering. He is a Fellow of IET and a senior member of the IEEE. His research interests include parallel and distributed computing, Big Data, and emerging technologies.

Received: May 15, 2020; Accepted: September 10, 2020.

Computer Science and Information Systems 18(1):169–188 https://doi.org/10.2298/CSIS200317034R

Spatio-temporal Summarized Visualization of SmartX Multi-View Visibility in Cloud-native Edge Boxes

Muhammad Ahmad Rathore[1] and JongWon Kim[1]

[1] School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
Room No. 207, EECS Building C, Gwangju Institute of Science and Technology, 123 Cheomdangwagi-ro, Buk-gu, Gwangju 61005, South Korea
ahmadrathore@gist.ac.kr
[2] jongwon@gist.ac.kr

Abstract. The existing data summarization (and archival) techniques are generic and are not designed to leverage the unique characteristics of the spatio-temporal visualization at multi-resource level. In this paper, we propose and explore a family of data summaries that take advantage of the multiple layers i.e. physical/virtual resources with temporal and spatial correlation among distributed edge boxes. Significant challenges in measuring spatio-temporal data, however, contribute to both a tendency towards identifying efficient metrics with summarizing functionalities and effective verification methods. In this paper, we present our idea of maintaining summarized spatio-temporal data and verify through visualization of gathered operational data.

Keywords: spatio-temporal, edge, visualization, cloud-native.

1. Introduction

With the continuous development and popularization of IoT devices, high-performance computing, and storage technology, a type of time series data with space information, called spatio-temporal data, has emerged. This spatio-temporal data consists of useful and meaningful information that needs to be automatically mined, leading to the continuous development of spatio-temporal data processing technologies [1]. Maintaining such data has hard limits on service quality constraints that must be maintained, despite network fluctuations and varying peaks of load [2]. Furthermore, equipping cloud-native distributed data platform for edge devices with effective spatio-temporal query processing capability, well established in a centralized database is, therefore, a challenge for smooth operations.

Aligned with Future Internet testbeds, we launched “OF@TEIN+: Open/Federated Playgrounds for Future Networks” in 2017, to further enhance, extend and expand OF@TEIN collaboration [3]. OF@TEIN+ multi-site cloud (denoted as OF@TEIN+ Playground) connects around 10 international sites in 10 countries (Korea, Malaysia, Thailand, Indonesia, Laos, Cambodia, Vietnam, Myanmar, Bhutan, and India) spread across 14 research institutes and interconnected via OF@TEIN+ network. As depicted in Fig 1, to automatically build, operate, and utilize OF@TEIN+ Playground resources, we deployed Playground Tower as a logical space in a centralized location. It systematically covers various functional requirements of operating multi-site Playground by employing Provisioning / Visibility / Orchestration Centers. For example, the Visibility Center covers playground visibility and provides visualization support. The underlay network in the playground spread across many research and educational networks (REN) under distinct administrative domain. Previously from late 2013, a hyper-converged SmartX Box [4] was introduced to virtualize and merge the functionalities of four devices (i.e. Management Worker node, Capsulator node, OpenFlow switch, and Remote power device.) into a single box. Multiple SmartX Boxes are deployed and interconnected through SDN in the distributed environments. In OF@TEIN+ Playground, a multi-site affordable cloud-native version of the SmartX Playground [5] is established that consists of software-defined (i.e., composable) Playground with hyper-converged Box-style resources [6] named *"SmartX Micro-Box"*. These Boxes consist of interfaces for management/control and data, IoT devices, and remote power control. The SmartX Micro-Box is configured to support Cloud-native computing having a software stack that is microservices oriented with virtualized/-containerized IoT-SDN-Cloud functionalities. In addition, capabilities of edge computing in these resources enable technologies that allow computation to be performed at the network edge near data sources.

Identifying and maintaining collected temporal knowledge of distributed resources through monitoring and visualization are important operational activities to ensure smooth operations of the OF@TEIN+ Playground. However, burdens of large volume generated by collecting spatio-temporal multi-layer visibility over distributed multi-site cloud with scalable edge devices at regular intervals and transferred to a centralized cloud, lead to inefficient utilization of bandwidth, storage, and computing resources. To effectively operate OF@TEIN+ Playground, it is truly requisite to recognize how physical, virtual, and container resources are running in a steady-state over a concerned time period. Spatio-temporal knowledge is useful to analyze and troubleshoot server and network issues before they affect the developers. Lately, we have been developing tools for monitoring and measurement of OF@TEIN+ Playground resources enabled with cloud-native edge computing. Leveraging the 'SmartX Multi-View Visibility Framework (MVF)' we can monitor various dynamic visibility metrics from multiple measurement points across both physical and virtualized resources and associated flows of the playground [7]. The time-series multi-view metrics collected over multi-layer resources termed as *"Spatio-temporal visibility"* delivers a rich understanding of operations that can provide deep insights into current operations as well as the possibility of supporting continuous, agile delivery of applications to end-users.

In this paper, we propose a data spatio-temporal visibility technique that introduces a synchronized and timely collection of monitoring metrics over physical, virtual, and flow layers from distributed multi-site cloud-native edge Boxes.

- First, we summarized a multi-view visibility data scheme at a centralized location collected persistently over six months. By keeping useful visibility data without hurting the time-line we summarize and align temporal multi-view visibility in a much smaller size, i.e., the summarized visibility is 10% of original data.
- Second, based on a large-scale multi-view visibility data collected from the OF@TEIN+ Playground, our proposed approach can reduce the storage requirements up to 80% while maintaining high accuracy (with bounded-errors) on the query results.

- Third, we evaluate the multi-layer spatio-temporal data through effective visualization schemes, i.e. as multi-parameter, single site, and multi-site summarization view. In addition, analysis schemes are employed to extract useful patterns and trends from visibility data.

The remainder of this paper is organized as follows: Section II elaborates on the requirements together with the overall approach and challenges entailed by the inclusion of visibility for cloud-native edge Boxes. The design and implementation are described in Section III. Verification results are detailed in Section IV. Finally, we conclude the paper in the conclusion section.

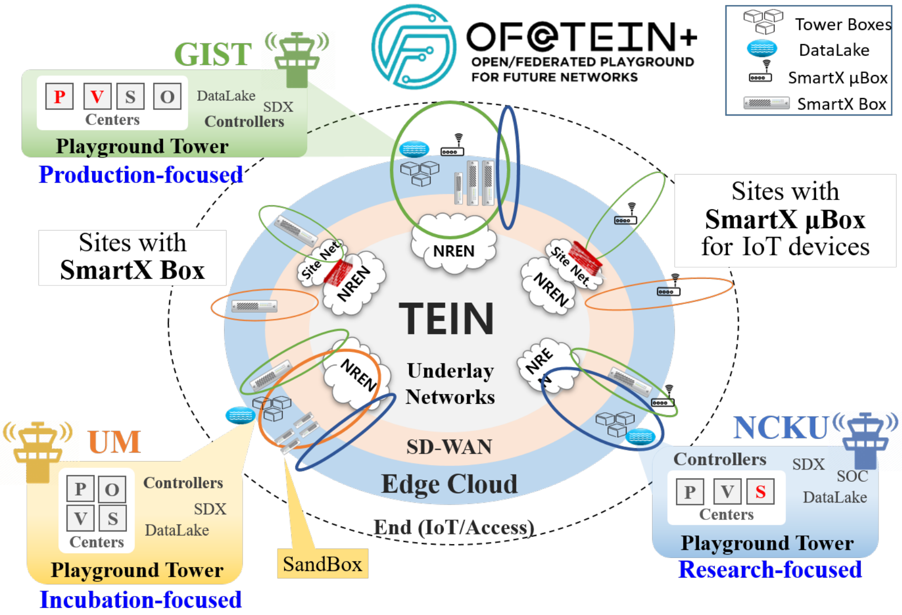


Fig. 1. OF@TEIN+ Playground as multi-site clouds.

2. Background and Requirements of Spatio-temporal Visibility

In this section, we briefly highlight the related work. Next, we mention the concept of the OF@TEIN+ Playground. In the end, we mention the requirements for spatio-temporal summarized visibility at distributed (multi-site) edge clouds by leveraging SmartX MVF.

2.1. Related Work

One of the most important development is handling a time series data based on various concepts such as visibility [8][9], correlations [10][11], recurrence analysis [12], phase-space reconstructions [13] and transition probabilities [14]. Studies have shown that researchers can summarize the characteristics of a time series into compact metrics, which can then be used to understand the dynamics or learn how the system will evolve with time. [15]. This data summarization can be employed to find a compact representation of a dataset [16]. It is important for data compression as well as for making pattern analysis more convenient. Summarization can be done on classical data, spatial data, as well as spatio-temporal data [17]. Spatio-temporal data contains the timestamp of a spatial object, a raster layer as well as the duration of a process. Spatio-temporal summarization is often performed after or in conjunction with spatio-temporal partitioning so that objects in each partition can be summarized by aggregated statistics or representative objects.

Automated identification of interesting patterns is developed in time series [18]. A clustering technique is adopted to summarize and refine the description of silent features and their relations. Ahmed et al. analyze non-stationary, volatile, and high-frequency time series data to present summarization [19]. Multi-scale wavelet analysis is employed for separating the trend, cyclical fluctuations, and auto-correlation effects. It is useful to summarize the data about the ‘chief features’ of the data.

A solution for flexible co-programming architecture is offered to support the life cycle of time-critical cloud-native applications [2]. Similarly, mobility-driven cloud-fog-edge offered collaborative real-time framework solution, which has IoT, Edge, Fog and Cloud layers [20].

In this research, multi-view visibility data is collected leveraging SmartX MVF. However, SmartX MVF has not been tested in an operational environment as time-series data. SmartX MVF was designed with minimal consideration given at the requirements needed to conduct spatio-temporal data mining research including space-time summarization. In addition, MVF was not implemented on cloud-native edge Boxes having IoT devices. These mentioned challenges limit the capabilities of SmartX MVF for maintaining multi-view visibility both at distributed resources (Micro-Box) and centralized collection center (Visibility-Center). To overcome these limitations, our solution added time-series summarization with an improved visualization scheme in an updated environment of cloud-native edge Boxes.

2.2. OF@TEIN+ Playground with multi-site cloud-native Edge Boxes

OF@TEIN+ Playground is a miniaturized overlay-interconnected, multi-site Playground over heterogeneous underlay networks with hyper-converged Box-style resources. To facilitate developers in learning operational and development issues as well as perform various experiments, OF@TEIN+ Playground supports multiple resource types: physical, virtual, and container types [21]. Generally, physical resources consist of physical servers and switches, and physical interconnects. Virtual resources, mainly constitute virtual machine instances (via the support of hypervisors), virtual switches, and virtual interconnects. In the given environment, we distributed cloud-native edge enabled Boxes i.e. Micro-Box, at multi-site edge locations of OF@TEIN+ Playground. Container instances recently entered the scene to provide a new lightweight category of resources for flexible workload deployment. Unlike cloud-based infrastructure that hosts virtual machines and has fixed allocated resources for users, in cloud-native, we adapt for containers to deploy applications over these Boxes. As mentioned in the Introduction section, Micro-Boxes are commodity server-based hyper-converged resources (compute /storage/networking) to allow experiments (Cloud/Network Function Virtualization) over OF@TEIN+ Playground.

Micro-Boxes have combined interfaces for management/control traffic as one and data, IoT devices, and remote power control.

2.3. Requirements of Spatio-temporal Visibility for monitoring

This research is focused on identifying requirements of summarized spatio-temporal multi-view visibility to understand the long-term operations of the Playground. Maintaining summarized spatio-temporal visibility in cloud-native infrastructure poses several challenges. First Boxes may join and leave the network (e.g. shutdown, connection failure). The operators for troubleshooting purposes require temporal Information on the on-going status of these resources. Secondly, job scheduling, resource provisioning, and allocation mechanism require monitoring metrics with a real-time collection and low latency. However, edge Boxes, in a geographically distributed infrastructure could induce delay. This creates the need for keeping the monitoring data at the edge and keeping the resource collection at a reasonable volume. Thirdly visualization of collected visibility data with analysis is a requirement to verify the experiment and provide an environment for decision-making. To address these challenges, requirements of maintaining persistence multi-view visibility are as follow,

R1: To design, develop, and implement infrastructure that is capable of collecting visibility metrics as spatio-temporal multi-view visibility on cloud-native edge Boxes.

R2: Reducing the data size of flow-layer with spatio-temporal compression. Specifically, a compression algorithm is developed based on spatial similarities between multiple streams of data over a specific time.

R3: Aggregate the time-series visibility metrics to identify spatio-temporal summarized visibility. Leveraging summarized visibility facilitates in identifying meaningful information such as missing collection, mean value for a day and percentage uptime, etc.

R4: To verify spatio-temporal visibility using multiple visualization schemes. Provide visualization support to enable support for time-series based visibility data analysis.

3. Spatio-Temporal SmartX Multi-View Visibility: Design and Implementation

In this section, we discuss the proposed design of spatio-temporal summarized multi-view visibility in cloud-native edge Boxes and identify key terminologies. Afterward, we discuss detailed components design and implementation as functionalities responsible for summarizing spatio-temporal multi-view visibility from the measurement phase at the Micro-Box up-to-the visualization phase at the Visibility-Center.

3.1. Design for Spatio-Temporal SmartX Multi-View Visibility

Spatio-temporal summarized monitoring and measurement leveraging SmartX MVF is proposed to deal with the multiple layers such as resource layer (underlay, physical), flow-layer, and workload-layer [22]. In resource-layer visibility, monitoring and visualization of Playground physical resources and inter-connects (e.g. paths and links) are considered. Whereas flow-layer visibility monitors overlay network traffic in near real-time through packet tracing. Flow-layer visibility deals with different levels of flow information (i.e.,

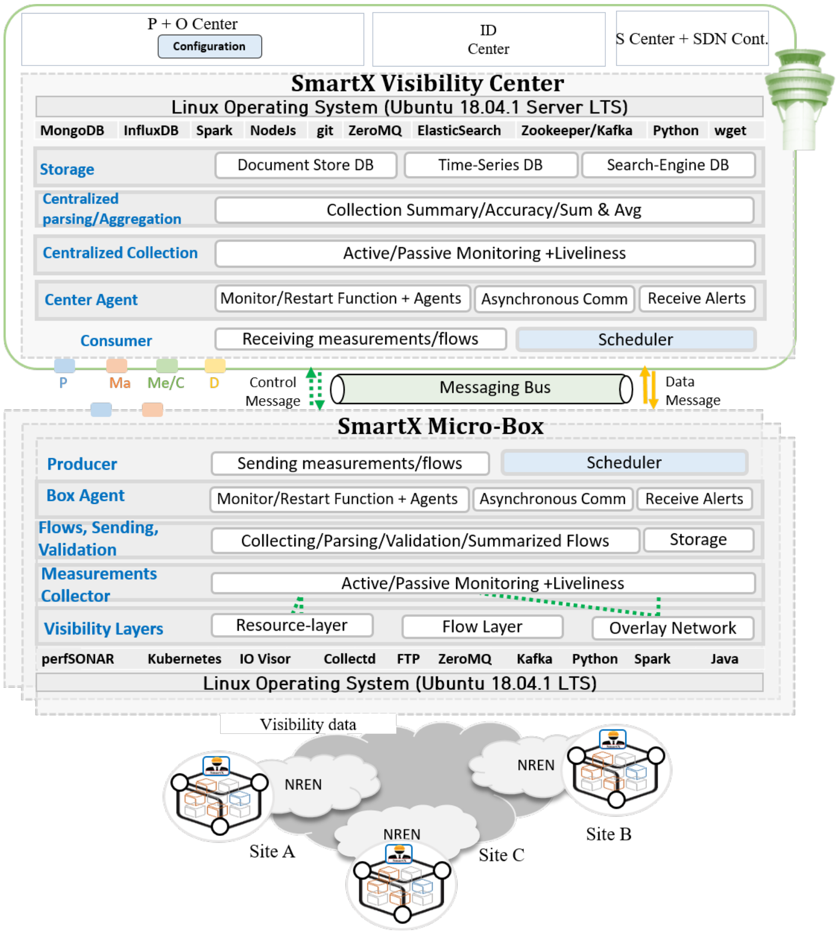

P + O Center
Configuration
ID Center
S Center + SDN Cont.
SmartX Visibility Center
Linux Operating System (Ubuntu 18.04.1 Server LTS)
MongoDB InfluxDB Spark NodeJs git ZeroMQ ElasticSearch Zookeeper/Kafka Python wget
Storage
Document Store DB
Time-Series DB
Search-Engine DB
Centralized parsing/Aggregation
Collection Summary/Accuracy/Sum & Avg
Centralized Collection
Active/Passive Monitoring +Liveliness
Center Agent
Monitor/Restart Function + Agents
Asynchronous Comm
Receive Alerts
Consumer
Receiving measurements/flows
Scheduler
P
Ma
Me/C
D
Control Message
Messaging Bus
Data Message
SmartX Micro-Box
Producer
Sending measurements/flows
Scheduler
Box Agent
Monitor/Restart Function + Agents
Asynchronous Comm
Receive Alerts
Flows, Sending, Validation
Collecting/Parsing/Validation/Summarized Flows
Storage
Measurements Collector
Active/Passive Monitoring +Liveliness
Visibility Layers
Resource-layer
Flow Layer
Overlay Network
perfSONAR Kubernetes IO Visor Collectd FTP ZeroMQ Kafka Python Spark Java
Linux Operating System (Ubuntu 18.04.1 LTS)
Visibility data
NREN
NREN
NREN
Site A
Site B
Site C

Fig. 2. Design for spatio-temporal summarized functionalities at Micro-Box and Visibility-center.

collected, clustered, identified, and un-clustered flow) by utilizing a balanced flow collection, clustering, and tagging. A network flow is typically a sequence of network packets that belongs to a certain network session between two endpoints. Next, workload-layer visibility is responsible for monitoring and visualization of inter-connected containerized functions (e.g. Web, App, and DB) and their deployments for tenant-based applications.

To handle the challenges of spatio-temporal operations, we leverage SmartX Micro-Box distributed at multi-site edge locations. These Boxes support cloud-native (containerized) IoT-Gateway with DataPond functionality (with IoT-Cloud Hub and other application functions) and prepared as Kubernetes-orchestrated workers with SDN-coordinated special connectivity to other SmartX Boxes. As shown in Fig 2, SmartX-Micro-Box is deployed at multiple sites in OF@TEIN+ Playground with capabilities of sending control/data messages at the Visibility-Center employing multiple tools. At each stage, multiple functionalities are deployed to ensure smooth operations. In the Micro-Box *"Vis-*

ibility Collection and Validation“ stage collects and validates monitoring and measurement data based on selected visibility metrics. Formatted visibility data is sent to *“Visibility Integration and Storage“* stage where data is stored and integrated to accommodate historical/latest records. *“Aggregation and Summarization stage“* deals with generating consolidated reports and to prepare for data analysis as well as identify any anomalies. Finally, *“Spatio-temporal Visualization“* stage accesses processed data and auto-generates graphical views. Following in this section, we discuss the aforementioned four main stages of spatio-temporal SmartX Multi-View Visibility.

3.1.1 Multi-layer Visibility Measurement At the Micro-Box the visibility collection and validation stage collects and validates monitoring and measurement data based on selected visibility metrics. For reliable and resilient operation network measurement for all traffic passing through, Micro-Boxes are enabled with end-to-end link quality monitoring.

Table 1. A selected list of multilayer visibility metrics for OF@TEIN SDN-enabled multi-site clouds

Visibility-Layer	Metric type	Measured Metrics	Measurement Interval
Underlay-Resource	Active monitoring (connectivity status)	Ping (Between Site-to-site and site-visibility centre)	10 minutes
	Active Monitoring (Network performance)	Latency (Between Site-to-site)	10 minutes
	Active Monitoring (Network capacity)	Bandwidth (tcp, udp)	2 time/day
Flow	Overlay network traffic as passive monitoring	Packet tracing	30 seconds
	Summarized Flows	Flows from Packet tracing	5 minutes
Physical-Resource	Traffic on Interface	Bytes received/sec, Bytes sent/sec	10 seconds
	System performance stats	CPU/Load/memory/disk	10 seconds
Workload	Agent-based monitoring and alert	Applications running status /alerts	5 minutes

As shown in table 1, to capture visibility data for link quality, we collected specific measurement metrics with regular intervals termed as “active monitoring” [23]. For packet precise collection, we capture the network packets from each network interface of SmartX Micro-Box termed as “passive monitoring”. To persistently monitor OF@TEIN+ infrastructure, reliable visibility data transfer is applied to sustain the data locally during network failure or application failure. Box-Agents are deployed to convey the status of running functionalities with the support of asynchronous messaging during Agents-based communication.

3.1.2 Integration and Storage Next Visibility Integration stage integrates collected data for generating consolidated reports over a period and identifies any anomalies. Formatted visibility data is sent to Visibility Storage and Staging stage where data is stored.

Table 2. Raw format Multi-View Visibility data for parsing metrics Multiple DB

Visibility Data for (CPU) in JSON Format at InfluxDB	Visibility Data (Load) in JSON Format at Elastic Search	Visibility Data (latency) in JSON Format at MongoDB

[”2020-02-01T02:55:01.54Z”,
”smartx-microbox-gist-1”,
”idle”,
99.5003109715353
], [”2020-02-01T02:55:34.28Z”,
”smartx-microbox-gist-1”,
”system”,
5.60003485634267
], [
”2020-02-01T02:57:21.43Z”,
”smartx-microbox-gist-1”,
”softirq”,
0.0333330630191482]
	{
	”_index”:
	”pers_collectd_cpu”,
	”_type”: ”mirror”,
	”_id”: ”120”,
	”_score”: 1,
	”_source”: {
	”@version”: ”1”,”@timestamp”:
	”2019-06-27T10:46:33.738Z”,
	”boxid”: ”smartx_microbox_um_2”,
	”plugin”: ”cpu”,”type_instance”:
	”softriq”,”values(%)”: 0.50001657}
	}}
		{
		”_id” : ObjectId
		(”5ce19081497327a”),
		”timestamp” :
		”2019/05/20 02:21:05 KST
		”microbox-SOURCE” :
		”microbox-vnu-1
		”microbox-gist-1” : ”224
		”microbox-gist-2” : ”222
		”microbox-um-1” : ”76.5
		”microbox-um-2” : ”75.0
		”microbox-chula-1” : ”87.5
		”microbox-itc-1” : ”43.5
		”microbox-ucsm-1” : ”101
		”microbox-drukren-1” : ”128
		”microbox-itb-1” : ”78.7
		”microbox-ptit” : ”0.295”}

Table 2 shows the collected metrics for physical/underlay resource layer metrics. These metrics are parsed and integrated into multiple databases according to measurement type. As shown in Table 3 metrics for multiple layers of visibility collection after getting parsed and validated are stored to one of NoSQL Data Stores, which are deployed in our DataLake.

Depending on various monitoring requirements, we choose InfluxDB, MongoDB, and Elasticsearch. We consider three options i.e. MongoDB is suitable for configuration data and Elasticsearch is good for logs, and InfluxDB is suitable for time-series data. Currently, we extensively utilize MongoDB to store Playground configuration and various Playground entities status data such as real-time resource-layer data and aggregated data used for generating a daily visibility report based on aggregation of visibility. While InfluxDB and Elasticsearch are used to store near-real-time metrics data and flows data respectively. Since MongoDB (document-oriented), Elasticsearch (index-oriented), and InfluxDB (time series) are all special-purpose NoSQL Data Stores, they depend on separate store configuration and status data for different resources of OF@TEIN+ Playground. The configuration is handled via MongoDB collection. At the backend, java-based plugins are utilized to store and update Playground metrics in respective databased at regular intervals.

3.1.3 Aggregation and Summarization In this stage, we are concerned with multiview visibility through time-series summarization and aggregation support. Summarization is performed first at the Micro-Box where packet tracing is summarized on a specific time window (5 minutes) to generate flows. These summarized flows are used to extract useful information and to reduce the size of visibility data.The implementation for the summarization is depicted in Algorithm 1.

Algorithm 1 Packet Tracing for Base Collection

```
Input: Basic IP Packet                                              ▷ p = packet buffer
Output: 6-tuples of header form IP-only packet as tracing criteria       ▷ 6-tuple packet
procedure BASE TRACING($p$)
    if Packet in the Interface does not match the given IP ($p.ethtype \neq IP$) then
        Drop the packet p
    end if
    Extract source address ($srcaddr \leftarrow p.ip.srcaddr$)
    Extract destination address $destaddr \leftarrow p.ip.destaddr$
    Extract packet length$length \leftarrow p.ip.totallength$
    if $p.ip.nextproto \neq TCP$ then
        if $p.ip.nextproto \neq UDP$ then
            Drop the packet p
        end if
    end if
    Extract protocol ($protocol \leftarrow p.ip.nextproto$)
    Extract destination port ($dstport \leftarrow p.ip.tcp.dstport$)
    Extract source port ($srcport \leftarrow p.ip.tcp.srcport$)
    return Headers
end procedure
```

An appropriate spatio-temporal data-mining algorithm is selected to run on the preprocessed data, and produce output patterns. Common output pattern families include spatio-temporal summarization and change patterns. Spatio-temporal data mining algorithms often have statistical foundations and integrate scalable computational techniques. Output patterns are post-processed and then assist Playground operators to find novel insights and refine data mining algorithms when needed.

Next, a summarization tool is utilized to generate a time-specific summarized multiview visibility data formatted as HTML format report. Figure 3 shows the equations used for aggregating visibility as Daily collection count, Daily collection percentage and Daily collections missing over a defined time. These compiled summarizations are dissipated animatedly at regular intervals to operators/administrators for troubleshooting and analysis purposes. These reports provide status of distributed Playground resources in one window by aggregating the data with average/percentage/collection for one-day duration. Summarization for multi-view visibility is performed as daily collection count, percentage, missing collection, etc.

3.1.4 Spatio-Temporal Visualization Visibility data in the databases is stored as multiview data in a time-series format. This data is processed and stored either as real-time

Daily collection count DCC for space(S) and time(T) can be calculated by a simple method,

$$DCC(S,T) = \sum_{k=1}^{t} d(c_t) \tag{1}$$

$$\text{where } d(c_t) \text{ can be defined as } d(c_t) = d(s_{i_k}, t_{i_k}),$$
$$t \text{ as measurement interval, } c \text{ as count,}$$
$$s_{i_k} \text{ as value of variable, } t_{i_k} \text{ as timestamp}$$

Similarly, Daily collection percentage can be calculated as,

$$DCP(S,T) = \sum_{k=1}^{t} d(c_t)/t * 100 \tag{2}$$

Finally Daily collections missing can be computed as,

$$DCM = \sum_{k=1}^{t} d(exp_{c_t}) - d(c_t) \tag{3}$$

$$\text{where } d(exp_{c_t}) \text{ is daily expected collection}$$

Fig. 3. Equations for Aggregation

measurements or as summarised data. In Visualization stage, processed data is accessed through multiple visualization tools to auto-generates graphical views for further exploration of graphical visibility. Visualization support with the previous approach was limited and desktop-oriented while work presented in this paper provides a long-term visualization in varying perspectives.

1. First measurement metrics that belongs to same category are summarised over a particular time-period to show resource distribution among them.
2. Second we provide visualization of multiple metrics for single-site to identify mutuality between them. For example CPU utilization metrics are visualized on one page to identify similar patterns.
3. Third we multi-site visualization for distributed resources on a single view with the same time-line. This visualization assists in identifying the differences between resources/sites behavior. To further explore the multi-site visualization we employ analysis schemes such as trends patterns and seasonal patterns.

To visualize time-series based metrics we utilize open-source software named Grafana and Kibana, Matplot, abnd Seaborn library. In order to realize the multi-belt onion-ring visualization, we leverage open-source visualization library called psd3 [9]. That is, psd3 is primarily based on D3.js and supports multi-level pie charts, whereas, for deployment Node.js, JavaScript runtime is selected.

As shown in Figure 4, for utilization of Playground operations by end-user, we developed a cloud-native service through Kubernetes and distributed it through OF@TEIN+ Playground. One Kubernetes cluster was formed with Edge IoT-gateways distributed at Multi-site of OF@TEIN+ Playground. First, the Gwangju Institute of Science and Technology (GIST) in Korea uses two Micro-Boxes and one Raspberry Pi as Kubernetes Worker nodes and the P+O Center of SmartX Playground Tower as the master node.

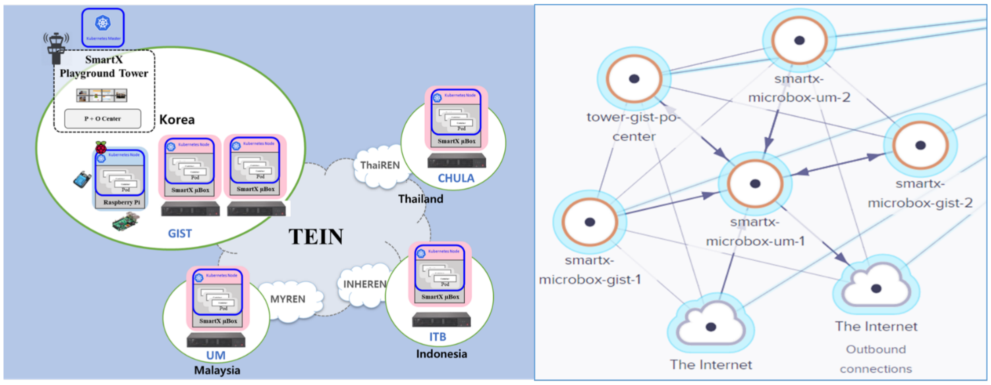


Fig. 4. Kubernetes clusters in OF-TEIN+ Playground.

We also set up a multi-site cloud-native environment with Micro-Box located at Chula-Thailand, ITB-Indonesia and UM-Malaysia. In a multi-site verification environment, Calico network add-on is used for functions connectivity.

3.2. Implementation for Spatio-Temporal MultiView Visibility

The smooth operations of visibility collection require first the identification of key monitoring metrics and corresponding collection tools. Secondly, tools are developed for later stages of visibility data Integration, storage, and summarization. Third, to manage visualization selective tools are employed.

Table 3. SmartX Micro-Box Software: Multiple Levels of Functionalities

Functionality	**Purpose**	**Responsible**
Core Functionalities	Base software functionality,to manage Micro-Box operation. OS, Kernel, Inter-connects, Kubernetes, Docker, etc.	Playground Operators
Basic Functionalities	Development at the Bare metal Monitoring of connection status/health with connected Boxes in the Playground,	Playground Operators
Application Functionalities	Applications in the form factor of containers, orchestrated through Kubernetes. Example:Smart Energy Service	Service Developers

The functionalities in SmartX Micro-Box and Visibility center are categorized as Core, Basic, and Application functionalities based on their purpose and user's role. As mentioned in table 3, *Core Functionalities* deals with base software such as OS, Kernel, etc. Whereas *Basic Functionalities* corresponds development at the bare-metal. The *Application Functionalities* are the services running in the containers.

Table 4. Raw format Multi-View Visibility data for parsing metrics Multiple DB

App Name	App Role	Layer	Form factor	Source
IOVisor	Passive Monitoring	Flow-layer	BareMetal	*Micro-Box
Collectd	Passive Monitoring	Flow-layer		
perfSONAR	Active Monitoring	Resource-layer		
Box Liveliness with Tower/App/Box	Active Monitoring		BareMetal	
Apache Kafka	Reporting	Resource-layer		
Box Agent	Reporting	Resource-layer		
ZeroMQ	Reporting	Resource-layer		
Docker Container	Accessibility	Resource-layer	VM/BareMetal	
Collector/Aggregator /Storage/Measurement Report	Preparation	Resource-layer		Visibility-Center
Automated Mail	Reporting	Resource-layer		
Integration	Summarization	Resource-layer		
Grafana/Kibana/Onion-Ring	Vizualization	Resource-layer		
Kubernetes	Orchestration	Resource-layer		P+O Center

Table 4 lists the selected functionalities as tools developed and deployed in Micro-Box and at the Visibility-Center along with their role, layer, and form factor. Currently, for resource layer visibility collection, we utilize Collectd to collect transfers and stores performance statistics of Micro-Box [24]. Besides, for flow-layer visibility collection, we use eBPF-based packet tracing tools [7]. While Apache-Spark with Scala is utilized to generate flows from packet tracing. Besides, for resource-layer visibility we employed PerfSONAR command-line tools for monitoring Playground resources, interconnects, and end-to-end network performance metrics. To ensure consistent running of monitoring applications we placed Box-agents written in python at each resource that periodically check the running status and actively start the application within a short interval. Besides sending status, Box-Agents communicate with Centre-Agent through zeroMQ-based communication to handle asynchronous communication. To reliably transfer the visibility data enduring temporary network loss and delays, we apply Apache Kafka for messaging. A customized python-based tool format the data in the required format with tags and field values. At the visibility Center together with Apache Kafka, to manage reliable and persistent transport of visibility data, we are using Apache Zookeeper, which provides automatic management of metadata and synchronization issues. A customized java-based tool parse and validate the visibility data before storing it in the appropriate databases (i.e., MongoDB, ElasticsSearch, and InfluxDB). At the end of the day, a java based tool aggregates values of multi-view visibility measurements which can be useful for operators and administrators for analysis purposes. For a single view unified visualization, we apply multi-belt onion-ring visualization. Besides, for visualization of measurements from flow-layer and resource-layer visibility, we utilize Kibana and Grafana tools. For orchestration of containerized application, we have employed Kubernetes.

4. Spatio-Temporal Summarized Visualization of SmartX Multi-View Visibility: Verification

For verifying the prototype implementation, we utilize the OF@TEIN+ Playground. We consider metrics from visibility tools and visualized visibility data at each step of visibility workflow.

4.1. Verification Setup

In the initial-stage, deployment of Visibility-center utilizes server-based hardware with the specs: Intel® Xeon CPU E5-2690 V2@3.00GHz, HDD 5.5TB, memory DDR3 12x8GB, and 4 network interfaces of 1 Gbits/s. SmartX Micro-Box consists of Supermicro SuperServer E300-8D server. This Mini-1U server consists of 4 CPU cores with 2.2GHz Intel processor, 240 GB of hard disk, and 32 GB memory. There is one dedicated physical interface for IPMI-based remote access management through CLI (command-line interface) and the web UI (user-Interface). In addition, there are 2 10G + 6 1 GB network interfaces. Visibility Center is configured with Ubuntu 16.04.4 LTS OS, while SmartX Micro-Box loads Ubuntu 18.04.2 LTS OS, with a minimum kernel version of 4.4.0. A dedicated tenant is provided to developer for executing different experiments. In OF@TEIN+ Playground eight SmartX Micro-Boxes are included under current with three types of functionalities i.e. Core, Basic, and Application. Each Box is configured with a scheduled application at synchronized time and intervals for reliable and consistent operations of the Playground.

4.2. SmartX Multi-View Visibility collection and storage

In the proposed solution, measurements are collected at defined regular intervals and sent through Kafka producer at the Visibility Center, where Kafka Consumer fetch the collected measurements. A customized Java-based application validates and parses and format these measurements to write the visibility data in the Database i.e. MongoDB, ElasticSearch, and InfluxDB. Each measurement is tagged with a timestamp and source identifier.

To emphasize the time-series collection several months of visibility data is collected. Figure 5 shows a daily collection of flows at the Visibility-Center from Micro-Box at GIST-1 site. Each collection is tagged with a timestamp and name of source site before storing in the database. In the distributed environment, visibility collections from multiple resources are collected and stored regularly at the center. Administrators are required to maintain an overview of incoming data in terms of volume generated and any missing collection for the observed time.

To facilitate the administrators for smooth operations, we build a customized tool that provides verification of collection through summarized visibility data. Multi-view visibility data is aggregated in-terms of collection values, count, and percentage. Figure 6 shows resource layers metric collection aggregated for a 24-hour duration.

In SmartX Multi-View visibility, the flow layer is key to integrate the multiple visibility layers. To integrate flow-layer with the resource layer, we inspect each network packet and correlate them with predefined virtual and physical resources from the Playground

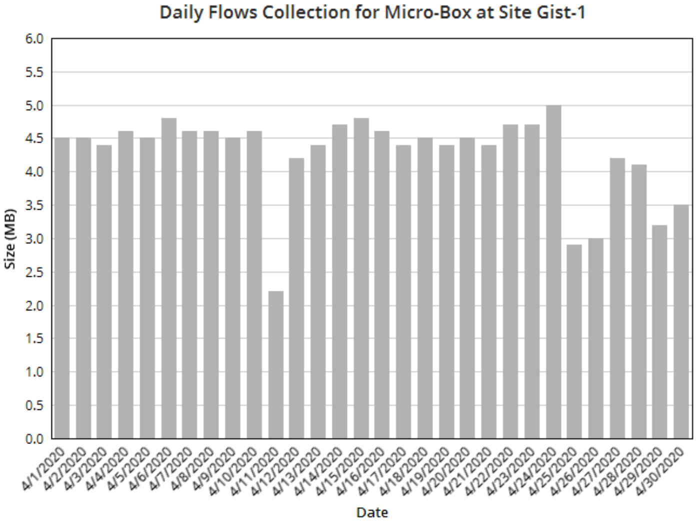
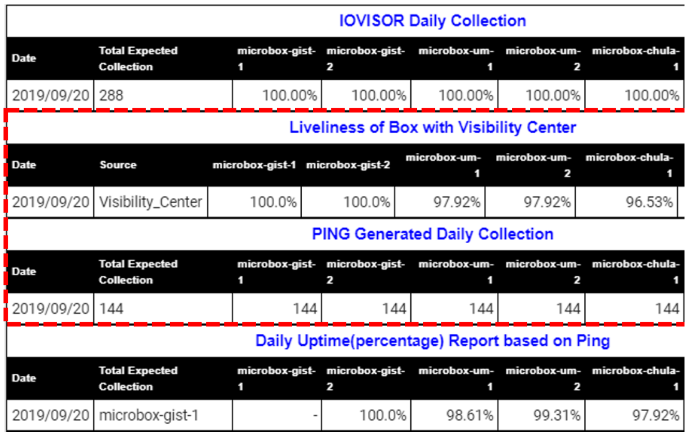


Fig. 5. Result of one-month flows collection for Micro-Box at Site Gist-1



IOVISOR Daily Collection

Date	Total Expected Collection	microbox-gist-1	microbox-gist-2	microbox-um-1	microbox-um-2	microbox-chula-1
2019/09/20	288	100.00%	100.00%	100.00%	100.00%	100.00%

Liveliness of Box with Visibility Center

Date	Source	microbox-gist-1	microbox-gist-2	microbox-um-1	microbox-um-2	microbox-chula-1
2019/09/20	Visibility_Center	100.0%	100.0%	97.92%	97.92%	96.53%

PING Generated Daily Collection

Date	Total Expected Collection	microbox-gist-1	microbox-gist-2	microbox-um-1	microbox-um-2	microbox-chula-1
2019/09/20	144	144	144	144	144	144

Daily Uptime(percentage) Report based on Ping

Date	Total Expected Collection	microbox-gist-1	microbox-gist-2	microbox-um-1	microbox-um-2	microbox-chula-1
2019/09/20	microbox-gist-1	-	100.0%	98.61%	99.31%	97.92%

Fig. 6. Daily Visibility Summarized Collection for analysis or multi-view visibility

database. The aggregated network traffic can be used to instantly highlight congested links and identify the source of the flow and associated applications. To minimize the induced size overhead, we have modified SmartX MVF to implement flow aggregation at the Micro-Box with the integration capabilities at the Visibility Center as shown in Figure 7 .

4.3. Spatio-Temporal Summarized Visualization

For verification, in this section we provided three types of summarized visualizations, i.e. (i) Single-site Visualization, (ii) Multi-Metrics on Single site Visualization, and (iii)

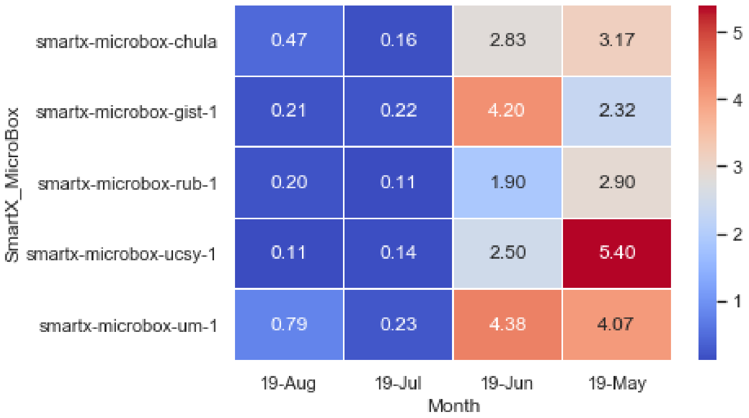


Fig. 7. Results for storage volume of summarized flows and packet tracing without summarization over several months of collection

Multi-site Visualization. Each of these visualizations highlights an important visualization aspect for operators/administrators. We start with summarized time-based multi-view visibility data for analysis purposes. Next, we provide visualization for a single system metric for a single site with multiple parameters on the time-line. Followed by a comparison of multiple system metrics on a single site to understand their correlation. Next, we compared measurement from multiple sites to understand the metrics change together over time. Finally, we showed visualizations analysis, leveraging trends, and seasonal pattern schemes.

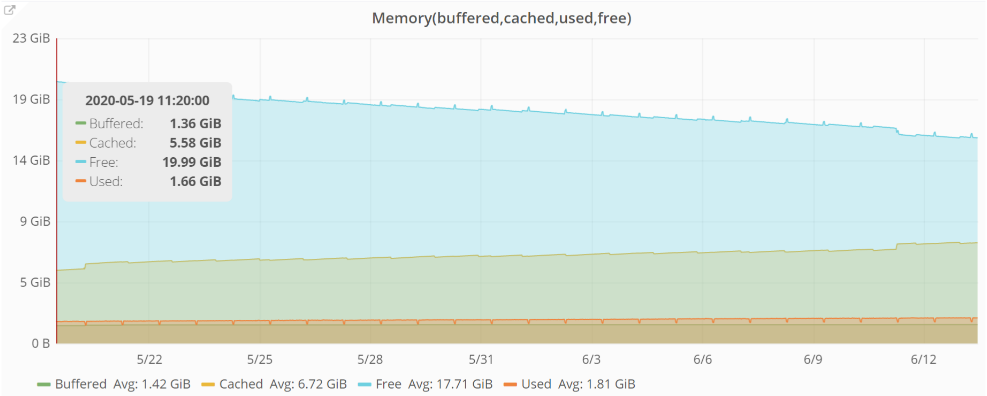


Fig. 8. Visualization results for system Utilization metrics (Memory)

Next we demonstrate time-series visualization for single metrics on a single site with multiple parameters. As shown in 8, Metrics for Memory plugin collects physical memory utilization with values for multiple parameters such as System, idle, nice.

After this we demonstrate summarized visualization for multiple system performance metrics (i.e. load, interface, disk, memory) on a single site in one layer. Unlike previous research, where these metrics were visualized separately. Here we have visualized on a single layer for comparison and analyze the variation over time-line as shown in figure 9 .

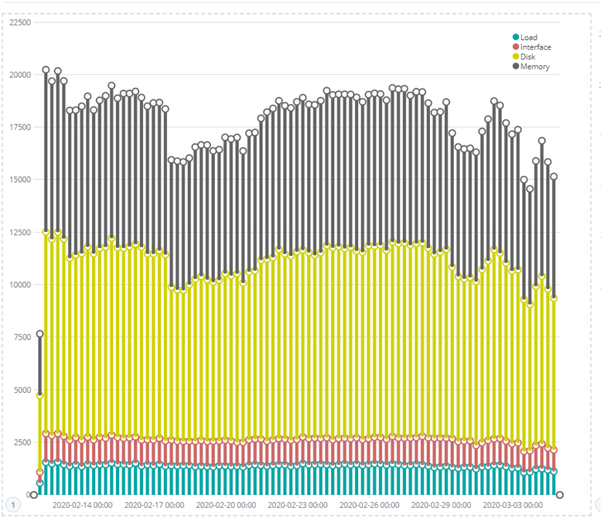
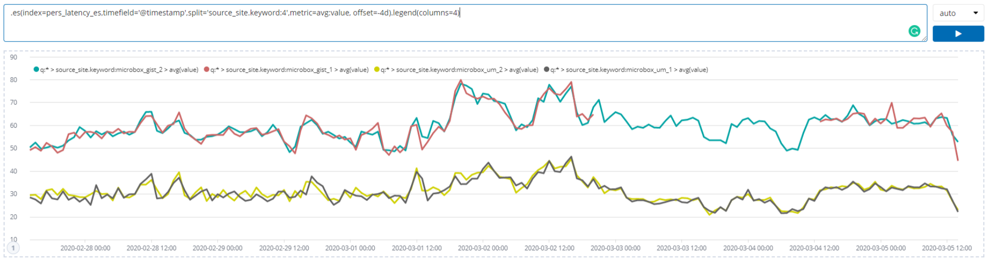


Fig. 9. Results of visualization for multiple metrics of physical resource layers from a single site over time-line



```
.es(index=pers_latency_es,timefield='@timestamp',split='source_site.keyword:4',metric=avg:value, offset=-4d).legend(columns=4)
```

Fig. 10. Result for comparison of visibility measurements from multiple sites over time-line

Figure 10 demonstrate summarized visualization from multiple sites (Micro-Boxes) for underlay resource-layer metrics i.e. latency . Such visualization is useful to identify the comparison of heterogeneous networks for identifying how these metrics change and differ from other sites. As shown in Figure, measurements from four sites are visualized over for a month. There is a clear similarity between measurements of sites (Gist-1 and Gist-2 at South-Korea), while the other two sites (UM-1 and UM-2 at South-Korea) have similar values. However, these two pairs of sites have different latency measurement between them.

In Spatio-temporal analysis, another useful function to use is trend. This function can be used to display a trend line using either a log or linear regression algorithm. There are several ways to think about identifying trends in time series. One popular way is by taking a rolling average, which means that, for each time point, average of the points is taken on either side of it. The number of points is specified by a window size, which needs to be chosen. Figure 11 shows trend analysis for resource layer metrics, i.e. Memory utilization for multiple resources on a single time-line.

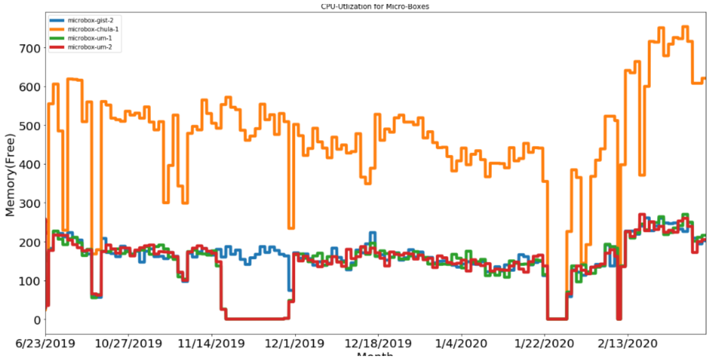


Fig. 11. Trends pattern based visualization of memory utilization for multi-sites

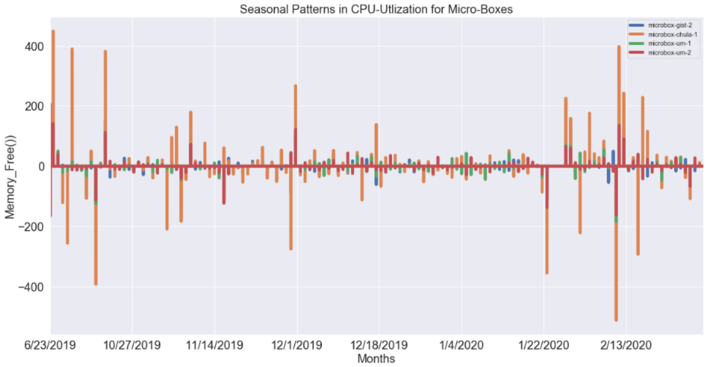


Fig. 12. Seasonal pattern based visualization of memory utilization for multi-sites

The seasonal component in spatio-temporal analysis describes the recurring variation of the time series. Seasonal patterns are utilized to detect seasonality in a time series i.e., Seasons are usually described in the context of mean values averaged per month or several months, and detailing relates mainly to spatial information[25]. For identifying seasonal patterns we subtract the trend computed above (rolling mean) from the original values. This, however, will be dependent on how many data points you averaged. Figure 12 shows seasonal Patterns as time Series Data for memory utilization in the physical resource-layer.

5. Conclusion

In this paper, we presented our initial effort to provide summarized spatio-temporal operations for OF@TEIN+ playground developers and operators to effectively operate and maintain the playground. We verified the work by visualizing multiple layers of visibility leveraging SmartX Multi-view visibility. As future step, we are working on incorporating visualization for containerized applications as temporal visibility collection.

Acknowledgments. (Security) Project: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00421, Cyber Security Defense Cycle Mechanism for New Security Threats). AI Graduate School Project: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST))

References

1. Shifen Cheng, Feng Lu, Peng Peng, and Sheng Wu. Multi-task and multi-view learning based on particle swarm optimization for short-term traffic forecasting. *Knowledge-Based Systems*, 180:116–132, 2019.
2. Polona Štefanič, Matej Cigale, Andrew C Jones, Louise Knight, Ian Taylor, Cristiana Istrate, George Suciu, Alexandre Ulisses, Vlado Stankovski, Salman Taherizadeh, et al. Switch workbench: A novel approach for the development and deployment of time-critical microservice-based cloud-native applications. *Future Generation Computer Systems*, 99:197–212, 2019.
3. J Kim, B Cha, Jongryool Kim, Namgon Lucas Kim, Gyeongsoo Noh, Youngwan Jang, Hyeong Geun An, Hongsik Park, J Hong, D Jang, et al. Of@ tein: An openflow-enabled sdn testbed over international smartx rack sites. *Proceedings of the Asia-Pacific Advanced Network*, 36:17–22, 2013.
4. Aris Cahyadi Risdianto, Junsik Shin, and JongWon Kim. Building and operating distributed sdn-cloud testbed with hyper-convergent smartx boxes. In *International Conference on Cloud Computing*, pages 224–233. Springer, 2015.
5. Aris Cahyadi Risdianto, Muhammad Usman, and JongWon Kim. Smartx box: Virtualized hyper-converged resources for building an affordable playground. *Electronics*, 8(11):1242, 2019.
6. Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016.
7. Muhammad Usman, Aris Cahyadi Risdianto, Jungsu Han, Moonjoong Kang, and JongWon Kim. Smartx multiview visibility framework leveraging open-source software for sdn-cloud playground. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–4. IEEE, 2017.

8. Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuno. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, 2008.
9. Bartolo Luque, Lucas Lacasa, Fernando Ballesteros, and Jordi Luque. Horizontal visibility graphs: Exact results for random time series. *Physical Review E*, 80(4):046103, 2009.
10. Jie Zhang and Michael Small. Complex network from pseudoperiodic time series: Topology versus dynamics. *Physical review letters*, 96(23):238701, 2006.
11. Yue Yang and Huijie Yang. Complex network-based time series analysis. *Physica A: Statistical Mechanics and its Applications*, 387(5-6):1381–1386, 2008.
12. Norbert Marwan, Jonathan F Donges, Yong Zou, Reik V Donner, and Jürgen Kurths. Complex network approach for recurrence analysis of time series. *Physics Letters A*, 373(46):4246–4254, 2009.
13. Xiaoke Xu, Jie Zhang, and Michael Small. Superfamily phenomena and motifs of networks induced from time series. *Proceedings of the National Academy of Sciences*, 105(50):19601–19605, 2008.
14. Grégoire Nicolis, A Garcia Cantu, and Catherine Nicolis. Dynamical aspects of interaction networks. *International Journal of Bifurcation and Chaos*, 15(11):3467–3480, 2005.
15. Andriana SLO Campanharo, M Irmak Sirer, R Dean Malmgren, Fernando M Ramos, and Luís A Nunes Amaral. Duality between time series and networks. *PloS one*, 6(8), 2011.
16. Zhe Jiang and Shashi Shekhar. Spatial and spatiotemporal big data science. In *Spatial Big Data Science*, pages 15–44. Springer, 2017.
17. Shashi Shekhar, Zhe Jiang, Reem Y Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. Spatiotemporal data mining: a computational perspective. *ISPRS International Journal of Geo-Information*, 4(4):2306–2338, 2015.
18. Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. Ftw: fast similarity search under the time warping distance. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 326–337, 2005.
19. Saif Ahmad, Tugba Taskaya-Temizel, and Khurshid Ahmad. Summarizing time series: Learning patterns in 'volatile'series. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 523–532. Springer, 2004.
20. Shreya Ghosh, Anwesha Mukherjee, Soumya K Ghosh, and Rajkumar Buyya. Mobi-iost: mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications. *IEEE Transactions on Network Science and Engineering*, 2019.
21. Muhammad Usman, Nguyen Tien Manh, and JongWon Kim. Multi-belt onion-ring visualization of of@ tein testbed for smartx multi-view visibility.
22. Salman Taherizadeh, Andrew C Jones, Ian Taylor, Zhiming Zhao, and Vlado Stankovski. Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review. *Journal of Systems and Software*, 136:19–38, 2018.
23. Sihyung Lee, Kyriaki Levanti, and Hyong S Kim. Network monitoring: Present and future. *Computer Networks*, 65:84–98, 2014.
24. Florian Forster and S Harl. collectd–the system statistics collection daemon, 2012.
25. Yury Kolokolov and Anna Monovskaya. Multidimentional analysis and visualization of changes in characteristic seasonal patterns of local temperature dynamics. In *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, volume 1, pages 321–327. IEEE, 2017.

Muhammad Ahmad Rathore received his BS in Computer Science from Comsats Institute of Information Technology, Islamabad, Pakistan, and his MS in Information and Communication Systems Security from Royal Institute of Technology, Stockholm, Sweden. Currently, he is pursuing Ph.D. Degree from Networked Intelligence Lab, School of

Electrical Engineering and Computer Science, Gwangju Institute of Technology, GIST, South Korea. His main research interests are in cloud-based technologies, visualization, and data analysis.

JongWon Kim received Ph.D. degree in Control and Instrumentation Engineering from Seoul National University, Seoul, Korea, in 1994. In 1994-1999, he was with the Department of Electronics Engineering at the KongJu National University, KongJu, Korea, as an Assistant Professor. From 1997 to 2001, he was visiting the Signal & Image Processing Institute (SIPI) of Electrical EngineeringSystems Department at the University of Southern California, Los Angeles, USA, where he has served as a Research Assistant Professor since Dec. 1998. From Sept. 2001, he has joined Gwangju Institute of Science & Technology (GIST), Gwangju, Korea, where he is now working as the chair of GIST AI Graduate School, which was established late 2019 as one of 5 government-sponsored AI graduate schools in Korea.

Received: March 17, 2020; Accepted: September 15, 2020.

Computer Science and Information Systems 18(1):189–212 https://doi.org/10.2298/CSIS200223033Y

A QPSO Algorithm Based on Hierarchical Weight and Its Application in Cloud Computing Task Scheduling

Guolong Yu[1], Yong Zhao[2], Zhongwei Cui[1], and Yu Zuo[1]

[1] School of Mathematics and Big Data, Guizhou Education University,
Guiyang, 550000, China
heihuzhiguang@163.com

[2] School of Information Engineering, Shenzhen Graduate School of Peking
University, Shenzhen, 518000, China
516636425@qq.com

Abstract. The computing method of the average optimal position is one of the most important factors that affect the optimization performance of the QPSO algorithm. Therefore, a particle position weight computing method based on particle fitness value grading is proposed, which is called HWQPSO (hierarchical weight QPSO). In this method, the higher the fitness value of a particle, the higher the level of the particle, and the greater the weight. Particles at different levels have different weights, while particles at the same level have the same weight. Through this method, the excellent particles have higher average optimal position weight, and at the same time, the absolute weight of a few particles is avoided, so that the algorithm can quickly and stably converge to the optimal solution, and improve the optimization ability and efficiency of the algorithm. In order to verify the effectiveness of the method, five standard test functions are selected to test the performance of HWQPSO, QPSO, DWC-QPSO and LTQPSO algorithm, and the algorithms are applied to the task scheduling of the cloud computing platform. Through the test experiment and application comparison, the results show that the HWQPSO algorithm can converge to the optimal solution of the test function faster than the other three algorithms. It can also find the task scheduling scheme with the shortest time consumption and the most balanced computing resource load in the cloud platform. In the experiment, compared with QPSO, DWC-QPSO and LTQPSO algorithm, HWQPSO execution time of the maximum task scheduling was reduced by 35%, 23% and 21% respectively.

Keywords: QPSO algorithm, hierarchical weight, cloud computing, task scheduling, average optimal location.

1. Introduction

Particle swarm optimization (PSO) is one of the most widely used swarm intelligence algorithms. The algorithm is relatively easy to implement, needs to determine fewer parameters and has the advantages of efficient parallel search, which can effectively solve complex optimization problems. Its performance is a hot research issue in the field of Intelligent Computing in recent years, and it has been widely used in resource scheduling, pattern recognition, complex optimization and other issues. However, the PSO algorithm is easy to fall into the local optimal solution when searching, later the particle convergence efficiency is lower, and it can not converge to the optimal solution with probability

1[1-3]. To improve the global search ability of particles, SUN et al. based on the aggregation of particle swarm, established the delta potential well model in quantum state, and then set the control parameters according to the coordination and self-organization of particle swarm, proposed the quantum behaved particle swarm optimization algorithm, i.e. QPSO algorithm[4,5]. In the QPSO algorithm, particles in the quantum space can appear at any point in the search space with a certain probability, and the motion state of the particle is represented by wave function instead of Newton space motion of the particle, and the probability density function of wave function is used to determine the position of the particle in solution space. This position is random, as long as the particle iterations continuously, it will pass through any position in solution space with a certain probability [6-8]. In this way, particles update their positions according to the quantum behavior, and gradually iterate to the global optimal solution. Compared with PSO algorithm, QPSO algorithm increases the randomness of particles, makes particle updating equation simple, has few control parameters and fast convergence speed.

Although the QPSO algorithm has more advantages than the PSO algorithm, there are still many shortcomings in the QPSO algorithm. Many researchers have made a lot of optimization improvements in its contraction and expansion coefficient, population diversity, convergence efficiency, decision-making strategy of average optimal location, etc. For example, Zhen-Lun Y proposed an improved quantum-behaved particle swarm optimization with elitist breeding (EB-QPSO) for unconstrained optimization in reference [9]. During the iterative optimization process of EB-QPSO, when criteria met, the personal best of each particle and the global best of the swarm are used to generate new diverse individuals through the transposon operators. The new generated individuals with better fitness are selected to be the newpersonal best particles and global best particle to guide the swarm for further solution exploration. In addition to the above optimization of individual and global optimal particles, the dual-group search is also an important QPSO optimization method. Such as a dual-group QPSO with different well centers (DWC-QPSO) algorithm, is proposed by constructing the master-slave subswarms in reference [10]. This algorithm avoids the rapid disappearance of swarm diversity and enhances the global searching ability through collaboration between subswarms. Xue T considers the method of mixed optimization of QPSO under complex conditions, so he proposed a hybrid improved quantum behaved particle swarm optimization (LTQPSO) in reference [11]. The algorithm combines the individual particle evolutionary rate and the swarm dispersion with the natural selection method in the particle evolution process. The algorithm has good robustness and convergence. In order to improve the evolution of quantum individuals and the ability to converge to the optimal solution of the QPSO algorithm, Chen W proposed a mixed quantum algorithm based on local optimization strategy and improved optimization rotation angle in reference [12].

Not limited to the above introduction, many researchers have done a lot of work to optimize the convergence efficiency of the QPSO algorithm. In this paper, from the point of view that the weight of each particle's position should be different in the calculation of the average optimal position of different particles, it is considered that the excellent particles should have a larger decision weight, while the inferior particles should have a relatively smaller decision weight. A weight calculation method is proposed, which classifies the weights based on the fitness value of particles, to improve the global search ability and search efficiency of the QPSO algorithm. After the standard test function is tested in the HWQPSO algorithm and the original QPSO algorithm, the DWC-QPSO algorithm in reference [10] and the LTQPSO algorithm in reference [11]. The optimized QPSO algorithm in this paper has more advantages than other algorithms not only in local accuracy and global search ability, but also in convergence speed and stability.

With the wide application of cloud computing technology, due to the large amount of data calculation, the computing efficiency of the platform is paid more and more attention. In addition to improving the hardware performance of the platform, the computing efficiency of the software system also greatly restricts the overall performance of the cloud computing platform. One of the most concerned methods is how to achieve efficient resource scheduling of the cloud computing platform. As one of the most excellent swarm intelligence algorithms, QPSO algorithm has strong optimization ability. It has obvious advantages to apply QPSO algorithm to resource scheduling strategy optimization of cloud computing platform. In this paper, the optimization performance of QPSO algorithm is optimized. By grading the weight coefficients in the average optimal position calculation of particles, the fairness of the average optimal position calculation is improved, and the average optimal position can guide particles to converge to the optimal solution more accurately and quickly. Finally, the optimized QPSO algorithm is applied to the cloud computing task scheduling, and HWQPSO algorithm is used to allocate the tasks of cloud computing to different cloud computing resources reasonably, so as to the overall computing efficiency of the task set is more efficient. The simulation experiment on the CloudSim cloud platform shows that the algorithm in this paper can provide efficient task scheduling strategy for the cloud computing platform, make the resource load of the cloud computing platform more balanced, and improve the computing efficiency of the cloud platform.

2. QPSO Algorithm Model

QPSO algorithm is a kind of PSO algorithm with quantum behavior. Unlike PSO algorithm, the particle in QPSO algorithm is in quantum space, and the particle appears at any point in space according to probability. It is assumed that there are *N* particles representing the solution in the solution space, The position of the *i*-th particle in the *D* dimensional search space is $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. The local optimal position of particle *i* is $pb_i = (pb_{i1}, pb_{i2}, \ldots, pb_{iD})$. The global optimal position of the whole particle swarm is $gb_i = (gb_{i1}, gb_{i2}, \ldots, gb_{iD})$. Using wave function ψ to determine the state of particles in quantum space, the probability of a particle appearing at a certain position in space can be expressed by $|\psi|^2$. If the potential well in *D* dimension is $pb_{id}(t)$ in the *t*-th iteration of particle *i* [13-15].

The wave function ψ(x, t) is used to describe the particle's position and search speed in space, X=(x, y, z), which is a vector, is the position of particles in three-dimensional space, then $|\psi|^2$ is the probability density of particles appearing in three-dimensional space (x, y, z) at time t, as shown in formula (1).

$$|\psi|^2 d_x d_y d_z = Q d_x d_y d_z \qquad (1)$$

In the formula, Q is the probability density function. Q should meet the normalization requirements, such as formula (2).

$$\int_{-\infty}^{+\infty} |\psi|^2 d_x d_y d_z = \int_{-\infty}^{+\infty} Q d_x d_y d_z = 1 \qquad (2)$$

In QPSO algorithm, the state change of each particle in the system follows the Schrodinger equation. At the same time, the δ potential well is introduced into the system, and the potential well is established at p_{id} point. The potential energy function is as formula (3). The steady state Schrodinger equation of the particle in the potential well can be obtained, such as formula (4).

$$V(x) = -\gamma\delta(X - p_{id}) \qquad (3)$$

$$\frac{d^2\psi}{d(X - p_{id})^2} + \frac{2m}{h^2}[E + \gamma\delta(X - p_{id})]\psi = 0 \qquad (4)$$

E is the energy of the particle, h is the Planck constant, and m is the mass of the particle.

The wave function can be obtained by solving the Schrodinger equation, such as formula (5).

$$\psi(X - p_{id}) = \frac{1}{\sqrt{L}} e^{-|X - p_{id}|/L}, L = 1/\beta = h^2 m\gamma \qquad (5)$$

Monte Carlo method is used to sample the particle position randomly, and the position component of the *i*-th particle in the *d* dimension is obtained in the *(t + 1)*-th iteration, as shown in formula(6).

$$x_{id}(t+1) = pb_{id}(t) \pm \frac{L_{id}(t)}{2} ln[\frac{1}{u_{id}(t)}] \qquad (6)$$

In the formula(6), $u_{id}(t) \sim U(0,1)$. The characteristic length of potential well $L_{id}(t)$ is calculated by formula(7).

$$L_{id}(t) = 2\alpha(t)|mb_d(t) - x_{id}(t)| \qquad (7)$$

The *mb* is called the average optimal position, it is the center of the optimal position of all particles. In *D* dimensional space, mb(t) can be calculated by formula(8).

$$mb(t) = (mb_1(t), mb_2(t), ..., mb_D(t)) = \frac{1}{N}\sum_{i=1}^{N} pb_i(t)$$

$$= (\frac{1}{N}\sum_{i=1}^{N} pb_{i1}(t), \frac{1}{N}\sum_{i=1}^{N} pb_{i2}(t), ..., \frac{1}{N}\sum_{i=1}^{N} pb_{iD}(t)) \qquad (8)$$

In the formula(7), α is the contraction expansion coefficient, whose value will directly affect the convergence performance of the algorithm. The value of $\alpha(t)$ in this paper is shown in formula(9).

$$\alpha(t) = 0.5 + \frac{(1-0.5)(t_{max} - t)}{t_{max}} \qquad (9)$$

In the formula(9), *t* is the current number of iterations and t_{max} is the maximum number of iterations. The necessary and sufficient condition for QPSO algorithm to converge to the center of potential well is that the coefficient α <1.78 [16].

The updated formulas of particle's current optimal position pb_i and global optimal position *gb* are shown in formula (10) and formula (11) respectively.

$$pb_i(t+1) = \begin{cases} x_i(t+1) & f[x_i(t+1)] < f[pb_i(t)] \\ pb_i(t) & f[x_i(t+1)] \geq f[pb_i(t)] \end{cases} \qquad (10)$$

$$gb_g(t+1) = argmin\{f[pb_i(t)]\} \qquad (11)$$

In the formula, *f* is the objective function.

In the QPSO algorithm, the particle has only displacement value but no velocity vector in quantum space. The determination of particle position is mainly to obtain the wave function by solving the Schrodinger equation, such as formula (6), to calculate the probability density function and the probability of particle appearing at a certain point in quantum space, and then use Monte Carlo method to randomly sample the particle position to obtain the particle position component, such as formula(6). In the potential well characteristic length $L_{id}(t)$, as shown in formula(7), the average best position *mb* of particles is introduced, as shown in formula (8), to measure the creativity of particles. To improve the ability of interaction between particle swarm and enhance the global search ability of the algorithm. Therefore, the average optimal position *mb*, which is the center of the optimal position of all particles, is one of the core parameters of the whole algorithm.

3. Average Optimal Position of Particles Based on Hierarchical Weight

The biggest difference between the QPSO algorithm and the PSO algorithm is that the particle position update method is different. When updating the particle position, it not only considers the local and global optimal position of the current particle, but also introduces the average optimal position *mb*, which increases the interaction between particles and strengthens the global search ability of particle swarm.

The average optimal position *mb* of the original QPSO algorithm is shown in formula(8). It can be seen that it is the average of the local optimal value of each particle position, which determines the update of particle position. In the calculation process of *mb*, the weight of the local optimal value $pb_i(t)$ of each particle is the same, as shown in formula(12), the proportion of each particle's position in the calculation of *mb* is 1, that is, each particle has the same influence on the final average optimal position *mb* decision. This is not in line with the group intelligent decision-making strategy. In reality, the decision weight of excellent particles is higher than that of inferior particles.

$$mb(t) = (mb_1(t), mb_2(t), ..., mb_D(t)) = \frac{1}{N}\sum_{i=1}^{N} pb_i(t) = \frac{1}{N}\sum_{i=1}^{N}[1 \times pb_i(t)] \qquad (12)$$

Aiming at the problem of unbalanced influence of particles in the calculation of the average optimal position in QPSO algorithm, this paper also introduces a weight factor δ, $\delta_i(t)$ represents the weight of the local optimal value $pb_i(t)$ of the *i*-th particle in the calculation of the average optimal position *mb(t)* of the particle in the *t*-th iteration. After introducing the weight factor δ, the calculation formula(8) of the average optimal position *mb(t)* of the particle can be expressed as the formula(13) [17-20].

$$mb(t) = \frac{1}{N}\sum_{i=1}^{N}[\delta_i(t) \times pb_i(t)] \qquad (13)$$

First, the fitness value $f_i(1 \leq i \leq N, i \in Z)$ of particles is sorted from large to small, and the fitness value after sorting is $f'_j(1 \leq j \leq N, j \in Z)$, $f'_1 \geq f'_2 \geq f'_3 \geq ... \geq f'_N$. According to the fitness value *f*, the particles are divided into $r(1 \leq r \leq N, r \in Z)$ levels, $F_1, F_2, F_3...F_r$. Particles with the same level have the same weight value δ, and the weight values of particles with different levels are $\delta_1, \delta_2, \delta_3, ..., \delta_r, \delta_1 \geq \delta_2 \geq \delta_3 \geq ... \geq \delta_r$. Let δ_r obey the uniform distribution of some subinterval on $[\theta_1, \theta_2]$, and assume that $\theta_1 \leq a_r \leq b_r \leq ... \leq a_3 \leq b_3 \leq a_2 \leq b_2 \leq a_1 \leq b_1 \leq \theta_2$, then $\delta_1 \sim U_1(a_1, b_1), \delta_2 \sim U_2(a_2, b_2), \delta_3 \sim U_3(a_3, b_3), ..., \delta_r \sim U_r(a_r, b_r)$. Then the weight value $\delta_i(t)$ in formula(9) can be calculated by formula(14).

$$\delta_i(t) = \begin{cases} \delta_1(t) \sim U_1(a_1, b_1), & f'_1 \geq f[pb_i(t)] \geq f'_{\lceil (b_1-a_1)\cdot N \rceil} \\ \delta_2(t) \sim U_2(a_2, b_2), & f'_{\lfloor (b_1-b_2)\cdot N \rfloor} \geq f[pb_i(t)] \geq f'_{\lceil (b_1-a_2)\cdot N \rceil} \\ \delta_3(t) \sim U_3(a_3, b_3), & f'_{\lfloor (b_1-b_3)\cdot N \rfloor} \geq f[pb_i(t)] \geq f'_{\lceil (b_1-a_3)\cdot N \rceil} \\ \qquad\qquad \vdots & \\ \delta_r(t) \sim U_r(a_r, b_r), & f'_{\lfloor (b_1-b_r)\cdot N \rfloor} \geq f[pb_i(t)] \geq f'_N \end{cases} \qquad (14)$$

Through the calculation method of formula(10), the particles with the same fitness level will have the uniform distribution value that obeys the corresponding level interval of $[\theta_1, \theta_2]$. This can make the particles with higher fitness occupy a higher weight in the calculation of individual average optimal position *mb(t)*, otherwise, the smaller the weight is, so that the decision-making influence of particles with different fitness can be well balanced. It is beneficial to increase the interaction between particles and enhance the search ability of particle swarm. The comparison between the weight value of the improved algorithm and the original QPSO algorithm is shown in figure 1 and figure 2. In figure 1, the traditional particle swarm optimization method, no matter what the ability of particle optimization is, each particle gets the same weight 1. In figure 2, the improved method in this paper, the stronger the particle optimization ability is, the higher the weight is in the calculation of the average optimal position.

4. HWQPSO for Function Optimization

In order to evaluate the performance of HWQPSO algorithm, it has been applied to some well-known benchmark functions, these functions are used in both reference [10] and [11]. These standard test functions have been adopted in many literatures and are very

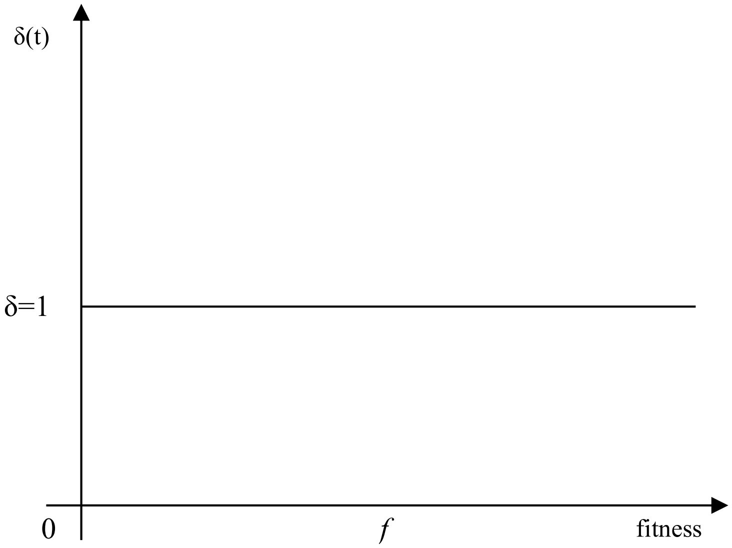


Fig. 1. Weight value graph of QPSO algorithm

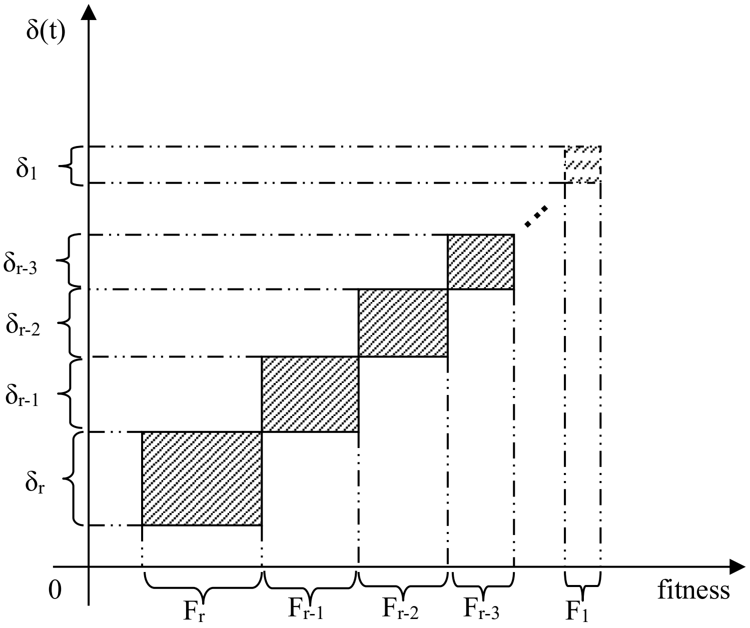


Fig. 2. Weight value graph of HWQPSO algorithm

representative. They can well evaluate the performance of the optimization algorithm. The details of the benchmark functions are given in Table 1, including function name, specific formula, range min value and search ability. These benchmark functions are minimization problems and the global best value for all these functions is zero. The experimental results of HWQPSO are compared with those of QPSO [4], DWC-QPSO [10] and LTQPSO [11]. The parameters of QPSO, DWC-QPSO and LTQPSO select original paper parameters. The weight parameter of HWQPSO algorithm is shown in Table 2. r=4, θ_1=0.5, θ_2=1.5. we compare the convergence rate of the four algorithms in the process of 8 standard function tests, in which the average optimal value of the objective function changes with the number of iterations. Figures 3 $\sim$ 10 show average of convergence curves for 50 runs of QPSO, DWC-QPSO, LTQPSO and HWQPSO under the condition of N = 40, D = 30, M = 2000, but the Schaffer function under the condition of N = 40, D = 2, M = 2000. In order to show the results more intuitively, when drawing the contrast curves, the log value of the fitness value is calculated on the vertical axis.

Table 1. Mathematical benchmark functions

Function	Formulation	Range	Min value	Search ability
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	(-100,100)	0	Local
DeJong's	$f_2(x) = \sum_{i=1}^{D} i \cdot x_i^4$	(-100,100)	0	Local
Rosenbrock	$f_3(x) = \sum_{i=1}^{D-1}(100 \cdot (x_{i+1}$ $-x_i^2)^2 + (x_i - 1)^2)$	(-5.12,5.12)	0	Global/ Local
Griewank	$f_4(x) = \sum_{i=1}^{D} x_i^2/4000$ $- \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	(-600,600)	0	Global
Rastrigin	$f_5(x) = \sum_{i=1}^{D}(x_i^2 - 10 \cdot \cos(2\pi x_i) + 10)$	(-5.12,5.12)	0	Global
Ackley	$f_6(x) = -20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2})$ $-exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	(-32,32)	0	Global
Schaffer	$f_7(x) = 0.5 + ((\sin\sqrt{x_1^2 + x_2^2})^2$ $-0.5)/((1 + 0.001(x_1^2 + x_2^2))^2)$	(-2.048,2.048)	0	Global
Schwefel	$f_8(x) = 418.9829D - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	(-500,500)	0	Global

From the experimental results curves in figure 3 to figure 10, it can be seen that the HWQPSO algorithm has the best optimization ability and search stability compared with the other three algorithms, Among them, HWQPSO algorithm performs best in the Sphere and DeJong's function tests, and worst in Rosenbrock function tests. Among the eight

Table 2. Particle weight distribution table

The value of r	Distribution interval	Distribution ratio
r=1	$\delta_1 \sim U_1(1.4, 1.5)$	10%
r=2	$\delta_2 \sim U_2(1.2, 1.4)$	20%
r=3	$\delta_3 \sim U_3(0.9, 1.2)$	30%
r=4	$\delta_4 \sim U_4(0.5, 0.9)$	40%

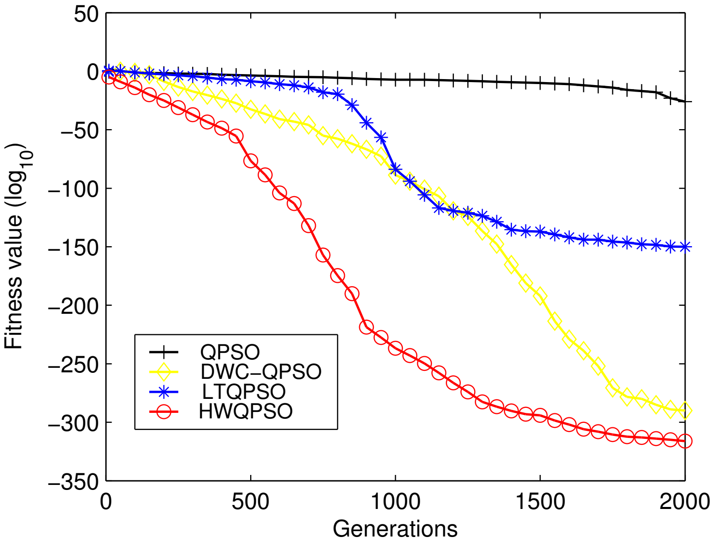


Fig. 3. Convergence curve of the Sphere function

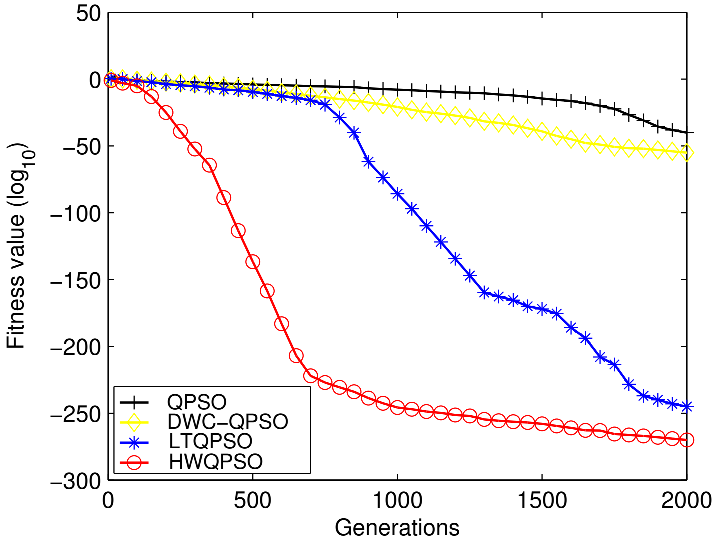


Fig. 4. Convergence curve of the DeJong's function

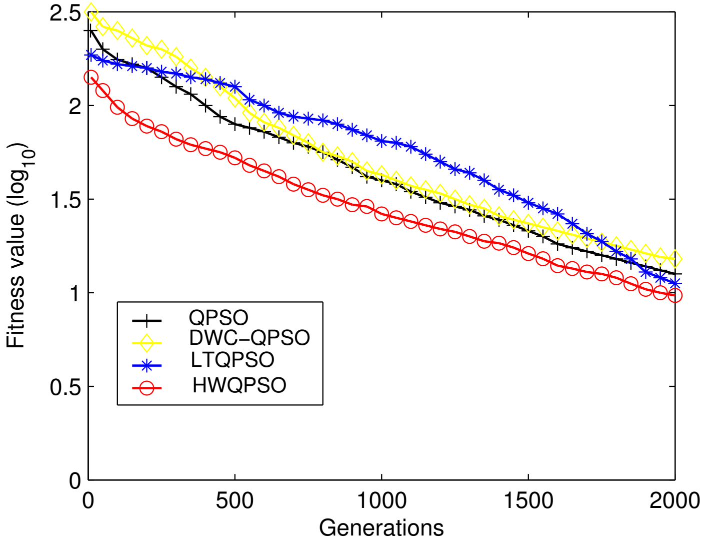


Fig. 5. Convergence curve of the Rosenbrock function

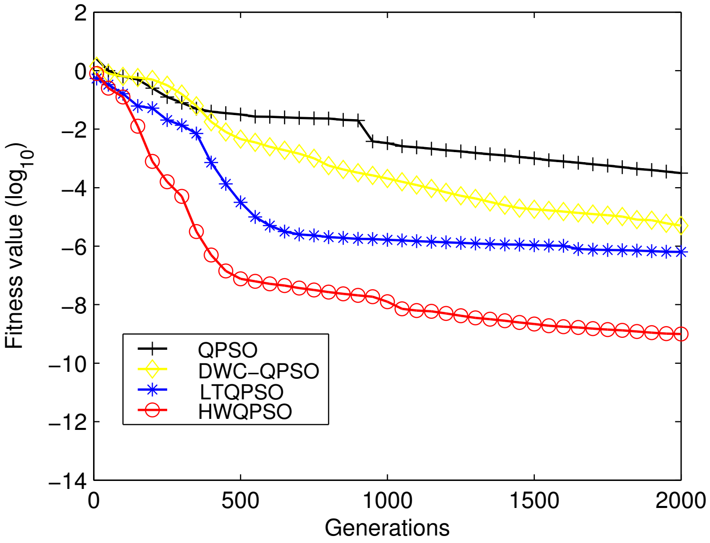


Fig. 6. Convergence curve of the Griewank function

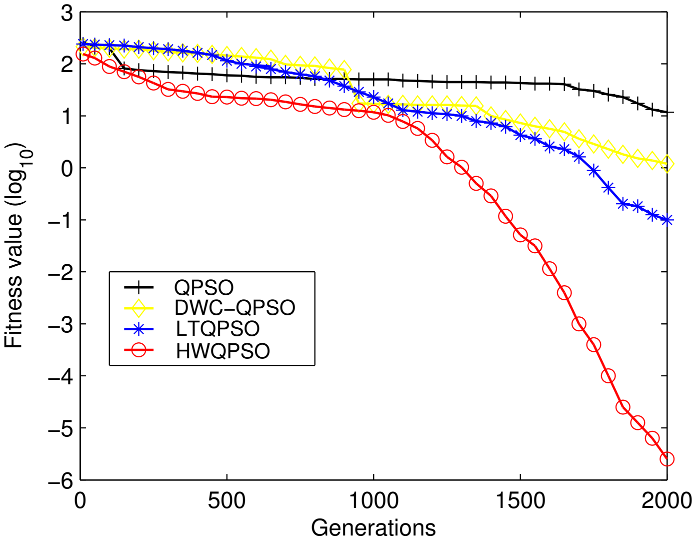


Fig. 7. Convergence curve of the Rastrigin function

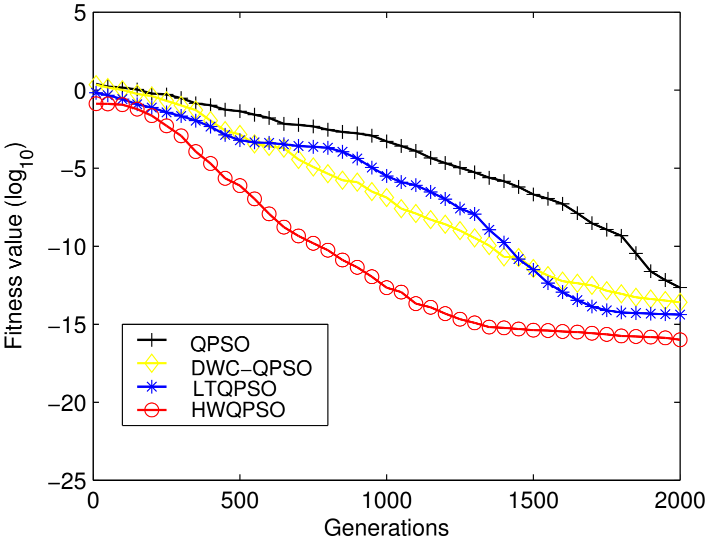


Fig. 8. Convergence curve of the Ackley function

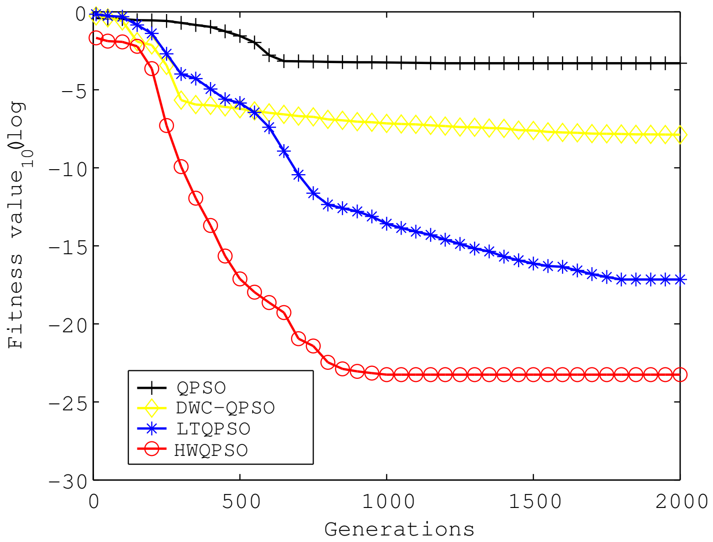


Fig. 9. Convergence curve of the Schaffer function

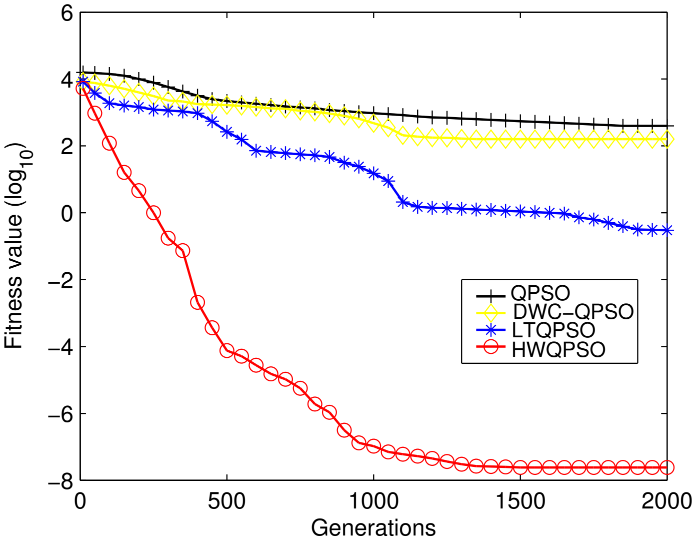


Fig. 10. Convergence curve of the Schwefel function

standard test functions, Sphere and DeJong's are unimodal functions, which are generally used to test the local search ability of the algorithm. From the comparison of the experimental results on these two functions, HWQPSO algorithm has better local search ability and search stability than the other three algorithms, but from the average optimal value data, DeJong's function has higher search accuracy and stability than Sphere function, and DWC-QPSO algorithm also shows the performance close to HWQPSO algorithm on Sphere function. Rosenbrock function is usually used to test the local and global search ability of optimization algorithm. Each contour line of Rosenbrock function is approximately parabola shaped, and its global minimum value is located in the parabola shaped Valley, which is easy to find, but because the value in the valley changes little, it is very difficult to find the global minimum value. So in eight test functions, the results of four algorithms on Rosenbrock function are the worst, but the results of hwqpso algorithm in this paper are still better than the other three algorithms. Griewank, Rastrigin, Ackley, Schaffer and Schwefel functions are nonlinear multi peak functions, which are usually used to test the global search ability of optimization algorithms. The experimental results show that the optimization effect of the four algorithms on these five functions is not as good as that of sphere and DeJong's unimodal functions, but better than Rosenbrock functions. The global optimization results of HWQPSO are better than those of other three algorithms. Among them, the search performance of Rastrigin and Schwefel functions is better than that of the other three functions.

On the other hand, for HWQPSO algorithm, it can be found that the fitness values of the HWQPSO algorithm are lower than those of the QPSO, DWC-QPSO and LTQPSO in 2000 iterations of 8 standard functions. That is to say, the red curve representing the iterative fitness values of the HWQPSO algorithm in 8 figures is always lower than the other three curves. At the same number of iterations, the local solution found by the HWQPSO algorithm is better than the other three algorithms. Especially in Sphere, DeJong's, Griewank, Schaffer, and Schwefel functions, it shows that the local optimization accuracy of the HWQPSO algorithm is better, and each iteration can find a better solution than the other three algorithms.

At the same time, the lower red curve also means that the convergence speed of the HWQPSO algorithm is faster than other three algorithms. The optimal solution of the HWQPSO algorithm is closer to the optimal value, it will quickly approach the optimal solution. For example, when 500 iterations in figure 10, the fitness value of the HWQPSO algorithm is 10^{-4}, which converges to the optimal solution 0 faster than other three algorithms.

It can also be found from the experimental results that the HWQPSO algorithm has the strongest global optimization ability, which is also higher than the other three algorithms. That is to say, at the abscissa 2000 point in the experimental result graph, the HWQPSO algorithm has the least fitness value compared with the other three algorithms in 8 standard functions, that is, the solution is optimal.

All these are mainly due to the particles with higher fitness value of the HWQPSO algorithm, and the larger the calculated weight of the average optimal position, which makes the average optimal position tend to be excellent particles and find better solutions under the leadership of excellent particles.

To sum up, through the experiments of the QPSO, DWC-QPSO, LTQPSO and HWQPSO algorithm on 8 standard test functions, the results show that the HWQPSO algorithm proposed in this paper has more advantages than the QPSO, DWC-QPSO and LTQPSO in local accuracy, convergence speed and global search ability.

5. HWQPSO for Task Scheduling in Cloud Computing

Assuming that there are *H* computing resources available in the cloud computing platform, the computing resources are the position *X* of particles in the space search, that is, the resource set $X = \{x_1, x_2, x_3, \ldots, x_H\}$; There are *D* computing task requests in the cloud computing system at a certain time, Task set $S = \{s_1, s_2, s_3, \ldots, s_D\}$; The matrix *T* is used to represent the time when different tasks calculate data on different resources, such as formula (15), that is t_{ij} represents the time required for the *i*-th task to complete data processing on the *j*-th calculation resource, $1 \leq i \leq D, 1 \leq j \leq H$. In the process of searching space, the position of the *i*-th particle is $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$, $1 \leq i \leq D$, and the value of dimension *D* is the number of tasks in the cloud computing platform. x_{id} means that task *d* is scheduled to execute on resource x_i, and its matrix representation is as shown in formula (16). *X* is a $0-1$ matrix, when $x_{ij} = 1$, it means that the data to be processed by the *i*-th task is processed by the *j*-th computing resource; when x_{ij}=0, it means that the data to be processed by the *i*-th task is not processed by the *j*-th computing resource[21-24].

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & ... & t_{1H} \\ t_{21} & t_{22} & t_{23} & ... & t_{2H} \\ t_{31} & t_{32} & t_{33} & ... & t_{3H} \\ ... & ... & ... & ... & ... \\ t_{D1} & t_{D2} & t_{D3} & ... & t_{DH} \end{bmatrix} \quad (15) \quad X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & ... & x_{1D} \\ x_{21} & x_{22} & x_{23} & ... & x_{2D} \\ x_{31} & x_{32} & x_{33} & ... & x_{3D} \\ ... & ... & ... & ... & ... \\ x_{N1} & x_{N2} & x_{N3} & ... & x_{ND} \end{bmatrix} \quad (16)$$

In the task scheduling method in this paper, it is expected that all computing tasks in the task set will be completed, and the less the total time T_{total} is, the better. From the above analysis, it can be concluded that the total time T_{total} taken by all calculation tasks is shown in formula (17).

$$T_{total} = max_{j=1}^{H} \sum_{i=1}^{D} x_{ij} \times t_{ij} \qquad (17)$$

Through the above optimization of QPSO algorithm, it can be seen that the higher the fitness of particles, the higher the proportion of particles should be when calculating the average optimal position mb of particles. Therefore, the fitness function f of particles is defined as shown in formula (18).

$$f = \frac{1}{T_{total}} = \frac{1}{max_{j=1}^{H} \sum_{i=1}^{D} x_{ij} \times t_{ij}} \qquad (18)$$

It can be seen from formula (14) that the longer the total execution time of the calculation task set is, the smaller the value of fitness f will be, and the lower the calculation efficiency of the cloud computing platform will be; on the contrary, the shorter the total execution time of the calculation task set is, the larger the value of fitness f will be, and the higher the processing efficiency of the cloud computing platform will be, which meets the processing efficiency expectation of the cloud computing platform.

6. Experiments and Analysis

In this paper, Cloudsim4.0, a cloud computing simulation tool, is used as the experimental platform, and its parameters are shown in Table 3. The main classes and methods of the simulation process and its implementation based on Java development environment are shown in Figure 11. According to the modeling requirements of this paper, the task model and virtual machine model are adjusted on the platform, and the DatacenterBroker and Cloudlet classes are rewritten. The HWQPSO algorithm is implemented in DatacenterBroker, and the standard QPSO algorithm, DWC-QPSO algorithm and LTQPSO algorithm are reproduced. In order to test the search performance of this algorithm in cloud computing task scheduling, this section will compare the application of QPSO, DWC-QPSO, LTQPSO and HWQPSO algorithms in cloud platform task scheduling, and verify the application performance of this algorithm in cloud computing task scheduling from the following two perspectives [25-28].

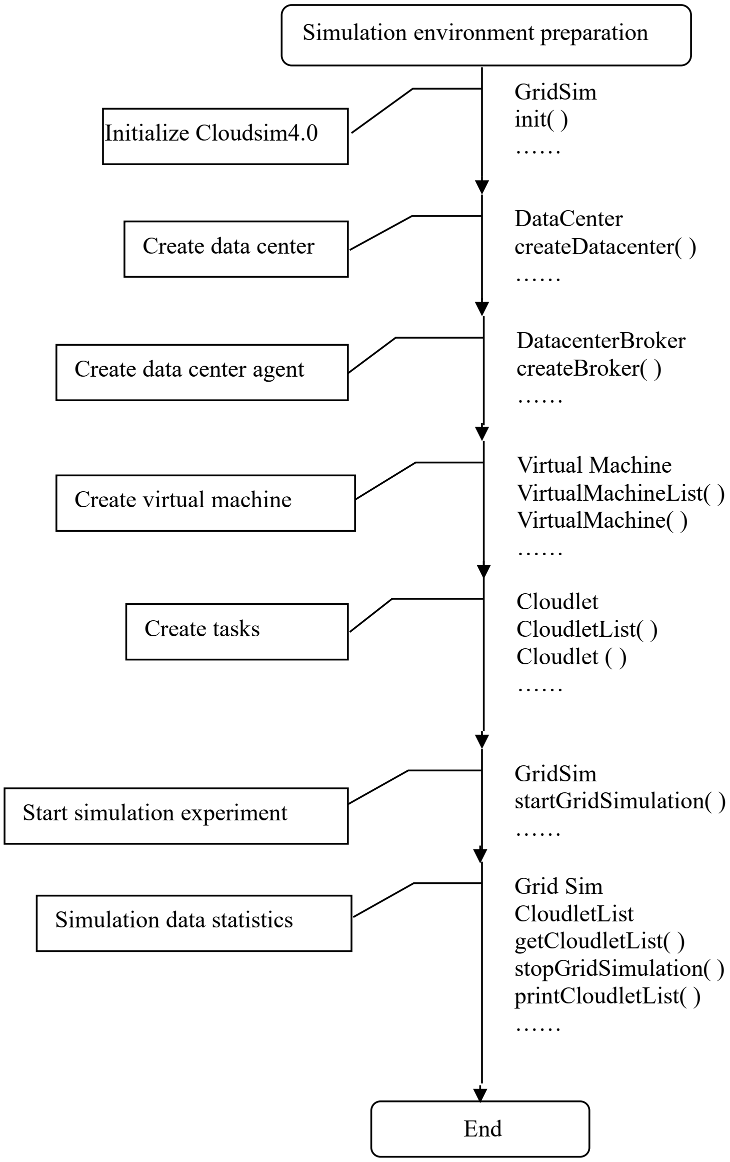


Fig. 11. Cloudsim4.0 simulation experiment process

Table 3. Cloud Sim simulator parameter list

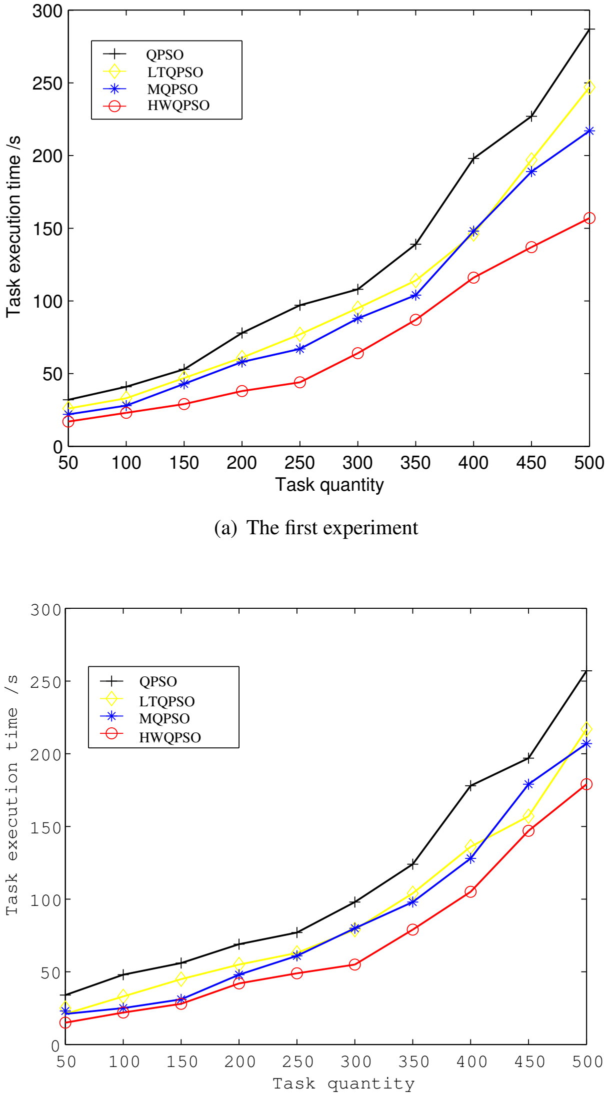
Type	Parameters	Value
Datacenter	number of Datacenter	6
	number of Host per Datacenter	2-5
	type of Manager	Space_shared/ Time_shared
Virtual Machine(VM)	total number of VMs	30
	number of PE per VM	4-12
	MIPS of PE (processing element)	300-1500 (MIPS)
	VM memory	512-2048(MB)
	Bandwidth	500-1000 bit
	Type of Manager	Time_shared
Task	Total number of task	50-500
	Length of task	5000-15000MI(Million Instruction)
	Number of PEs requirement	1-6

6.1. Task Execution Time Comparison Experiment

This experiment is based on the task scheduling schemes of QPSO, DWC-QPSO, LTQPSO, HWQPSO. The experiment is carried out under different scale tasks. The number of tasks is between [50,500], and the increment is 50. By randomly selecting the cloud computing resource configuration parameters, the execution time of four different task scheduling strategies was compared for four times, and the results are shown in figure 12.

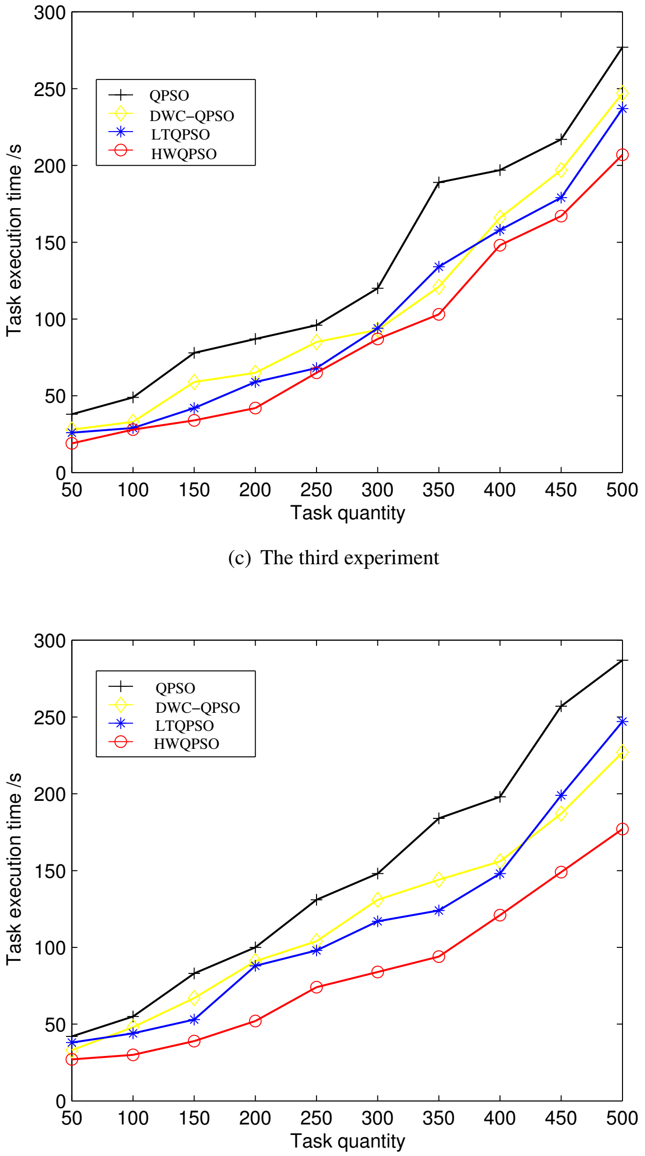
According to the experimental results curves in figure 12, the red curve is the task scheduling time curve of the HWQPSO algorithm. The horizontal axis is the number of tasks, and the vertical axis is the time spent scheduling tasks. From the overall trend, the original QPSO algorithm takes the most time when scheduling the same number of tasks, while the HWQPSO algorithm in this paper takes the least time, and the DWC-QPSO and LTQPSO algorithm have their own advantages and disadvantages. As shown in figure 12(a) (d), when scheduling the same number of tasks, the red curve representing the HWQPSO algorithm is much lower than the black curve representing QPSO algorithm. The blue curve representing LTQPSO algorithm and the Yellow curve representing DWC-QPSO algorithm are interwoven, with high and low among them. For example, in figure 12(d), when the number of tasks is 350, the scheduling time of DWC-QPSO algorithm is more than that of LTQPSO algorithm, but when the number of tasks is 450, the scheduling time of DWC-QPSO algorithm is less than that of LTQPSO algorithm.

When the number of tasks is small, the difference of task scheduling time between the four algorithms is small, because the QPSO algorithm has strong optimization ability.



(a) The first experiment

(b) The second experiment



(c) The third experiment

(d) The fourth experiment

Fig. 12. Execution time comparison

When the small-scale task set scheduling is optimized, the optimal scheduling scheme can be found quickly. However, with the increase of task scale, the task scheduling efficiency of the HWQPSO algorithm is significantly higher than the other three. For example, when the number of tasks is 100, the task scheduling time gap of the four algorithms is much smaller than that of 500 tasks. This is because the algorithm in this paper improves the decision weight of excellent particles, so that all particles quickly approach to the optimal particles, and finally quickly converge to the optimal solution, that is to find the optimal scheduling scheme.

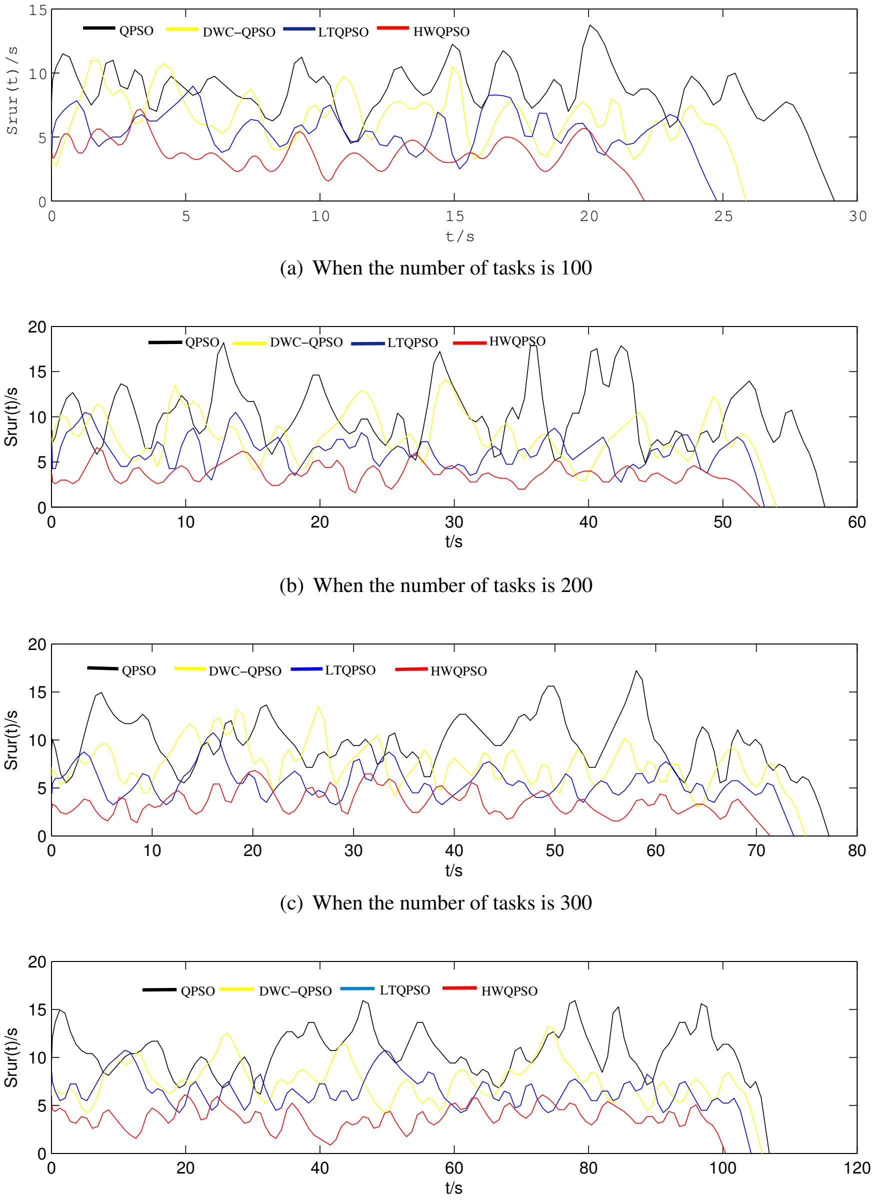
6.2. Computing Resource Load Comparison Experiment

This experiment is based on the task scheduling schemes of QPSO, DWC-QPSO, LTQPSO, HWQPSO. When the number of tasks is 100, 200, 300 and 400 respectively, the resource load of the four scheduling algorithms are compared. In the experiment, for each virtual machine, the time needed to complete all tasks scheduled to the virtual machine is used as the load measurement of the virtual machine. For the convenience of comparison, we calculate the standard deviation of computing resource load of all virtual machines in the cloud platform to describe the balance of computing resource load of the cloud platform at this moment. Assuming that there are N_{vm} virtual machines in the cloud platform in the current experiment, the computing resource load of the *i*-th virtual machine at time *t* is $u_i(t)$, then at time *t*, the calculation of the standard deviation $S_{rur}(t)$ of the computing resource load of all virtual machines in the cloud platform is shown in formula (19) and formula (20). The comparison experiment results of the cloud platform computing resource load of the four scheduling algorithms are shown in figure 13.

$$S_{rur}(t) = \sqrt{\frac{1}{N_{vm}} \sum_{i=1}^{N_{vm}} (u_i(t) - \overline{u(t)})^2} \qquad (19)$$

$$\overline{u(t)} = \frac{1}{N_{vm}} \sum_{i=1}^{N_{vm}} u_i(t) \qquad (20)$$

When the $S_{rur}(t)$ of computing resource load of all virtual machines is smaller, it means that the load of each virtual machine is relatively more balanced, and vice versa. According to the experimental results in Figure 13, the red curve is the load balance curve of the HWQPSO algorithm in resource scheduling, the standard deviation of computing resource load of all virtual machines in the cloud platform fluctuates greatly under different tasks of the four algorithms. As shown in Figure13(a)$\sim$(d), the curves corresponding to the four algorithms are interleaved in varying degrees. However, from the overall trend analysis, the standard deviation of the HWQPSO algorithm in this paper is smaller than that of the other three algorithms, which shows that the load of each virtual machine is more balanced than that of the other three algorithms. And the standard deviation of computing resource load using the original QPSO algorithm is the largest, and the computing resource load is the most unbalanced, followed by the LTQPSO and DWC-QPSO algorithm. For example, in figure 13(a)$\sim$(d), the overall trend of the black curve representing the load standard deviation of all virtual machine resources scheduled by QPSO algorithm is at the top, the red curve representing HWQPSO algorithm is at the bottom, and the blue



(a) When the number of tasks is 100

(b) When the number of tasks is 200

(c) When the number of tasks is 300

(d) When the number of tasks is 400

Fig. 13. Comparison of computing resource load

and yellow curves representing the LTQPSO and the DWC-QPSO algorithm tend to be in the middle. At the same time, in figure 13(a)∼(d), it can be seen from the horizontal axis that HWQPSO scheduling time is the shortest, this also shows that HWQPSO algorithm has better optimization ability. And the red curve fluctuates the least, it shows that HWQPSO is more stable. All these benefit from the HWQPSO algorithm average optimal location classification weight strategy, improve the optimization accuracy of the algorithm and make the computing resources of each virtual machine can be better used, thus play a role of balancing the computing resources load.

7. Conclusion

In this paper, an average optimal position calculation method of the QPSO algorithm is proposed, which is based on the classification of particle fitness value, and it is used in cloud computing task scheduling. The selection of the average optimal position in the QPSO algorithm determines the global search ability and the final convergence speed of the algorithm. By setting high-level particles with high weight, we can improve the discourse power of excellent particles in the process of optimization, so that particles can quickly approach the optimal solution, to improve the search ability and efficiency of the algorithm. In this paper, five standard test functions are selected to test QPSO, DWC-QPSO, LTQPSO and HWQPSO. The experimental results show that the convergence accuracy and speed of the HWQPSO algorithm proposed in this paper are higher than those of the other three algorithms. At the same time, the HWQPSO algorithm proposed in this paper is applied to the task scheduling of the cloud computing platform. The performance of the HWQPSO algorithm proposed in this paper is tested by comparing the efficiency of the four algorithms QPSO, DWC-QPSO, LTQPSO and HWQPSO in the CloudSim4.0 simulation experiment platform. In the application, when scheduling the same number of tasks, the algorithm in this paper takes shorter time than the other three algorithms, and the load of computing resources is more balanced, so the efficiency of cloud platform is significantly improved. Experiments and application results show that the average optimal position calculation method based on particle fitness value classification improves the local search accuracy and global search ability of the QPSO algorithm, and the search stability is also improved.

Acknowledgment. This work was the achievements of the project" earth thesis project "of Guizhou Education University; And it was supported by the Youth Science and Technology Talents Development Project of Guizhou Education Department in 2018(No. Qian Jiao He KY[2018]257); Reform Project of Teaching Content and Course System in Guizhou Province in 2017(2017520065); 2016 Science and Technology Foundation Project of Guizhou Province(No. Qian Ke He Ji Chu [2016]1114); Key Disciplines of Guizhou Province-Computer Science and Technology (ZDXK [2018]007); Key Supported Disciplines of Guizhou Province-Computer Application Technology (No.QianXueWeiHeZi ZDXK[2016]20).

References

1. Van Den Bergh Frans. *An analysis of particle swarm optimizers*. University of Pretoria, 2002.

2. Yangyang Li, Xiaoyu Bai, Licheng Jiao, and Yu Xue. Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. *Applied Soft Computing*, 56:345–356, 2017.
3. Weiping Ding, Chin Teng Lin, Senbo Chen, Xiaofeng Zhang, and Bin Hu. Multiagent-consensus-mapreduce-based attribute reduction using co-evolutionary quantum pso for big data applications. *Neurocomputing*, page S0925231217311876, 2017.
4. Jun Sun, Bin Feng, and Wenbo Xu. Particle swarm optimization with particles having quantum behavior. In *Congress on Evolutionary Computation*, 2004.
5. Jun Sun, Wenbo Xu, and Bin Feng. A global search strategy of quantum-behaved particle swarm optimization. In *Proc IEEE Conference on Cybernetics & Intelligent Systems*, 2004.
6. Guoqiang Liu, Weiyi Chen, Huadong Chen, and Jiahui Xie. A quantum particle swarm optimization algorithm with teamwork evolutionary strategy. *Mathematical Problems in Engineering*, 2019(8):1–12, 2019.
7. Anupam Trivedi, Dipti Srinivasan, Subhodip Biswas, and Thomas Reindl. Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem. *Swarm & Evolutionary Computation*, 23:50–64, 2015.
8. Qianqian Zhang, Shifeng Liu, Daqing Gong, Hankun Zhang, and Qun Tu. An improved multi-objective quantum-behaved particle swarm optimization for railway freight transportation routing design. *IEEE Access*, 7:157353–157362, 2019.
9. Zhen Lun Yang, Angus Wu, and Hua Qing Min. An improved quantum-behaved particle swarm optimization algorithm with elitist breeding for unconstrained optimization. *Comput Intell Neurosci*, 2015:1–12, 2015.
10. Tao Wu, Xi Chen, and Yusong Yan. Study of on-ramp pi controller based on dural group qpso with different well centers algorithm. *Mathematical Problems in Engineering*, 2015(PT.5):814871.1–814871.10, 2015.
11. Tao Xue, Renfu Li, Myongchol Tokgo, Junchol Ri, and Gyanghyok Han. Trajectory planning for autonomous mobile robot using a hybrid improved qpso algorithm. *Soft Computing*, 2017:2421–2437, 2017.
12. Wei Chen, Hong Yang, and Yifei Hao. Scheduling of dynamic multi-objective flexible enterprise job-shop problem based on hybrid qpso. *IEEE Access*, PP(99):1–1, 2019.
13. Lin Lin, Feng Guo, Xiaolong Xie, and Bin Luo. Novel adaptive hybrid rule network based on ts fuzzy rules using an improved quantum-behaved particle swarm optimization. *Neurocomputing*, 149(pt.b):1003–1013, 2015.
14. W. J. Chen, C. X. Lin, Y. T. Chen, and J. R. Lin. Optimization design of a gating system for sand casting aluminium a356 using a taguchi method and multi-objective culture-based qpso algorithm. *Mrs Bulletin*, 40(1):46–52, 2011.
15. Wen Jong Chen, Chuan Kuei Huang, Qi Zheng Yang, and Yin Liang Yang. Optimal prediction and design of surface roughness for cnc turning of al7075-t6 by using the taguchi hybrid qpso algorithm. *Transactions of the Canadian Society for Mechanical Engineering*, 40(5):883–895, 2016.
16. Jun Sun, Xiaojun Wu, Vasile Palade, Wei Fang, Choi Hong Lai, and Wenbo Xu. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Information Sciences*, 193(none):81–103, 2012.
17. Chunming Zhang, Yongchun Xie, Da Liu, and Li Wang. Fast threshold image segmentation based on 2d fuzzy fisher and random local optimized qpso. *IEEE Transactions on Image Processing*, 26(3):1355–1362, 2017.
18. Tianyu Liu, Licheng Jiao, Wenping Ma, Jingjing Ma, and Ronghua Shang. Cultural quantum-behaved particle swarm optimization for environmental/economic dispatch. *Applied Soft Computing*, 48:597–611, 2016.
19. Rajashree Nayak and Dipti Patra. An edge preserving ibp based super resolution image reconstruction using p-spline and mucso-qpso algorithm. *Microsystem Technologies*, 23(3):1–17, 2016.

20. Wei Zhang, Weifeng Shi, and Jinbao Zhuo. Bdi-agent-based quantum-behaved pso for shipboard power system reconfiguration. *International Journal of Computer Applications in Technology*, 55(1), 2017.
21. P. Sivakumar, S. Kalaiyarasi, and S. Shilpa. Qpso based load balancing mechanism in cloud environment. *ICSCAN*, 2019.
22. Zhong Kai Feng, Wen Jing Niu, and Chun Tian Cheng. Multi-objective quantum-behaved particle swarm optimization for economic environmental hydrothermal energy system scheduling. *Energy*, 131(Jul.15):165–178, 2017.
23. Obaid Ur Rehman, Shiyou Yang, Shafiullah Khan, and Sadaqat Ur Rehman. A quantum particle swarm optimizer with enhanced strategy for global optimization of electromagnetic devices. *IEEE Transactions on Magnetics*, PP(99):1–4, 2019.
24. R. Chakraborty, R. Sushil, and M. L. Garg. Icqpso-based multilevel thresholding scheme applied on colour image segmentation. *IET Signal Processing*, 3(13):387–395, 2019.
25. Xingquan Zuo, Guoxiang Zhang, and Tan Wei. Self-adaptive learning pso-based deadline constrained task scheduling for hybrid iaas cloud. *IEEE Transactions on Automation Science & Engineering*, 11(2):564–573, 2014.
26. S. Zhan and H. Huo. Improved pso-based task scheduling algorithm in cloud computing. *Journal of Information & Computational Science*, 9(13):3821–3829, 2012.
27. Jena and K. R. Multi objective task scheduling in cloud environment using nested pso framework. *Procedia Computer Science*, 57:1219–1227, 2015.
28. Mohammad Masdari, Farbod Salehi, Marzie Jalali, and Moazam Bidaki. A survey of pso-based scheduling algorithms in cloud computing. *Journal of Network & Systems Management*, 25(1):122–158, 2017.

Guolong Yu from Guiyang Guizhou Province, received Master degree in 2010 from Lanzhou University of Technology, China. Now, he is an associate professor in college of mathematics and big data, Guizhou Education University. His research interests include computational intelligence and big data.

Yong Zhao (Corresponding Author) from Shenzhen Guangdong Province, received Ph.D. degree in 1991 from Southeast University, China, and Postdoctoral in 2000 from Concordia University, Dept. of ECE. Montreal, Canada. Now, he is a professor in college of information engineering, Peking University. His research interests include deep learning and machine vision.

Zhongwei Cui from Guiyang Guizhou Province, received Ph.D. degree in 2019 from Guizhou University, China. Now, he is an associate professor in college of mathematics and big data, Guizhou Education University. His research interests include internet of things and machine vision.

Yu Zuo from Guiyang Guizhou Province, received Master degree in 2010 from Guizhou University, China. Now, he is a professor in college of mathematics and big data, Guizhou Education University. His research interests include machine vision and big data.

Received: February 23, 2020; Accepted: October 15, 2020.

Computer Science and Information Systems 18(1):213–230 https://doi.org/10.2298/CSIS200428045H

Convexity of hesitant fuzzy sets based on aggregation functions

Pedro Huidobro[1,3], Pedro Alonso[2], Vladimír Janiš[3], and Susana Montes[1]

[1] Dept. of Statistics and Operational Research
University of Oviedo, Spain
{huidobropedro,montes}@uniovi.es
[2] Dept. of Mathematics
University of Oviedo, Spain
palonso@uniovi.es
[3] Dept. of Mathematics
Matej Bel University, Slovakia
vladimir.janis@umb.sk

Abstract. Convexity is one of the most important geometric properties of sets and a useful concept in many fields of mathematics, like optimization. As there are also important applications making use of fuzzy optimization, it is obvious that the studies of convexity are also frequent. In this paper we have extended the notion of convexity for hesitant fuzzy sets in order to fulfill some necessary properties. Namely, we have found an appropriate definition of convexity for hesitant fuzzy sets on any ordered universe based on aggregation functions such that it is compatible with the intersection, that is, the intersection of two convex hesitant fuzzy sets is a convex hesitant fuzzy set and it fulfills the cutworthy property.

Keywords: hesitant fuzzy set, alpha-cut, aggregation function, convexity.

1. Introduction

Convexity is a basic mathematical notion that has been used to analyse many different problems. Its practical applications in several areas are very important, like optimization [20], image processing [32], robotics [19] or geometry [15], among many others.

Since most of practical problems include approximate information, fuzzy convexity has been studied in deep in the literature. Thus, several types of convexity of fuzzy sets were studied by different authors (see, for instance, Ammar and Metz [2], Diaz et al. [10], Ramik and Vlach [25], Sarkar [29], Syau and Lee [31] and Yang [38]).

The necessity of dealing with imprecision in real world problems has been a long-term research challenge that has originated different extensions of fuzzy sets. A special case of type-2 fuzzy sets are the hesitant fuzzy sets. They can be considered as an extension of fuzzy sets different from Atanassov's intuitionistic fuzzy sets ([3]) or interval-valued fuzzy sets (introduced independently by Zadeh [40], Grattan-Guiness [11], Jahn [12], Sambuc [28] in the seventies). Hesitant fuzzy sets can be useful to deal with situations where the previous tools are not so efficient as, for instance, the modeling of an evaluation by a group of experts, when it is not possible or easy to obtain a consensus to unify the different opinions. Hesitant fuzzy sets were formally introduced by Torra [33], but the idea behind this concept was already considered previously, as we can see in Grattan-Guinness [11]. Although their definition is relatively new, it has attracted very quickly the attention of many researchers, since they could see the high potential of them for applications, especially in decision making, as we can see in [34,35,37,41]. Perhaps by this reason, several concepts, tools and trends related to this extension have to be studied. Taking into account the previous comments, we are specially interested in the concept of convex hesitant fuzzy sets. As far as we know, the first attempt to define convexity for hesitant fuzzy sets has been done by Rashid and Beg in 2016 [26]. That definition had some problems, which were solved by Janiš et al. in 2018 [14]. These two definitions seem to be quite different. However, we can notice that they are really related, since both of them are based on aggregation functions. Thus, Rashid and Beg considered the arithmetic mean and Janiš et al. considered the classical t-conorm of the maximum. Then, both definitions can be considered as particular cases of a general convexity based on aggregation functions. The main aim of this paper is to introduce a general definition and study its properties. In particular, we are going to study in depth the preservation of convexity for alpha-cuts (the cutworthy property) and under intersections, since the first one is a very important property in fuzzy set theory and the second one is a necessary property in many applications, as optimization. Thus, we will try to characterize the behavior of aggregation functions with respect to both properties.

The remainder of this paper is organized as follows. In Section 2, some basic concepts about convexity for fuzzy sets and hesitant fuzzy sets are recalled and the notation is fixed. Section 3 is devoted to the new definition of convexity for hesitant fuzzy sets based on an aggregation function with a detailed study of the preservation of convexity under intersections. In Section 4 we present an example from the area of optimization. Finally, some conclusions and open problems are formulated in Section 5.

2. Basic concepts

It is well-known that for any nonempty set X, usually called the universe, Zadeh defined a fuzzy set A in X by means of the map $\mu_A : X \to [0,1]$, which is said to be the membership function of A (see [39]). Thus, a way to describe the fuzzy set could be $A = \{\langle x, \mu_A(x)\rangle : x \in X\}$.

For any $\alpha \in (0,1]$, a crisp subset of X is associated to A as follows: $A_\alpha = \{x \in X : \mu_A(x) \geq \alpha\}$. This set is called the α-cut of A and the collection of all the alpha-cuts totally characterizes the fuzzy set.

A particular case of fuzzy sets are the convex fuzzy sets. A crisp subset A of a linear space X is convex if and only if $\lambda x + (1-\lambda)y \in A$ for any $x, y \in A$ and for any $\lambda \in [0,1]$ (for a detalied study on convex set see e.g. [18]). From this definition, a natural extension for fuzzy sets could be to require that the fuzzy set A fulfills the property: $\mu_A(\lambda x + (1-\lambda)y) \geq \lambda\mu_A(x) + (1-\lambda)\mu_A(y)$, for all $x, y \in X$ and for all $\lambda \in [0,1]$. However, this definition was not considered from the beginning, since as already Zadeh noticed in [39], there is not an equivalence between the convexity of the alpha-cuts and the convexity of the fuzzy set.

In order to solve this problem, the first definition of convex fuzzy set considered in that paper was that

$$\mu_A(\lambda x + (1-\lambda)y) \geq \min\{\mu_A(x), \mu_A(y)\}$$

for any $x, y \in X$ and any $\lambda \in [0,1]$ (see [39]). It is known that A is a convex fuzzy set if and only if A_α is a convex crisp set, for any $\alpha \in (0,1]$, that is, the cutworthy property is fulfilled. Apart from that, this concept is preserved by the intersection, that is, if A and B are two convex fuzzy sets, then $A \cap B$ is a convex fuzzy set, with the classical definition of intersection introduced by Zadeh. Another advantage of this definition is that the addition that appears on the right side of the inequality could make no sense when working in a more general environment (e.g. lattice-valued fuzzy sets), where such operation is not defined in general.

As we commented at the introduction, several generalizations of fuzzy sets have been considered in the literature. In this paper we are interested in the class of hesitant fuzzy sets. Now, instead of a single number, the membership function returns a set of membership values for each element in the domain. More precisely:

Definition 1 *[33,36] Let X be a universe. A hesitant fuzzy set on X is defined by means of a function $h_A : X \to \mathcal{P}([0,1])$, where $\mathcal{P}([0,1])$ denotes the power set of the interval $[0,1]$, such that for any element of X it returns a subset of the unit interval $[0,1]$. This can be represented by*

$$A = \{\langle x, h_A(x)\rangle : x \in X\},$$

where $h_A(x)$ is a set of values in $[0,1]$, denoting the possible membership degrees of the element $x \in X$ to the set A.

In this definition, any subset of the interval $[0,1]$ could be the membership degree for an element in X. However, some particular cases of specific subsets are the most important in the literature. Thus, for instance, the case of interval-valued hesitant fuzzy sets has been studied lately (see, for instance, [7]). However, it was already noted in [4] that in practical situations we deal frequently with only finite subsets. Thus, in most of the cases, the assumption that the membership degrees are finite and nonempty subsets is considered. Then we are able to work with a particular type of hesitant fuzzy sets, the typical hesitant fuzzy sets, defined as follows:

Definition 2 *[4] Let $\mathcal{H} \subset \mathcal{P}([0,1])$ be the set of all finite nonempty subsets of the interval $[0,1]$ and let X be a nonempty universe. A typical hesitant fuzzy set A over X is given by*

$$A = \{\langle x, h_A(x)\rangle : x \in X\},$$

where $h_A : X \to \mathcal{H}$.

Throughout this work we will deal only with typical hesitant fuzzy sets and, by simplicity, we will call them just hesitant fuzzy sets.

Several authors have studied different operations on hesitant fuzzy sets (see, e.g., [24,33,36]). In particular, a very important concept in this work is the intersection of two hesitant fuzzy sets proposed by Torra ([33]), that extends the classical definition of intersection of two fuzzy sets given by Zadeh ([39]).

Definition 3 *[33] Let A, B be hesitant fuzzy sets on a universe X. The intersection of A and B is a hesitant fuzzy set, denoted by $A \cap B$ defined by:*

$$h_{A\cap B}(x) = \{\gamma \in \{h_A(x) \cup h_B(x)\} \,|\, \gamma \leq \min\{\max\{h_A(x)\}, \max\{h_B(x)\}\}\}$$

for any $x \in X$.

Thus, the membership function of the intersection is obtained as the set of all values in any of the two sets which are lower than or equal to the smaller of the two maximums at a particular point. In order to clarify this definition, we will show an easy example.

Example 1 *If we consider an ordered space* $X = \{x_1, x_2, x_3\}$ *with* $x_1 < x_2 < x_3$ *and the hesitant fuzzy sets* A *and* B *defined by*

X	x_1	x_2	x_3
A	$\{0.2, 0.6\}$	$\{0\}$	$\{0.2, 0.4, 0.6, 0.8\}$
B	$\{0.4\}$	$\{0.6\}$	$\{1\}$

their intersection is defined as

X	x_1	x_2	x_3
$A \cap B$	$\{0.2, 0.4\}$	$\{0\}$	$\{0.2, 0.4, 0.6, 0.8\}$

The graphical representation of A, B *and* $A \cap B$ *is in Figure 1.*

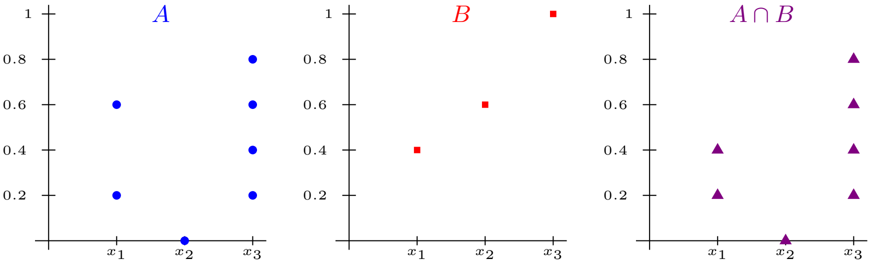


Fig. 1. Example of intersection of hesitant fuzzy sets.

The hesitant fuzzy logic associated to this operation and the corresponding union can be encompassed in the logic systems proposed in [8] by considering the appropriate lattices and set of functions. As a consequence of the concepts introduced in [9], we could consider some different approach to the idea of intersection of two hesitant fuzzy sets, but we have prefered to consider the usual definition of intersection, in order to be able to link our research with the related recent papers in the literature [1,5,14,24,26,27].

Alcantud [1] and Zhu et al. [42] used this definition of intersection on their works in order to define new intersections between dual hesitant fuzzy elements and dual extended hesitant fuzzy elements, respectively. Similarly, Pei and Yi [24] also used this definition to investigate about semilattices of hesitant fuzzy sets. Rodriguez et al. [27] presented an overview of hesitant fuzzy sets to provide a clear perspective on the different concepts, tools, and trends related to this extension of fuzzy sets. On [26], Rashid and Beg provided a definition of convexity based on the convexity of the score function that does not guarantee the preservation of convexity under intersections, but Janis et al. [14] presented a concept of convexity for hesitant fuzzy sets, based on the maximum operator, without this drawback.

Until now we were considering hesitant fuzzy sets in general, but we are particularly interested in convex (typical) hesitant fuzzy sets. The first definition given at the literature was based on the score function, which was introduced by Xia and Xu [36].

Definition 4 *Let A be a hesitant fuzzy set on a finite universe X. The map $s_A : X \to [0,1]$ defined by*

$$s_A(x) = \frac{1}{|h_A(x)|} \sum_{\gamma \in h_A(x)} \gamma$$

is called the score function of A.

It is clear the the score function is just the arithmetic mean of the membership function h_A of A at any point x in X. Based on this concept, Rashid and Beg introduced in 2016 the concept of convexity for hesitant fuzzy sets.

Definition 5 *[26] Let X be a finite linear space. A hesitant fuzzy set A on the universe X is said to be convex, if for all $x, y \in X$, and $\lambda \in [0,1]$ it holds that*

$$s_A(\lambda x + (1-\lambda)y) \geq \min\{s_A(x), s_A(y)\}.$$

The authors in fact call this property quasiconvexity instead of convexity, but we will use the previous name for simplicity. They also considered that for any hesitant fuzzy set, its alpha-cut is defined by:

$$A_\alpha = \{x \in X : s_A(x) \geq \alpha\}$$

for any $\alpha \in (0,1]$ and with this definition they prove the following result.

Proposition 1 *[26] Let X be a finite linear space and let A be a hesitant fuzzy set defined on X. The followings statements are equivalent:*

1. *A is a quasi-convex hesitant fuzzy set.*
2. *Any α-cut of A is convex crisp set.*

Apart from the cutworthy approach, one of the principal properties of convexity for fuzzy sets was its preservation under arbitrary intersections. This is a very important requirement, since it makes the collection of convex sets very important for applications as, for instance, optimization (see [2] or [30]). As it was shown in [14], the concept of convexity introduced by Rashid and Beg does not preserve this property. In that paper, a definition of convexity for hesitant fuzzy sets was given such that it fulfills the natural conditions: it extends the concept of convexity for fuzzy sets, it preserves convexity under intersections and equivalence of the convexity for cuts. Taking into account these properties, they arrived to a natural definition of convexity for hesitant fuzzy sets based on the maximum of the membership values at any point. More precisely,

Definition 6 *[14] Let X be a finite linear space and let A be a hesitant fuzzy set on X. Then A is convex, if*

$$\max\{h_A(\lambda x + (1-\lambda)y)\} \geq \min\{\max\{h_A(x)\}, \max\{h_A(z)\}\}$$

for each $x, z \in X$ and for each $\lambda \in [0,1]$.

Although the definitions 5 and 6 are not so similar at a first sight, they have a common idea behind. We define convexity based on the arithmetic mean or the maximum. Thus, in both cases, we use two particular examples of aggregation functions. Let us recall this concept.

Definition 7 *[21] Let $\mathcal{A} : \cup_{i=1}^{n}[0,1]^i \to [0,1]$ such that*

- $\mathcal{A}(0,0,\ldots,0) = 0, \mathcal{A}(1,1,\ldots,1) = 1$
- $\mathcal{A}(x) = x$ *for all* $x \in [0,1]$
- $\mathcal{A}$ *is monotone at each variable*

then $\mathcal{A}$ is called an aggregation function.

This will be the starting point for our general definition of convexity for hesitant fuzzy sets.

3. General convexity for hesitant fuzzy sets

Once we have defined an aggregation function, we are able to formulate convexity for any hesitant fuzzy set defined on a finite universe. In order to consider the most general possible definition, we will also consider any ordered space as the universe, instead of a linear space. This is not a real generalization, but it could be more appropriate at the environment we use to work.

Definition 8 *Let X be an ordered space, let A be a hesitant fuzzy set on X, let $\mathcal{A}$ be an aggregation function. Then A is $\mathcal{A}$-convex, if for each $x < y < z$ there is*

$$\mathcal{A}(h_A(y)) \geq \min\{\mathcal{A}(h_A(x)), \mathcal{A}(h_A(z))\}.$$

By the second axiom in Definition 7, it is immediate that a convex fuzzy set considered as a hesitant fuzzy set with singleton values is convex.

Another usual requirement is the cutworthy property. Thus, first of all, we will propose a reasonable definition of cut for hesitant fuzzy sets.

Definition 9 *Let X be an ordered space, let A be a hesitant fuzzy set on X, let $\mathcal{A}$ be an aggregation function. The α-cut of A with respect to $\mathcal{A}$ is defined as the crisp set:*

$$A_\alpha^{\mathcal{A}} = \{x \in X : \mathcal{A}(h_A(x)) \geq \alpha\}$$

for any $\alpha \in (0,1]$.

With respect to this definition, convexity of a hesitant fuzzy set is equivalent to convexity of its cuts.

Proposition 2 *Let X be an ordered space, let A be a hesitant fuzzy set on X, let $\mathcal{A}$ be an aggregation function. A is $\mathcal{A}$-convex if and only if $A_\alpha^{\mathcal{A}}$ is convex for any $\alpha \in (0,1]$.*

Proof. Let A be a hesitant fuzzy set and let x, y, z be elements in X with $x < y < z$.

On one hand, let us consider that A is $\mathcal{A}$-convex. If $x, z \in A_\alpha^\mathcal{A}$ and $\alpha \in (0, 1]$, we know that $\mathcal{A}(h_A(x)), \mathcal{A}(h_A(z)) \geq \alpha$. Thus, by the $\mathcal{A}$-convexity of A, we know that $\mathcal{A}(h_A(y)) \geq \min\{\mathcal{A}(h_A(x)), \mathcal{A}(h_A(z))\} \geq \alpha$, that is, $A_\alpha^\mathcal{A}$ is a convex crisp set.

On the other hand, if the α-cuts of A w.r.t. $\mathcal{A}$ are convex for any $\alpha \in (0, 1]$, we can consider in particular the value $\alpha_0 = \min\{\mathcal{A}(h_A(x)), \mathcal{A}(h_A(z))\}$. It is clear that $x, z \in A_{\alpha_0}^\mathcal{A}$. By the convexity of this set, we have that $y \in A_{\alpha_0}^\mathcal{A}$ and, therefore, $\mathcal{A}(h_A(y)) \geq \alpha_0$. By taking into account the definition of α_0, we have that A is an $\mathcal{A}$-convex hesitant fuzzy set. □

Thus, this definition generalizes the idea of convexity for fuzzy sets and it is equivalent to the convexity of its associated cuts. The remaining natural property is the preservation of convexity under intersections. Thus, we are going to study whether the intersection of convex hesitant fuzzy sets is a convex hesitant fuzzy set. As we will see, this property is not fulfilled in general. Therefore our main aim is to characterize those aggregations, that lead to a class of convex hesitant fuzzy sets, for which the convexity of their intersections is preserved. To shorten our formulations, we will use the notion "$\mathcal{A}$ preserves convexity", if the class of convex hesitant fuzzy sets obtained using $\mathcal{A}$ in Definition 8 (we have called them $\mathcal{A}$-convex) is closed under intersections.

A similar question of preserving convexity under aggregation for fuzzy sets has been solved in [13].

We will distinguish several cases for the aggregation function $\mathcal{A}$, namely the following ones:

1. $\mathcal{A}$ is the maximum or the minimum.
2. $\mathcal{A}$ is lower than $\min$ at some point.
3. $\mathcal{A}$ is greater than $\max$ at some point.
4. $\mathcal{A}$ is between $\min$ and $\max$ but it is different from both, maximum and minimum.

3.1. The case of the maximum/minimum

If we choose $\mathcal{A}(a, b) = \max\{a, b\}$ for all $a, b \in [0, 1]$ as the aggregation function, convexity is preserved and the same happens for $\mathcal{A}$ being the minimum.

Proposition 3 *Let X be an ordered space. If $\mathcal{A}(a, b) = \max\{a, b\}$ for all $a, b \in [0, 1]$, then $\mathcal{A}$ preserves convexity.*

Proof. Let A and B be two $\mathcal{A}$-convex hesitant fuzzy sets. Let $x, y, z \in X$ such that $x \leq y \leq z$. For any $t \in X$ we have

$$\begin{aligned}\mathcal{A}(h_{A\cap B}(t)) &= \max\{h_{A\cap B}(t)\} = \min\{\max\{h_A(t)\}, \max\{h_B(t)\}\} \\ &= \min\{\mathcal{A}(h_A(t)), \mathcal{A}(h_B(t))\}\end{aligned}$$

by taking into account the definition of $\mathcal{A}$ and Definition 3. Thus,

$$\mathcal{A}(h_{A\cap B}(y)) = \min\{\mathcal{A}(h_A(y)), \mathcal{A}(h_B(y))\}$$

and by the A-convexity of A and B,

$$\mathcal{A}(h_{A\cap B}(y)) \geq \min\{\min\{\mathcal{A}(h_A(x)), \mathcal{A}(h_A(z))\}, \min\{\mathcal{A}(h_B(x)), \mathcal{A}(h_B(z))\}\} =$$

$$\min\{\min\{\mathcal{A}(h_A(x)), \mathcal{A}(h_B(x))\}, \min\{\mathcal{A}(h_A(z)), \mathcal{A}(h_B(z))\}\} =$$

$$\min\{\mathcal{A}(h_{A\cap B}(z)), \mathcal{A}(h_{A\cap B}(z))\}.$$

Therefore, $A \cap B$ is also $\mathcal{A}$-convex. □

This proof is inspired by a similar result in [14]. However, in that case the definition of convexity was slightly different, and it is here adapted to the new definition.

Proposition 4 *Let X be an ordered space. If $\mathcal{A}(a, b) = \min\{a, b\}$ for all $a, b \in [0, 1]$, then $\mathcal{A}$ preserves convexity.*

Proof. Due to the definition of intersection it is clear that:

$$\min\{h_{A\cap B}(x)\} = \min\{h_A(x), h_B(x)\}, \forall x \in X.$$

As A is min-convex,

$$\min\{h_A(y)\} \geq \min\{\min\{h_A(x)\}, \min\{h_A(z)\}\}$$

and the same happens for B,

$$\min\{h_B(y)\} \geq \min\{\min\{h_B(x)\}, \min\{h_B(z)\}\}.$$

Let us check if $A \cap B$ fulfills our definition:

$$\min\{h_{A\cap B}(y)\} = \min\{h_A(y), h_B(y)\} \geq \min\{\min\{h_A(x)\},$$

$$\min\{h_A(z)\}, \min\{h_B(x)\}, \min\{h_B(z)\} = \min\{\min\{h_{A\cap B}(x)\}, \min\{h_{A\cap B}(z)\}\}$$

and we can see that $A \cap B$ is min-convex. □

3.2. The case under the minimum at some point

The second case is when there exist two points $\alpha_1, \alpha_2 \in [0, 1]$ such that $\mathcal{A}(\alpha_1, \alpha_2) < \min\{\alpha_1, \alpha_2\}$.

Proposition 5 *Let X be an ordered space. If $\mathcal{A}$ is an aggregation function such that there is at least one pair of mutually distinct elements $(\alpha_1, \alpha_2) \in [0, 1]^2$ for which $\mathcal{A}(\alpha_1, \alpha_2) < \min\{\alpha_1, \alpha_2\}$, then $\mathcal{A}$ does not preserve convexity.*

Proof. We are going to find a general counterexample, which can be used for any aggregation function assuming at least one value under the minimum. Let $\mathcal{A}$ be an aggregation function such that there are $\alpha_1, \alpha_2 \in [0, 1]$ with $\mathcal{A}(\alpha_1, \alpha_2) < \min\{\alpha_1, \alpha_2\}$.

Let us consider $X = \{x, y, z\}$ with $x < y < z$. Then we can define two hesitant fuzzy sets on X as:

$$h_A(x) = h_A(z) = \{\alpha_1, \alpha_2\}, h_A(y) = \{\mathcal{A}(\alpha_1, \alpha_2)\},$$

and

$$h_B(x) = h_B(y) = h_B(z) = \min\{\alpha_1, \alpha_2\}.$$

It is easy to check that both A and B are $\mathcal{A}$-convex. Their intersection is the hesitant fuzzy set defined by

$$h_{A\cap B}(x) = h_{A\cap B}(z) = \min\{\alpha_1, \alpha_2\}, h_{A\cap B}(y) = \{\mathcal{A}(\alpha_1, \alpha_2)\}.$$

Then

$$\mathcal{A}(h_{A\cap B}(y)) = \mathcal{A}(\alpha_1, \alpha_2) < \min\{\mathcal{A}(h_{A\cap B}(x)), \mathcal{A}(h_{A\cap B}(z))\} =$$

$$\min\{\min\{\alpha_1, \alpha_2\}, \min\{\alpha_1, \alpha_2\}\} = \min\{\alpha_1, \alpha_2\}.$$

So, the intersection is not $\mathcal{A}$-convex. □

This proof is illustrated in Figure 2, where we suppose that $\alpha_1 < \alpha_2$.

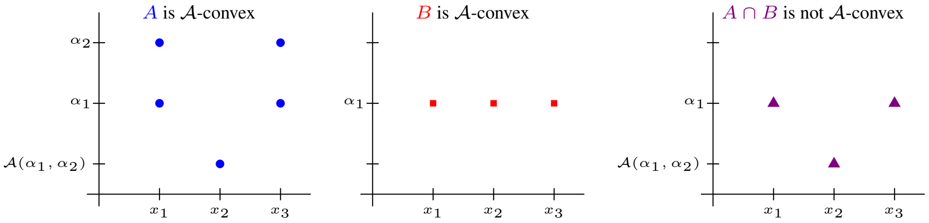


Fig. 2. Graphical proof of Proposition 5.

It is known that triangular norms (t-norms for short) are aggregation functions that fulfill $T(x, y) \leq \min\{x, y\}$ for all $(x, y) \in [0, 1]^2$. Thus, any t-norm different from the minimum fulfills that there exists a point $(\alpha_1, \alpha_2) \in [0, 1]^2$ such that $T(\alpha_1, \alpha_2) < \min\{\alpha_1, \alpha_2\}$. Therefore, by applying Proposition 5, any t-norm different from the minimum is not an appropriate choice for defining convexity, if we would expect convexity to be preserved under intersections.

3.3. The case over the maximum at some point

In the third case, we suppose that there exist points $\alpha_1, \alpha_2 \in [0, 1]$ such that the aggregation function fulfills $\mathcal{A}(\alpha_1, \alpha_2) > \max\{\alpha_1, \alpha_2\}$. The behavior with respect to the preservation of convexity under intersections and the way to prove is analogous to the previous case.

Proposition 6 *Let X be an ordered space. If $\mathcal{A}$ is an aggregation function such that there is at least one pair of mutually distinct elements $(\alpha_1, \alpha_2) \in [0, 1]^2$ for which $\mathcal{A}(\alpha_1, \alpha_2) > \max\{\alpha_1, \alpha_2\}$, then $\mathcal{A}$ does not preserve convexity.*

Proof. Now we are going to find a general counterexample, which can be used for any aggregation function under the conditions of the statement.

Let $X = \{x, y, z\}$ be the universe with $x < y < z$ and let A and B the hesitant fuzzy sets whose membership functions are:

$$h_A(x) = h_A(z) = \{\alpha_1, \alpha_2\}, h_A(y) = \{\mathcal{A}(\alpha_1, \alpha_2)\},$$

and

$$h_B(x) = h_B(y) = h_B(z) = \max\{\alpha_1, \alpha_2\}.$$

It is easy to check that both A and B are $\mathcal{A}$-convex. Their intersection is the hesitant fuzzy set given by

$$h_{A\cap B}(x) = h_{A\cap B}(z) = \{\alpha_1, \alpha_2\}, h_{A\cap B}(y) = \max\{\alpha_1, \alpha_2\}.$$

Then

$$\mathcal{A}(h_{A\cap B}(y)) = \mathcal{A}(\max\{\alpha_1, \alpha_2\}) = \max\{\alpha_1, \alpha_2\} <$$
$$\min\{\mathcal{A}(h_{A\cap B}(x)), \mathcal{A}(h_{A\cap B}(z))\} = \min\{\mathcal{A}(\alpha_1, \alpha_2), \mathcal{A}(\alpha_1, \alpha_2)\} =$$
$$\mathcal{A}(\alpha_1, \alpha_2) > \max\{h_{A\cap B}(x), h_{A\cap B}(z)\}.$$

So, the intersection is not $\mathcal{A}$-convex. □

This proof is again graphically illustrated. In this case in Figure 3, where we suppose that $\alpha_1 < \alpha_2$.

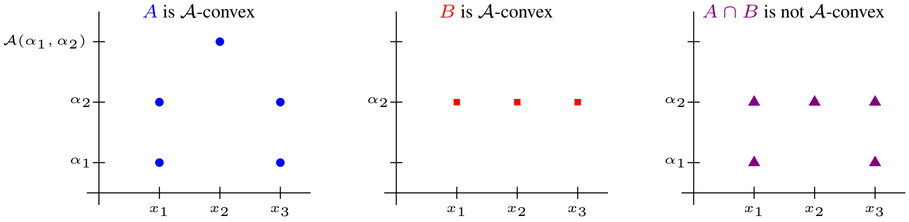


Fig. 3. Graphical proof of Proposition 6.

It is known that triangular conorms are aggregation functions that fulfill $S(x, y) \geq \max\{x, y\}$ for all $(x, y) \in [0, 1]^2$. Thus, any triangular conorm different from the maximum does not preserve convexity.

3.4. The case between minimum and maximum

Now we will study the only remaining case, aggregation functions which are between the minimum and the maximum, but they are not equal to any of them. In this case, we cannot describe the general behavior of this aggregation functions, since some of them preserve convexity for the intersection and some of them do not preserve it.

As we commented previously, the arithmetic mean is an aggregation function between minimum and maximum such that the intersection of two convex hesitant fuzzy sets need not be convex. This was already commented in [14] for their definition, but it is also true now for the new one, as we can see from the following example.

Example 2 *Let $X = \{x, y, z\}$ be the universe with $x < y < z$ and let A and B the hesitant fuzzy sets defined by:*

X	x	y	z
A	$\{0.4\}$	$\{0.2, 0.6\}$	$\{0.4\}$
B	$\{0.4\}$	$\{0.4\}$	$\{0.4\}$

It is clear that if $\mathcal{A}$ is the arithmetic mean, A and B are $\mathcal{A}$-convex. However, their intersection is defined as

X	x	y	z
$A \cap B$	$\{0.4\}$	$\{0.2, 0.4\}$	$\{0.4\}$

but $\mathcal{A}(y) = 0.3 < \min\{\mathcal{A}(x), \mathcal{A}(z)\} = 0.4$. Thus, $A \cap B$ is not $\mathcal{A}$-convex.

However, it is not true that any aggregation function in this family does not preserve convexity. We will introduce an example of an aggregation function in this family preserving convexity. First of all, we are going to consider a map on $[0,1]^2$.

This mapping is a function $\mathcal{A}_2 : [0,1]^2 \to [0,1]$ defined as follows

$$\mathcal{A}_2(x,y) = \begin{cases} \max\{x,y\} & \text{if } x,y \in [0,0.5] \\ \min\{x,y\} & \text{if } x,y \in (0.5,1] \\ 0.5 & \text{otherwise.} \end{cases}$$

Its graphical representation is given at Figure 4.

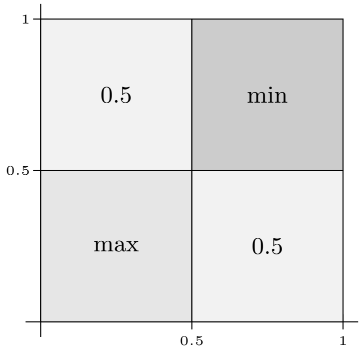


Fig. 4. Graphical representation of $\mathcal{A}_2$.

This mapping is a nullnorm (see [6]) and it is known that nullnorms are associative. So, it can be in a natural way extended to a mapping $\mathcal{A}_n : \cup_n [0,1]^n \to [0,1]$, which is also an aggregation function.

By definition it is also trivial that $\min\{\mathbf{x}\} \leq \mathcal{A}_n(\mathbf{x}) \leq \max\{\mathbf{x}\}$ for any $\mathbf{x} \in \cup_n [0,1]^n$. Moreover, $\mathcal{A}_2(0.7, 0.2) = 0.5$, so it is also clear that $\mathcal{A}_n$ is not equal in general to the minimum or the maximum.

Finally, we will see that $\mathcal{A}_n$-convexity is preserved by intersections.

Proposition 7 *The intersection of any two $\mathcal{A}_n$-convex hesitant fuzzy sets is a $\mathcal{A}_n$-convex hesitant fuzzy set.*

Proof. Let us suppose that X is an ordered space. Let A and B be $\mathcal{A}_n$-convex hesitant fuzzy sets. Let $x, y, z \in X$ such that $x < y < z$.

We will divide the proof into three cases:

1. The case $\mathcal{A}_n(h_{A\cap B}(y)) > 0.5$.
   (a) If $\mathcal{A}_n(h_{A\cap B}(x)) \leq 0.5$ or $\mathcal{A}_n(h_{A\cap B}(z)) \leq 0.5$ then

   $$\min\{\mathcal{A}_n(h_{A\cap B}(x)), \mathcal{A}_n(h_{A\cap B}(z))\} \leq 0.5 < \mathcal{A}_n(h_{A\cap B}(y))$$

   and therefore the condition to be $A \cap B$ $\mathcal{A}_n$-convex is fulfilled in this case.

   (b) If $\mathcal{A}_n(h_{A\cap B}(x)) > 0.5$ and $\mathcal{A}_n(h_{A\cap B}(z)) > 0.5$ then, by the definition of $\mathcal{A}_n$, $\mathcal{A}_n(h_{A\cap B}(x)) = \min\{h_{A\cap B}(x)\}$ and $\mathcal{A}_n(h_{A\cap B}(z)) = \min\{h_{A\cap B}(z)\}$ and considering the definition of the intersection (Definition 3), we have that $0.5 < \mathcal{A}_n(h_{A\cap B}(x)) = \min\{h_{A\cap B}(x)\} = \min\{h_A(x), h_B(x)\}$ and that $0.5 < \mathcal{A}_n(h_{A\cap B}(z)) = \min\{h_{A\cap B}(z)\} = \min\{h_A(z), h_B(z)\}$.
   Then,

   $$\min\{\mathcal{A}_n(h_{A\cap B}(x)), \mathcal{A}_n(h_{A\cap B}(z))\} = \min\{h_A(x), h_B(x), h_A(z), h_B(z)\} =$$

   $$\min\{\min\{\min\{h_A(x)\}, \min\{h_A(z)\}\}, \min\{\min\{h_B(x)\}, \min\{h_B(z)\}\}\}.$$

   As we noticed that $0.5 < \min\{h_A(x), h_B(x)\}$ and $0.5 < \min\{h_A(z), h_B(z)\}$, by definition of $\mathcal{A}_n$, we have that $\mathcal{A}_n(h_A(x)) = \min\{h_A(x)\}$, $\mathcal{A}_n(h_A(z)) = \min\{h_A(z)\}$, $\mathcal{A}_n(h_B(x)) = \min\{h_B(x)\}$ and $\mathcal{A}_n(h_B(z)) = \min\{h_B(z)\}$.
   Then,

   $$\min\{\mathcal{A}_n(h_{A\cap B}(x)), \mathcal{A}_n(h_{A\cap B}(z))\} =$$

   $$\min\{\min\{\mathcal{A}_n(h_A(x)), \mathcal{A}_n(h_A(z))\}, \min\{\mathcal{A}_n(h_B(x)), \mathcal{A}_n(h_B(z))\}\}.$$

   But, by the $\mathcal{A}_n$-convexity of A and B we have that

   $$\min\{\mathcal{A}_n(h_{A\cap B}(x)), \mathcal{A}_n(h_{A\cap B}(z))\} \leq \min\{\mathcal{A}_n(h_A(y)), \mathcal{A}_n(h_B(y))\}$$

   On the other hand, $\mathcal{A}_n(h_{A\cap B}(y)) > 0.5$, we also have that $\mathcal{A}_n(h_{A\cap B}(y)) = \min\{h_{A\cap B}(y)\} = \min\{h_A(y), h_B(y)\} = \min\{\min\{h_A(y)\}, \min\{h_B(y)\}\} = \min\{\mathcal{A}_n(h_A(y)), \mathcal{A}_n(h_B(y))\}$.
   Thus, we have proven that

   $$\min\{\mathcal{A}_n(h_{A\cap B}(x)), \mathcal{A}_n(h_{A\cap B}(z))\} \leq \mathcal{A}_n(h_{A\cap B}(y)).$$

2. If $\mathcal{A}_n(h_{A\cap B}(y)) < 0.5$, then $\mathcal{A}_n(h_{A\cap B}(y)) = \max\{h_{A\cap B}(y)\}$.
   By the definition of the intersection, we have $\max\{h_{A\cap B}(y)\} = \max\{h_A(y)\}$ or $\max\{h_{A\cap B}(y)\} = \max\{h_B(y)\}$. Suppose we have the first case (the proof for the second case it totally analogous). Since $\max\{h_A(y)\} < 0.5$, then $\mathcal{A}_n(h_A(y)) = \max\{h_A(y)\}$. By applying that A is a $\mathcal{A}_n$-convex hesitant fuzzy set, we have $\mathcal{A}_n(h_A(x)) \leq \mathcal{A}_n(h_A(y))$ or $\mathcal{A}_n(h_A(z)) \leq \mathcal{A}_n(h_A(y))$. Let us consider that we have the first case (again the second case is analogous). Thus, $\mathcal{A}_n(h_A(x)) < 0.5$ and then $\mathcal{A}_n(h_A(x)) = \max\{h_A(x)\} < 0.5$. By considering again the definition of the intersection, we see that $\max\{h_{A\cap B}(x)\} \leq \max\{h_A(x)\} < 0.5$ and therefore $\mathcal{A}_n(h_{A\cap B}(x)) = \max\{h_{A\cap B}(x)\}$. Now, if we join the above inequalities and equalities, we have:

   $$\mathcal{A}_n(h_{A\cap B}(x)) = \max\{h_{A\cap B}(x)\} \leq \max\{h_A(x)\} = \mathcal{A}_n(h_A(x)) \leq$$

   $$\mathcal{A}_n(h_A(y)) = \max\{h_{A\cap B}(y)\} = \mathcal{A}_n(h_{A\cap B}(y))$$

   and then

   $$\mathcal{A}_n(h_{A\cap B}(y)) \geq \min\{\mathcal{A}_n(h_{A\cap B}(x)), \mathcal{A}_n(h_{A\cap B}(z))\}.$$

3. The case $\mathcal{A}_n(h_{A\cap B}(y)) = 0.5$.
   If $\mathcal{A}_n(h_{A\cap B}(x)) \leq 0.5$ or $\mathcal{A}_n(h_{A\cap B}(z)) \leq 0.5$, then the proof is trivial. Thus, we will consider that $\mathcal{A}_n(h_{A\cap B}(x)) > 0.5$ and $\mathcal{A}_n(h_{A\cap B}(z)) > 0.5$. In that case, $\mathcal{A}_n(h_{A\cap B}(x)) = \min\{h_{A\cap B}(x)\} = \min\{h_A(x)\cup h_B(x)\} > 0.5$ and $\mathcal{A}_n(h_{A\cap B}(z)) = \min\{h_{A\cap B}(z)\} = \min\{h_A(z) \cup h_B(z)\} > 0.5$. Then

   $$\min\{h_A(x)\}, \min\{h_A(z)\}, \min\{h_B(x)\}, \min\{h_B(z)\} > 0.5$$

   and therefore $\mathcal{A}_n(h_A(x)) = \min\{h_A(x)\} > 0.5$ and similarly we prove that

   $$\mathcal{A}_n(h_A(z)), \mathcal{A}_n(h_B(x)), \mathcal{A}_n(h_B(z)) > 0.5.$$

   As A and B are $\mathcal{A}_n$-convex, then $\mathcal{A}_n(h_A(y)) > 0.5$ and $\mathcal{A}_n(h_B(y)) > 0.5$. Then, $\min\{h_A(y) \cup h_B(y)\} > 0.5$ and therefore $\mathcal{A}_n(h_{A\cap B}(y)) = \min\{h_{A\cap B}(y)\} > 0.5$ which is a contradiction, so we can assure that $\mathcal{A}_n(h_{A\cap B}(x)) \leq 0.5$ or $\mathcal{A}_n(h_{A\cap B}(z)) \leq 0.5$ and therefore

   $$\mathcal{A}_n(h_{A\cap B}(y)) = 0.5 \geq \min\{\mathcal{A}_n(h_{A\cap B}(x)), \mathcal{A}_n(h_{A\cap B}(z))\}.$$

Thus, we have proven that $A \cap B$ is $\mathcal{A}_n$-convex. □

Note that in fact any nullnorm and its extensions could be used in the previous demonstration.

As all the four cases are studied, we know the behavior of the different aggregation functions with respect to the preservation of convexity under intersections. Now we can say that only minimum, maximum and some specific aggregation functions between them are appropriate to define convexity for hesitant fuzzy sets.

Nowadays, there are many experiences where more than one attribute is to be combined into one overall value. In lots of cases, the attributes compensate for each other, to some extent. For teachers, there is a common situation that there are students with high motivation which compensate for less intelligence or vice versa [44]. Compensatory operators were introduced by Zimmermann and Zysno [43]. The main idea of these operators is to provide compensation between the small and large degrees of membership when combining fuzzy sets. Since then, several studies of these operators were presented [16,22,23]. For instance, Kolesarova and Komornikova gave the following definition in 1999:

Definition 10 *[17] An aggregation operator $\mathscr{A}$ on the unit interval is called a compensatory operator if for each n-tuple $(x_1, \ldots, x_n) \in [0,1]^n$, $n \in \mathbb{N}$, such that $\mathscr{A}(x_1, \ldots, x_n) \in]0,1[$ there exist elements $y, z \in [0,1]$ such that $\mathscr{A}(x_1, \ldots, x_n, y) < \mathscr{A}(x_1, \ldots, x_n) < \mathscr{A}(x_1, \ldots, x_n, z)$*

We can see with these comments that the idea of compensation is just to obtain low values when the argues are small and vice versa. For instance, the arithmetic mean is a compensatory operator which does not preserve convexity.

4. Illustrative example: optimization

As we mentioned in the introduction, the notion of convexity has been applied in many different contexts. One of the most interesting is in the field of optimization. A simple example of a possible application is shown here. We can use it to see the possible applications of the developed results.

Let us consider we have to design two new types of paint, which are prepared by mixing three primary elements: r, b and y. To obtain one bottle of the paint type A we need 3 units of r and 2 units of b. To obtain one bottle of the paint type B we need 5 units of r and 1 unit of y. The availability of r, b and y are 15 units, 6 units and 2 units, respectively. If at least one bottle has to be prepared, then the set of possible solutions about the number of bottles prepared for A and B is represented by dots in Figure 5.

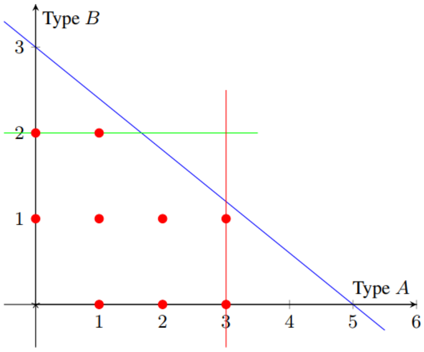


Fig. 5. Feasible region.

Depending on the number of bottles type A and B we are considering, we could obtain by different procedures, 9 different products, which will be represented by A_iB_j with $i = 0, 1, 2, 3$ and $j = 0, 1, 2$. If a committee of two experts evaluates the functionality and design of these products, we obtain two different hesitant fuzzy sets whose referential is the feasible region. Thus, let us suppose the values assigned to the functionality (F) and the design (D) for the different products are:

X	A_0B_1	A_0B_2	A_1B_0	A_1B_1	A_1B_2	A_2B_0	A_2B_1	A_3B_0	A_3B_1
F	$\{0.2, 0.6\}$	$\{0.2, 0.3\}$	$\{0.4, 0.6\}$	$\{0.5, 0.6\}$	$\{0.7, 0.6\}$	$\{0.6, 0.6\}$	$\{0.4, 0.5\}$	$\{0.2, 0.3\}$	$\{0.5, 0.1\}$
D	$\{0.4, 0.6\}$	$\{0.6, 0.5\}$	$\{0.6, 0.7\}$	$\{0.8, 0.7\}$	$\{0.9, 1\}$	$\{1, 1\}$	$\{0.9, 0.7\}$	$\{0.6, 0.7\}$	$\{0.5, 0.4\}$

If we consider the lexicographical order in $\mathbb{R}^2$, that is, $(x_1, y_1) \leq (x_2, y_2)$ iff $x_1 < x_1$ or $x_1 = x_2$ and $y_1 \leq y_2$, it is clear that F and D are min-convex hesitant fuzzy sets. Thus, the products fulfilling both properties could be represented as

X	A_0B_1	A_0B_2	A_1B_0	A_1B_1	A_1B_2	A_2B_0	A_2B_1	A_3B_0	A_3B_1
$F \cap D$	$\{0.2, 0.4, 0.6\}$	$\{0.2, 0.3\}$	$\{0.4, 0.6\}$	$\{0.5, 0.6\}$	$\{0.7, 0.6\}$	$\{0.6, 0.6\}$	$\{0.4, 0.5\}$	$\{0.2, 0.3\}$	$\{0.5, 0.1, 0.4\}$

By Proposition 4, the hesitant fuzzy set above, which represents the membership values to any product about both properties, is again a min-convex hesitant fuzzy sets. By

Proposition 2, any of its α-cut is a convex set. Thus, for a fixed level, we could optimize any objective function on this set by using the classical theorem about optimization. Of course, only in the case the aggregation function preserves the convexity under intersections, we could work in a similar way.

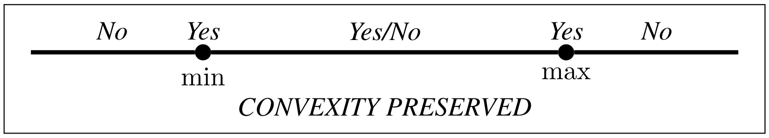
This easy example can be seen as a way to show the different possible applications of the obtained results for typical hesitant fuzzy sets. However, for a real application, some other structures could be considered in two senses. On one hand, the case the membership value is a multiset, to work in a different way if a value is considered just by one expert or by one hundred. On the other hand, the case of interval instead of discrete sets as membership values, i.e. interval-valued fuzzy sets, could be also necessary in some cases. However, typical hesitant fuzzy sets have been revealed as a very useful tool, as we commented on the introduction.

5. Conclusions

In this paper, we have introduced a new definition of convexity for typical hesitant fuzzy sets, which are finite hesitant fuzzy sets, based on aggregation functions. This definition can be considered for any ordered universe. Moreover, it coincides with the classical definition for fuzzy sets. It also has a good behavior with respect to the equivalence for cuts. About the preservation of the convexity under intersections, we have studied the behavior for various ways of aggregating values of a hesitant fuzzy set. Our results may be summarized as follows:

- Convexity is preserved for the minimum and the maximum.
- Convexity is not preserved if the aggregation function is under the minimum, respectively over the maximum, at some point.
- For the case between the minimum and the maximum, there are cases when convexity is preserved as well as those when it is not.

These results can also be visualized as follows:



The question of the exact characterization of the aggregation functions between minimum and maximum that preserve convexity remains to be open. So does the study of the preservation of the convexity for some possible general concepts of the intersection. This study can be seen as the first step in a general study. Thus, once we finish the theoretical part, we would like to combine it with some other extension of fuzzy sets, in order to apply it for some real problems about optimization under uncertainty.

Acknowledgements. Authors would like to thank for the support of the grant no. 1/0150/21 provided by Slovak grant agency VEGA (V. Janiš), Spanish Ministry of Science and Technology project TIN-2017-87600-P (P. Alonso), Spanish Ministry of Science and Technology project PGC2018-098623-B-I00 (P. Huidobro and S. Montes) and FICYT Project IDI/2018/000176 (P. Alonso, P. Huidobro and S. Montes).

References

1. Alcantud, J.C., Santos-García, G., Peng, X., Zhan, J.: Dual extended hesitant fuzzy sets. Symmetry 11, 714 (2019)
2. Ammar, E., Metz, J.: On fuzzy convexity and parametric fuzzy optimization. Fuzzy Sets and Systems 49(2), 135 – 141 (1992)
3. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets and Systems 20(1), 87 – 96 (1986)
4. Bedregal, B., Santiago, R., Bustince, H., Parternain, D., Reiser, R.: Typical hesitant fuzzy negations. International Journal of Intelligent Systems 29, 525 – 543 (2014)
5. Bustince, H., Barrenechea, E., Pagola, M., Fernández, J., Xu, Z., Bedregal, B., Montero, J., Hagras, H., Herrera, F., De Baets, B.: A historical account of types of fuzzy sets and their relationships. IEEE Transactions on Fuzzy Systems 24(1), 179–194 (2016)
6. Calvo, T., De Baets, B., Fodor, J.: The functional equations of Frank and Alsina for uninorms and nullnorms. Fuzzy Sets and Systems 120, 385 – 394 (2001)
7. Chen, N., Xu, Z.: Properties of interval-valued hesitant fuzzy sets. Journal of Intelligent and Fuzzy Systems 27, 143–158 (01 2014)
8. De Miguel, L.: Computing with uncertainty truth degrees: a convolution-based degrees. PhD thesis, Public University of Navarra, Spain (2017)
9. De Miguel, L., Bustince, H., De Baets, B.: Convolution lattices. Fuzzy Sets and Systems 335, 67–93 (2018)
10. Díaz, S., Induráin, E., Janiš, V., Llinares, J.V., Montes, S.: Generalized convexities related to aggregation operators of fuzzy sets. Kybernetika 53, 383 – 393 (2017)
11. Grattan-Guinness, I.: Fuzzy membership mapped onto intervals and many-valued quantities. Mathematical Logic Quarterly 22(1), 149–160 (1976)
12. Jahn, K.: Intervall-wertige mengen. Math.Nach. 68, 115–132 (1975)
13. Janiš, V., Král', P., Renčová, M.: Aggregation operators preserving quasiconvexity. Information Sciences 228, 37–44 (2013)
14. Janiš, V., Montes, S., Renčová, M.: Convexity of hesitant fuzzy sets. Journal of Intelligent & Fuzzy Systems 34, 2099–2012 (2018)
15. Kim, Y., Xi, Z., Lien, J.: Disjoint convex shell and its applications in mesh unfolding. Computer-Aided Design 90, 180–190 (2017)
16. Klement, E., Mesiar, R., Pap, E.: On the relationship of associative compensatory operators to triangular norms and conorms. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 04(02), 129–144 (1996)
17. Kolesárová, A., Komorníková, M.: Triangular norm-based iterative compensatory operators. Fuzzy Sets and Systems 104(1), 109 – 120 (1999), triangular Norms
18. Lay, S.: Convex sets and their applications. Dover Publications Inc. (2007)
19. Li, M., Yang, W., Zhang, X.: Projection on convex set and its application in testing force closure properties of robotic grasping. vol. 6425, pp. 240–251 (11 2010)
20. Liberti, L.: Reformulation and convex relaxation techniques for global optimization. 4OR quarterly journal of the Belgian, French and Italian Operations Research Societies 2(3), 255–258 (2004)
21. Mesiar, R., Komorníková, M.: Aggregation operators. In: Proc. XI Conference on applied Mathematics PRIM' 96. D. Herceg and K. Surla, eds., Novi Sad, Serbia (1996)
22. Mizumoto, M.: Pictorial representations of fuzzy connectives, part ii: Cases of compensatory operators and self-dual operators. Fuzzy Sets and Systems 32(1), 45 – 79 (1989)
23. Moser, B., Tsiporkova, E., Klement, E.: Convex combinations in terms of triangular norms: A characterization of idempotent, bisymmetrical and self-dual compensatory operators. Fuzzy Sets and Systems 104(1), 97 – 108 (1999), triangular Norms
24. Pei, Z., Yi, L.: A note on operations of hesitant fuzzy sets. International Journal of Computational Intelligence Systems 8(2), 226–239 (2015)

25. Ramik, J., Vlach, M.: Generalized Concavity in Fuzzy Optimization and Decision Analysis. Kluwer Academic Publishers (2002)
26. Rashid, T., Beg, I.: Convex hesitant fuzzy sets. Journal of Intelligent and Fuzzy Systems 30(5), 2791–2796 (2016)
27. Rodríguez, R., Martínez, L., Torra, V., Xu, Z., Herrera, F.: Hesitant fuzzy sets: State of the art and future directions. International Journal of Intelligent Systems 29(6), 495–524 (2014)
28. Sambuc, R.: Function phi-flous, application a l'aide au diagnostic en pathologie thyroidienne. These de Doctorat en Medicine (1975)
29. Sarkar, D.: Concavoconvex fuzzy set. Fuzzy Sets and Systems 79, 267–269 (1996)
30. Syau, Y., Lee, E.: Fuzzy convexity with application to fuzzy decision making. In: Proceedings of the 42nd IEEE Conference on Decision and Control. p. 5221–5226 (2003)
31. Syau, Y., Lee, E.: Fuzzy convexity and multiobjective convex optimization problems. Computers & Mathematics with Applications 52(3), 351 – 362 (2006)
32. Tofighi, M., Yorulmaz, O., Kose, K., Kahraman, D., Cetin-Atalay, R., Cetin, A.: Phase and tv based convex sets for blind deconvolution of microscopic images. IEEE Journal of Selected Topics in Signal Processing 10(1), 81–91 (2015)
33. Torra, V.: Hesitant fuzzy sets. International Journal of Intelligent Systems 25(6), 529–539 (2010)
34. Wei, G., Wang, H., Zhao, X., Lin, R.: Approaches to hes- itant fuzzy multiple attribute decision making with incomplete weight information. Journal of Intelligent and Fuzzy Systems 26(1), 259–266 (2014)
35. Wei, G., Zhang, N.: A multiple criteria hesitant fuzzy deci- sion making with shapley value-based vikor method. Journal of Intelligent and Fuzzy Systems 26(2), 1065–1075 (2014)
36. Xia, M., Xu, Z.: Hesitant fuzzy information aggregation in decision making. International Journal of Approximate Reasoning 52(3), 395 – 407 (2011)
37. Xu, Z., Xia, M., Chen, N.: Some hesitant fuzzy aggregation operators with their application in group decision making. Group Decision and Negotiation 22(2), 259–279 (2013)
38. Yang, X.: A property on convex fuzzy sets. Fuzzy Sets and Systems 126, 269–271 (2002)
39. Zadeh, L.: Fuzzy sets. Information and Control 8(3), 338 – 353 (1965)
40. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning–I. Information Sciences 8(3), 199 – 249 (1975)
41. Zhang, Z.: Hesitant fuzzy power aggregation operators and their application to multiple attribute group decision making. Information Sciences 234, 150–181 (2013)
42. Zhu, B., Xu, Z., Xia, M.: Dual hesitant fuzzy sets. Journal of Applied Mathematics 2012 (05 2012)
43. Zimmermann, H., Zysno, P.: Latent connectives in human decision making. Fuzzy Sets and Systems 4(1), 37 – 51 (1980)
44. Zysno, P.: One class of operators for the aggregation of fuzzy sets. In: 8th meeting of the EURO III Congress. Amsterdam (05 1979)

Pedro Huidobro obtained a Mathematics degree from the University of Oviedo (Spain). He achieved a Master Degree of Education in Secondary Education, Vocational Training and Language Teaching from the University of León (Spain). He has cursed also some subjects a Master's Degree in Mathematical Modeling and Research, Statistics and Computing from the University of Oviedo. Nowadays he is a Ph.D. student at the University of Oviedo (Spain) and Matej Bel University (Slovakia) with a scholar grant "Severo Ochoa" given by the Asturias government.

Pedro Alonso received the M.Sc degree in Mathematical Sciences, from the University of Zaragoza (Spain) and the Ph.D. degree in Mathematical Sciences (Cum Laude) at the

University of Oviedo (Spain). Currently, he is a full professor in Applied Mathematics at the University of Oviedo, where he is the member of the research group UNIMODE. His research areas are framed in the study of algorithms (complexity, performance, stability, convergence, error, etc.), as well as in different aspects of fuzzy logic.

Vladimír Janiš graduated at Comenius University in Bratislava, Slovakia in Mathematical Analysis. He obtained the Ph.D. degree in Mathematical Analysis at Slovak Technical University in Bratislava. Currently he is a full professor in Applied Mathematics at Matej Bel University in Banská Bystrica, Slovakia. Works in the areas of metric and topologic properties of fuzzy sets and fuzzy logic.

Susana Montes received the M.Sc degree in Mathematics, option Statistics and Operational Research, from the University of Valladolid (Spain) and the Ph.D. (Cum Laude) degree from the University of Oviedo (Spain). She is a full professor in Statistics and Operational Research at the University of Oviedo, where she is the leader of the research group UNIMODE. She is currently the Secretary of EUSFLAT and Vice-President of IFSA.

Received: April 28, 2020; Accepted: November 5, 2020.

Computer Science and Information Systems 18(1):231–250 https://doi.org/10.2298/CSIS200424036S

Spoken Notifications in Smart Environments Using Croatian Language

Renato Šoić, Marin Vuković, Gordan Ježić

Faculty of Electrical Engineering and Computing,
10000 Zagreb, Croatia
{renato.soic, marin.vukovic, gordan.jezic}@fer.hr

Abstract. Speech technologies have advanced significantly in the last decade, mostly due to rise in available computing power combined with novel approaches to natural language processing. As a result, speech-enabled systems have become popular commercial products, successfully integrated with various environments. However, this can be stated for English and a few other "big" languages. From the perspective of a minority language, such as Croatian, there are many challenges ahead to achieve comparable results. In this paper, we propose a model for natural language generation and speech synthesis in a smart environment using Croatian language. The model is evaluated on 27 users to estimate the quality of user experience. The evaluation goal was to determine what users perceive to be more important – generated speech quality or grammatical correctness of the spoken content. It is shown that most users perceived grammatically correct spoken texts as being of the highest quality.

Keywords: natural language processing, smart environment, speech synthesis, natural language generation.

1. Introduction

Speech enabled computer systems have become a common presence in the last several years. Their domain has extended from personal devices and specialized services to all types of smart human-inhabited environments. As a result, there are many examples of employing speech interfaces in smart home systems, industrial facilities, smart vehicles, public services, etc. Regardless of the operational domain, spoken communication between human users and computer systems needs to address specific features related to the wider context of the target environment [1]. This includes events and conditions in the environment, current and future activities related to users, and the linguistic context when interaction is in process.

In this paper, we describe a model for natural language generation and speech synthesis in a distributed environment. The goal is to notify human users about conditions, changes, and events in the environment with naturally sounding spoken notifications. In order to do so, we integrate research efforts from two research domains – speech technologies and Internet of Things (IoT). The primary motivation is to provide support for Croatian language in a real smart IoT environment. Unfortunately, Croatian language is technologically seriously underdeveloped, with sparse resources available [2]. There are available services related to speech technologies in Croatian language provided by Google, Microsoft, Amazon, and Apple. However, these solutions are commercial products and therefore not available for research purposes. Furthermore, the performance of these systems is often not of desirable quality for a native Croatian speaker. In comparison, in case of English language, there are numerous solutions available, providing experience at a very high level of quality. Therefore, this paper proposes the process and resources required to design a functional system. The proposed system can operate in complex IoT environments consisting of numerous sensors and devices, while interacting with human users using a text-to-speech subsystem. The presented concept can be employed in different scenarios from various domains, such as smart homes, Industry 4.0, public transportation systems, etc.

Development of a system providing speech recognition and speech synthesis capabilities requires a vast amount of work and large volumes of data. From the perspective of an under-resourced minority language such as Croatian, the required efforts are even greater. The first challenge is related to the required resources. For speech synthesis, a large audio corpus from a limited number of different speakers is essential. The recorded speech should be of high quality, and it is desirable that recorded speakers are professionals [3]. In case of speech recognition, a large audio corpus from many different speakers is recommended, with variable sound quality, so the system can learn to handle interferences in a real-world scenario. In addition to speech corpora, flawless corresponding textual transcripts are also required. There are a few Croatian corpora available for research purposes, but their volumes and quality are not sufficient for employing data-driven methods. They are extracted from specific domains (e.g., weather forecast, radio shows, etc.) [4] or contain spontaneous speech in different dialects [5]. Therefore, these corpora are not convenient for the intended task as they do not adequately cover the required variety of linguistic features. In comparison, there are open corpora for English [6][7] and projects providing support for many other languages [8].

The second challenge is related to language modeling and cognitive processes. Unlike English, Croatian language is morphologically very rich [9]. This introduces many challenges related to speech recognition, but also cognitive analysis and synthesis, i.e. natural language understanding and generation. In case of natural language understanding, it is very important to recognize user intent correctly, as it results with an action being performed by the system. Similarly, when constructing a notification for the human user, the message content must be clear and precise.

In case of a distributed smart environment, a system which enables human-computer interaction needs to process all the events from the environment in real-time and provide feedback when necessary. In this case, understanding both user’s and environmental context is essential. User’s context is constructed based on their presence, physical and virtual (online) activities, requests, etc. Environmental context is related to information from the current state of the given environment, such as temperature, humidity, light, noise, etc. Additionally, it should also include future predictions, such as weather forecast, public events and happenings as well as personal planned events.

With all the described problems and challenges considered, the conclusion is that building such a complex system requires very careful system decomposition. Therefore, we adopted a modular approach in which each system module represents one major building block while relying on a specific set of services to accomplish its tasks. The proposed system consists of speech synthesis and a natural language generation subsystem and is organized as an orchestration of those subsystems and their related services required for contextual text to speech synthesis. To achieve high-quality context-based speech synthesis, all the proposed subsystems and related services need to be available and perform their functions. However, since the defined subsystems might be distributed across the IoT network with sometimes limited resources and potential connectivity issues, it is possible that one or more of the subsystems is unable to provide the required processing in real time. In this paper we present the ideal conditions with high-quality synthesized speech, but also examine the cases when one or more subsystems fails to perform the task. Finally, we compare the user perception of quality of synthesized speech recordings when one or more subsystems failed to provide the results to see whether it is possible to still have understandable speech in limited connectivity / processing environment, such as large and complex IoT systems.

The evaluation was performed as a survey with 27 participants who were presented with a set of generated spoken notifications. The presented notifications were generated using two different speech synthesis solutions for Croatian language developed in scope of our research. The first was a statistical parametric speech synthesis system trained using a small self-recorded dataset, while the other was a WaveGlow implementation trained on 15 hours of carefully selected audio from Croatian radio shows. Survey participants were expected to grade the notifications in terms of intelligibility, grammatical correctness, and overall quality. The goal was to determine what users perceive to be more important – the generated speech quality or the grammatical correctness of the spoken content.

The next section describes the current state in research and development of speech-enabled systems and their applications in various domains. The third section describes the proposed model for spoken notifications in a smart environment, concepts related to context-awareness and required subsystems with their architectures. In the fourth section we evaluate the proposed system operating in a smart home environment and discuss the results. Finally, a conclusion is given, with plans for future research and development.

2. Speech Technologies in Smart Environments

From the human point of view, interaction with complex systems which constitute smart environments tends to be complicated. The complexity is due to presence of many different devices performing various functions across the physical environment. These devices communicate with each other and can influence each other’s actions. Regardless of the application domain, smart systems are mostly distributed, provide different communication channels and users need to conform to specific user interfaces. The most used, but also most simple and straightforward method of interaction is a traditional graphical user interface, where a human user interacts with the system by using a client application which provides insight into various system features.

However, smart environments are still primarily supporting humans and the ultimate goal should be to achieve the most intuitive way of system interaction with human users. Speech has been recognized as the most natural and efficient method of communication for humans [10]. This requires that the system is capable of receiving spoken requests and providing spoken feedback to human users. These functionalities require speech recognition and speech synthesis capabilities, respectively. However, in the context of a smart system, speech recognition and speech synthesis components represent interaction interfaces, with no cognitive processing of users' requests and system feedback. Therefore, additional components are required, the ones which could enable translation between natural language and language used by the system (e.g. system events, commands, etc.). These are typically represented as the natural language understanding and natural language generation component.

Human-computer interfaces which enable spoken interaction between a human user and the computer system have been successfully applied in various domains, from smart home systems to industrial facilities. However, their capabilities are usually limited to recognition of specific commands and reproduction of predefined spoken notifications. The rise of Intelligent Personal Assistants (IPAs) improved the situation significantly. IPAs have evolved into systems capable of leading conversations with human users, understanding linguistic and semantic context. They have also provided options for integration with various devices and external systems, making the IPAs very flexible and extensible [11]. All distinguished IPA platforms enable their users to develop new functionalities for the devices, thus creating a constantly growing ecosystem of services, but also introducing possible security and privacy threats [12]. Privacy is a serious issue in all notable IPA systems, since they constantly record and analyze private conversations due to their dependency on cloud infrastructure [13]. Despite all the mentioned advantages, the IPA approach hasn't yet achieved full integration with IoT smart platforms, in sense that it is fully capable of processing all events in the given environment and understanding its context.

IoT systems are mostly focused on enabling machine-to-machine interactions between devices, while typically relying on traditional Graphical User Interface (GUI) for interaction with human users. However, for humans, spoken interaction is more intuitive than learning all the options of a GUI or pushing buttons on specific controller devices. This has been recognized, resulting with significant advancements in speech recognition and natural language understanding in domain of smart environments. Spoken interaction in smart home applications with customizable devices has been introduced, thus expanding the system capabilities [14]. Furthermore, in addition to identifying spoken words, emotion recognition has been introduced with purpose of detecting user's mood [15]. The given examples did not focus on speech synthesis which would enable the IoT system to present basic information in spoken format back to the user. An example of a speech enabled IoT system with a combination of cloud services for speech recognition and speech synthesis is described in [16].

In case of Croatian language, there are numerous challenges. In the domain of speech technologies, Croatian language is seriously underdeveloped. Therefore, it is required to first develop usable speech recognition and speech synthesis systems which could perform adequately and provide satisfactory results, then they can be integrated with a system operating in a real-world environment. Our goal is to provide a framework which could be efficiently deployed into different smart environments and would have the ability to support a wide array of applications. Additionally, our long-term focus is the application subsystem that would enable spoken interaction between human user and the distributed computer system.

Spoken interaction between a human user and a computer requires both speech recognition and speech synthesis capabilities, along with respective cognitive processing components. Additionally, it is essential to understand user's and environmental contexts, as they represent a critical role in interaction methods which could be employed.

In scope of this work, we present the model of the complete system, but focus on requirements related to spoken notifications, which are natural language generation and speech synthesis.

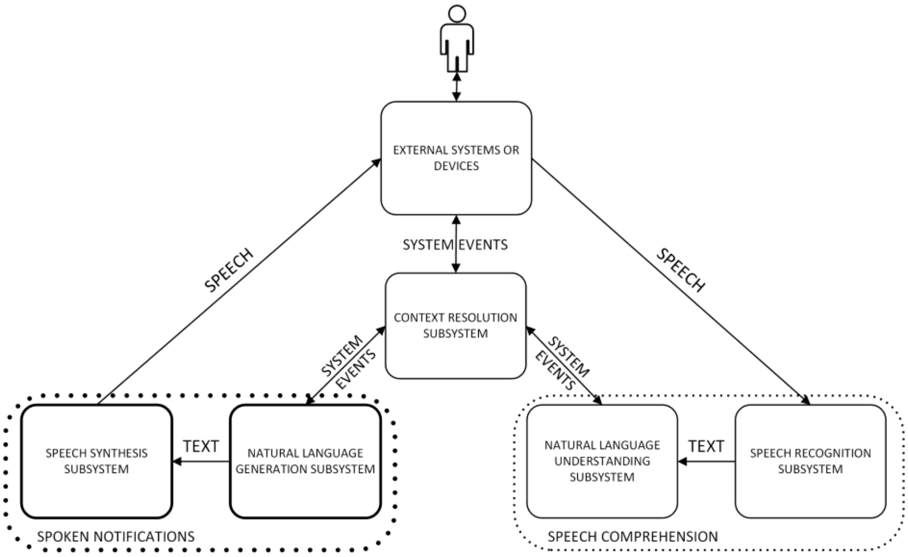

EXTERNAL SYSTEMS OR DEVICES
SYSTEM EVENTS
SPEECH
SPEECH
CONTEXT RESOLUTION SUBSYSTEM
SYSTEM EVENTS
SYSTEM EVENTS
SPEECH SYNTHESIS SUBSYSTEM
TEXT
NATURAL LANGUAGE GENERATION SUBSYSTEM
NATURAL LANGUAGE UNDERSTANDING SUBSYSTEM
TEXT
SPEECH RECOGNITION SUBSYSTEM
SPOKEN NOTIFICATIONS
SPEECH COMPREHENSION

Fig. 1. Spoken interaction in smart environments

When examining related work and the presented requirements, a proposed model of the complete system should have the building blocks as presented in Fig.1. As depicted in the figure, all external events, including speech, are captured using various external devices and sensors. Depending on the captured event, an action or a notification (or both) may follow. This is decided by the context resolution subsystem, which calculates importance of the received event and creates a notification request if necessary. In case of a spoken command, the recorded audio is forwarded to speech recognition subsystem, which transforms it into the textual transcription of the spoken sequence of words. The natural language understanding (NLU) subsystem then translates the received text into a system event which can be processed by the context resolution subsystem. This subsystem performs several challenging assignments. First, it monitors all events in the system, thus building the environmental context. Second, it monitors user activities in order to derive user context. Based on available information, context resolution subsystem can make decisions related to the state of the environment (actions or notifications), regarding both the system and the user. From the perspective of the system, context resolution subsystem tends to keep the system (and the environment) in a stable state. From the users' perspective, it enables the user to interact with the smart environment in a convenient and meaningful way.

The context resolution subsystem can decide when it is important to inform the user about something and initiate interaction. Furthermore, it provides an additional layer of understanding regarding user’s spoken requests. For instance, it will identify what the user’s context was when a spoken command was issued and deduce to which device(s) it refers. Understanding of context is crucial for human-computer interaction in smart environments as it can enable the computer system to independently decide when, how, and why to initiate interaction with the user [17]. Context is comprised of information collected from the environment through use of various sensors. This information then needs to be processed and specific context variables need to be inferred from the available data. As a result, a context descriptor is constructed, and all system components can use the newly available information. Depending on specific conditions in the target environment, a specific method of interaction may be more appropriate and more efficient than others. For instance, in a loud environment (e.g., factory setting) there is no point in using speech interface for interaction with the computer system. However, employing the visual approach by using a traditional graphical user interface makes an efficient interaction method. Additionally, in a smart home environment it is desirable to provide a method of interaction which allows users to issue requests in the most intuitive way, by using speech. However, environmental conditions are dynamic in both examples, therefore the employed interaction method needs to be adjusted according to environmental context [18].

Understanding context is essential in case of human-computer spoken interaction and presence of multi-modal interaction methods. However, this will not be discussed in detail here, since the model proposed in this paper is focused on spoken notifications from computer system to a human user.

3. Proposed Model for Spoken Notifications

Spoken notifications represent an interaction channel which enables human users to receive information from the smart IoT system in form of computer-generated speech. This functionality relies exclusively on natural language generation and speech synthesis subsystems, which are described in the following subsections.

3.1. Speech Synthesis Subsystem

Enabling speech synthesis for a given language requires multiple components performing specific functions. These components transform text from its initial form to a form enriched with information required for creation of its spoken counterpart. Speech synthesis systems have evolved significantly over the last two decades, with development of more sophisticated techniques for language [19] and acoustic modeling, and new methods for generation of audio signal [20]. These improvements were mostly related to application of novel machine learning techniques applied on large text and speech corpora.

Regarding Croatian language, speech technologies are underdeveloped, there are no commercially available systems which are designed and implemented exclusively for

Croatian language. Some available systems are solutions adapted from similar languages, such as Newton Dictate [21], a speech recognition system designed primarily for Czech, and AlfaNum TTS [22], a speech synthesis system designed primarily for Serbian, but these solutions do not fully comply with Croatian prosody and morphology. In academic circles, there have been initiatives and projects dealing with speech recognition [23] and speech synthesis [24], but without usable results from the consumer's point of view.

In the group of Slavic languages, there have been successful research efforts which have resulted with commercially available systems for both speech recognition and speech synthesis. In case of Czech and Serbian language, novel methods have been successfully applied in the field and there are commercially available systems based on recent research achievements. Even though there are many similarities between Croatian language and Czech or Serbian language, there are still specific challenges which need to be addressed. For example, Czech language is accentuated in written form, while Croatian is not. This makes speech synthesis more complicated, because in case of Croatian language pronunciation accent needs to be modelled either by employing rule-based approach or learning from available corpora. Compared to Serbian language, Croatian has different accentuation rules, essential for producing high-quality synthesized voice. Additionally, in Croatian, foreign names and words in textual form are written in original language, which makes speech synthesis more challenging.

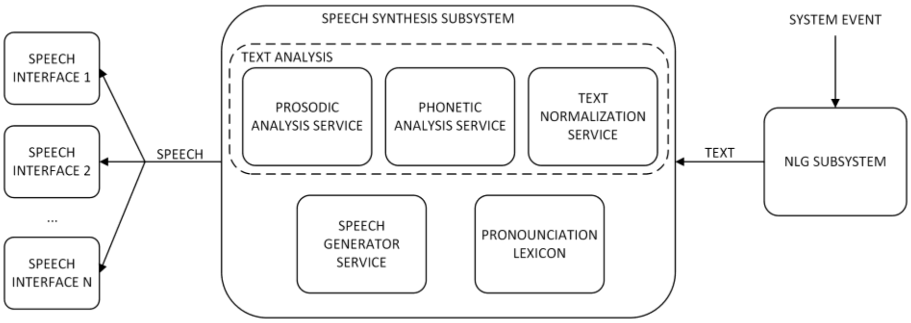


Fig. 2. Speech synthesis subsystem and related services

The proposed speech synthesis subsystem enables real-time generation of spoken notifications intended for the human user. Its purpose is to generate and reproduce spoken notifications using the convenient interaction channels. In this process, several specific tasks need to be performed. As shown in Fig. 2, these tasks can be represented with corresponding specialized services. The input text is typically received from the natural language generation subsystem, while the resulting synthesized speech can be reproduced on any available device (i.e., speech interface).

Text Analysis

Producing speech from the given text requires text analysis and transformation before the actual speech is generated. The text analysis process consists of three major steps:

text normalization, phonetic analysis and prosodic analysis. In the text normalization step, the entire text needs to be transformed into pronounceable units. This means that all non-standard words need to be identified and replaced with their corresponding fully extended counterparts. This includes numbers, time and date, acronyms, abbreviations and symbols (e.g., “7” becomes “seven”, “ºC” becomes “degree Celsius”, etc.). Text normalization is handled by employing a rule-based approach combined with a predefined dictionary. Text normalization in Croatian language is significantly more challenging than in English, because each fully extended form needs to be correct in terms of gender, case and number [25]. For this reason, once the initial normalization process is done, the result is validated by the spell-checker service which can detect and correct possible errors.

Completely normalized text is required for the next step – the phonetic analysis. In this step the text currently in form of sequence of graphemes transforms into a sequence of phonemes, which represent the content which will be pronounced. In case of native Croatian words, this transformation is straightforward because Croatian orthography is phonemic, i.e., phonetic representation of a word corresponds with its written form [26]. In English or German language this procedure is much more complicated because there are special rules required. However, Croatian language is very challenging in case of words coming from foreign languages. Regarding orthography, foreign words remain in their original form. This means that such words should be transformed to phonemes according to pronunciation rules in their original language. In this case, a pronunciation lexicon which contains phonetic transcriptions of foreign words is typically used, because foreign words are mostly names.

Prosodic analysis defines prosodic characteristics for the given content which will be synthesized. Prosodic features are duration, pitch, and intonation. Each feature is typically represented with its own model. Duration model is based on decision trees, which are used to determine phrase and sentence breaks. Pitch accent is typically applied when a word or a phrase in a sentence is emphasized. Intonation is related to variations in fundamental frequency in the given sentence. Pitch and intonation are modeled by employing machine learning algorithms on the available corpora.

Pronunciation Lexicon Service

Foreign words, names and phrases used in Croatian language retain their original orthography in the written form. However, when attempting to synthesize such content, the result is most often unrecognizable, as it does not conform to pronunciation rules present in Croatian language. For this reason, pronunciation lexicon service contains a collection of most common foreign words, names, and phrases and their phonetic transcriptions.

The lexicon approach proved itself to be the most viable option, since the alternative would be implementing language-specific pronunciation rules, which would introduce many additional challenges, such as language detection and support for language specific orthography.

Speech Generator Service

Selecting the most appropriate method for generating speech depends mostly on available resources and amount of work required. In our case, two approaches were considered – statistical parametric speech synthesis and generative neural network approach. In both cases, the speech generator relies on learning from the available data.

Statistical parametric approach is capable of building a generative model based on relatively small corpora. In the training phase, parametric representations of speech are extracted from a speech database and then modeled by using a set of generative models, typically Hidden Markov Models. During this process, linguistic units and corresponding parameters required for generating the waveform are identified. Additionally, a probabilistic model is built, whose purpose is to create parameters which were not present in the training dataset [27]. Statistical parametric speech generator was trained using a self-recorded speech corpus consisting of 654 sentences, summing up to two and a half hours of recorded speech. However, all sentences were recorded by a professional speaker. Regarding achieved results, this method provided satisfactory speech synthesis for Croatian language. Generated speech was intelligible, but not natural-sounding, with transitions between linguistic units often sounding abrupt.

Motivated by the significant improvements in quality of synthesized speech achieved by using deep neural networks for audio generation, we have decided to explore that approach. The generative approach using deep neural networks was first published in 2015 by DeepMind and has established itself as the most authentic speech synthesis method, based on English and Chinese (Mandarin) language [28]. The initial WaveNet implementation was not suitable for real-time applications because it was too computationally intensive. For instance, it would take approximately 50 seconds to generate 1 second of audio. The initial model and implementation were significantly improved in a very short amount of time, reducing the generation process to 50 milliseconds for 1 second of raw audio [29]. The generative deep neural network approach has been explored in various implementations, some of notable examples being Tacotron [30] and WaveGlow [31]. These approaches differ in underlying architectures of employed neural networks. Current results show that deep neural networks can successfully generate speech comparable to human speech in terms of intelligibility and naturalness. In case of Croatian language, there are still no notable research results or commercial systems which can produce speech of such high quality.

The main resource for developing a speech synthesis system based on generative neural networks is a large text and speech dataset (i.e., corpus) from which the system can be trained. In our case, the corpus consists of radio show recordings and corresponding transcripts obtained from Croatian Radio Television. The available corpora are radio shows in their original form, partially consisting from low-quality segments. Therefore, additional processing was required to create a dataset which could be used for training the neural network. For this purpose, additional services were developed, as displayed in Fig. 3.

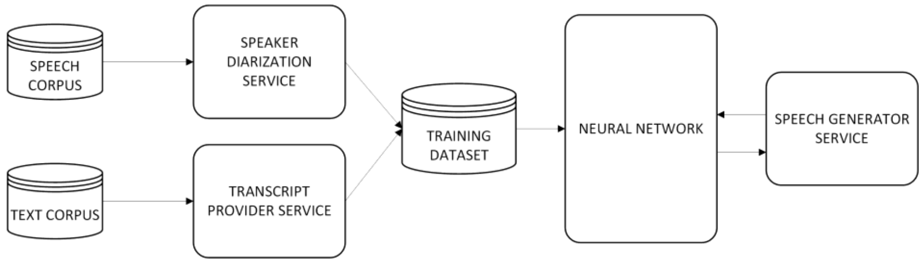


Fig. 3. Training the Speech Synthesis Subsystem

The speaker diarization service performs processing of audio files and extracts segments belonging to a designated speaker. This process is essential, because the available audio data consists of speech belonging to professional speakers (i.e., radio show hosts) and guests. Additionally, there were also segments of low-quality speech which originated from reporting by telephone. The results of speaker diarization process were audio segments organized according to speakers. The diarization is performed using adapted method provided by LIUM open source diarization toolkit [32].

The transcript provider service retrieves the text segments corresponding to the extracted audio segments. Not all available transcriptions had the timestamps which could be used for identifying the required segment, and there were also some erroneous entries. Therefore, the transcript provider service needs to combine transcripts with the original audio files in order to find correct text boundaries for the given extracted audio segments. This was achieved by analyzing spectrograms of the raw audio segments and recognizing word boundaries from the corresponding raw transcripts. The final quality of processed transcripts was satisfactory, but still lacking important content, such as some interpunction symbols like comma and colon. While they are not essential for training the generative model, these symbols are important for learning prosodic features (speed, intonation, pauses, etc.) which should be correctly reproduced from the input text.

The final step in preparation of the training dataset is removal of entries which contain foreign words and/or names. As previously explained, in Croatian language foreign names keep their original written form. Therefore, they introduce noise in the training of the neural network, as they do not conform to general rules valid in case of native words. This was performed by matching foreign names from a predefined list to the maximum extent possible and by further inspection of the prepared transcripts.

For the purpose of this research, we used approximately 15 hours of segmented audio from the radio shows with their corresponding transcripts. For speech generation, a WaveGlow implementation is used and the first results are promising, having in mind the relatively small training dataset. Evaluation of the resulting generated speech is presented in the next section.

3.2. Natural Language Generation Subsystem

Producing a spoken notification from an event which occurred in the environment requires a component capable of constructing a text sentence in human language, which human users could fully comprehend. There are several possible approaches to generating text notifications adapted for human users. They differ in terms of which input data is used as the knowledge source and the methods which are employed to construct the output text. We are using template-based approach, which is quite straightforward and the most convenient when dealing with structured input data which typically occurs in a smart environment [33]. In this context, template-based approach means that system event data (i.e. well-known elements and attributes) is extracted and organized into a raw textual representation which will be improved with help of additional language processing services.

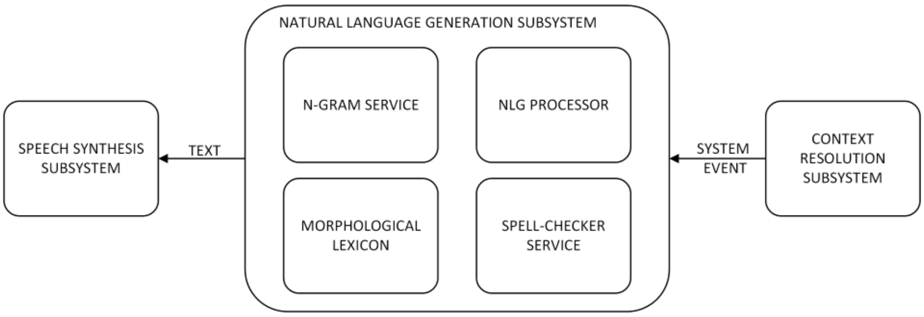

NATURAL LANGUAGE GENERATION SUBSYSTEM
N-GRAM SERVICE
NLG PROCESSOR
MORPHOLOGICAL LEXICON
SPELL-CHECKER SERVICE
SPEECH SYNTHESIS SUBSYSTEM
TEXT
SYSTEM EVENT
CONTEXT RESOLUTION SUBSYSTEM

Fig. 4. The natural language generation subsystem and related services

The natural language generation subsystem consists of several services performing specific tasks, as displayed in Fig. 4. Based on system event received from the context resolution subsystem, the NLG subsystem produces a textual notification which is then passed on to speech synthesis subsystem.

NLG Processor Service

The entire process of generating textual notifications is orchestrated by the NLG processor service. This service performs the following steps required to produce a natural sentence from synthetic event: text structuring, lexicalization and linguistic realization.

In the first step the initial notification content is generated by employing the template-based approach. The structured system event is transformed to a raw textual notification. In its initial form, the notification is certainly not correct from the linguistic perspective. In the lexicalization step, the notification is extended by substituting certain tokens with appropriate phrases. Finally, in the linguistic realization step, the notification needs to be transformed into a well-formed sentence. This includes choosing the right morphological forms, verb tenses and conjugations, and punctuation marks.

While performing the previously described steps in generating the notification, NLG processor service relies on several specialized services. In the lexicalization step, n-gram service and semantic lexicon service are used to improve the raw notification with appropriate phrases. In the linguistic realization step, morphological lexicon service is used to rectify morphological forms, which is essential for Croatian language, while spell-checker service performs final validation of the notification as a whole. These services are explained in more detail in the following section.

N-gram Service

Language models are essential components in both speech recognition and speech synthesis systems. Their purpose is to provide probability of a certain word occurring next in a given word sequence. In speech recognition, this is important for prediction of a next word which may be spoken by the human user, but also for correction of the erroneously recognized words, based on linguistic context. In speech synthesis, prediction of the following word enables better preparation of prosodic features of the generated speech.

Generally, n-grams are word sequences comprised of n words. An n-gram-based language model is constructed from a large text corpus, from which the word sequences are extracted. As a result, the model contains information about word transitions as they occur in real-life language usages. The n-gram model described in this paper was constructed by using an n-gram database obtained from a large general vocabulary text corpus collected by Hascheck, a Croatian spell-checker service. Currently, the n-gram service is based on 3-gram system, which contains word sequences comprised of three words. However, Hascheck contains n-gram collections of different lengths, with 2 <= n <= 7 [34].

An n-gram system can be represented by a directed graph, where words are represented by nodes, while connections in the graph represent the existence of a linguistic connection, with the associated probability. N-gram based language models are very useful in case of morphologically rich languages, such as Croatian. They provide information about exact word forms in the given linguistic context. For this purpose, we developed a 3-gram model consisting of approximately 2 million n-grams which was implemented into neo4j graph database. The n-grams were extracted from Hascheck's collection. This provided us the ability to query the database in order to predict the most probable following word based on the given word or a sequence of two words. Additionally, this approach enabled us to determine the probability of words preceding the given word.

Morphological Lexicon Service

Morphology is a linguistic discipline in which the smallest meaningful linguistic units (morphemes) are analyzed, including their form and their transformation depending on linguistic context. Croatian is a morphologically rich language, which means that words typically occur in various forms when used in a sentence. For instance, nouns are determined by their gender, number and case; verbs by verb tense, aspect, mood, and voice.

Morphological lexicon enables identification of word's morphological characteristics, such as word types, tenses, cases, etc. Additionally, it can provide information about the base form for any given word. In case of speech comprehension, this allows for better resolution of understanding user's intent. In scope of spoken notifications, it enables more correct and credible generated notifications, because they will sound more naturally when reproduced if the spoken words are in appropriate forms.

In domain of natural language processing, morphology is important in all related disciplines. In the process of translation from machine-generated events to natural language and vice-versa, understanding morphology plays the most important role regarding the quality of results. This makes the process of natural language understanding especially challenging, because minor errors in some word forms can cause potentially big differences in the meaning of a sentence.

Morphological lexicon was constructed using a segment of the morphological database available in Hascheck [34], summing up to approximately 700 000 entries. The implemented service provides the morphological descriptor for the given word, thus determining its exact morphological form. Additionally, it can provide the entire morphological tree for the given word.

4. Evaluation in a Smart Home Environment

The system described in previous sections was evaluated in a simulated smart home environment. The evaluation setup was designed to test how the users perceive spoken notifications of different levels of quality. For this purpose, several scenarios were examined with different proposed system components being involved. In this sense, we wanted to determine how certain components involved in the entire proposed process can affect the final result – synthesized spoken notifications.

The simulated smart-home environment consisted of three remote nodes equipped with standard sensors (temperature, humidity, microphone), interaction devices (i.e. speakers) and a central server which was collecting and processing system events. The entire process of generating textual and reproducing spoken notification starts with a specific system event received by the natural language subsystem. For example, when a temperature sensor reads a temperature above predefined threshold, it triggers an event that should result in a spoken notification to the user. We defined several templates for notifications according to the events, and the appropriate template for the examined event of temperature above threshold is:

`[`*`at`*` $timestamp, $source `*`in`*` $location `*`is`*` $value]` (in English)
`[`*`u`*` $timestamp, $source `*`u`*` $location `*`je`*` $value]` (in Croatian)

When dealing with Croatian and similar morphologically rich languages, the process of obtaining grammatically correct notification from such template becomes much more complex due to many grammatical rules. This is done by previously described services employed in the natural language generation process.

The timestamp value is transformed before it is inserted into the raw notification, as well as the Boolean type values, if any, while other values remain unchanged. A simple example notification in Croatian is given in Table 1. Each row represents a case where some or all of the NLG services are used, in order to illustrate the effect of each service on the resulting textual notification. The order of invoking the services is irrelevant since each service tackles a different task regarding the raw notification. The labels (RAW, RAW+, COR) in the table are used for easier discussion of evaluation results.

Table 1. Evaluation - NLG related services and their impact on resulting notifications

Label	Service in use?			Resulting NLG notification
	N-gram	Morphological lexicon	Spell-checker	
RAW	NO	NO	NO	U 18:35 temperatura zraka u dnevna soba je 28 stupanj Celzijev.
RAW+	NO	YES	YES	U 18:35 temperatura zraka u dnevna soba je 28 **stupnjeva Celzijevih**.
RAW+	YES	NO	YES	U 18:35 temperatura zraka u **dnevnoj sobi** je 28 stupanj Celzijev.
RAW+	YES	YES	NO	U 18:35 temperatura zraka u **dnevnoj sobi** je 28 **stupnjeva Celzijevih**.
COR	YES	YES	YES	U 18:35 temperatura zraka u **dnevnoj sobi** je 28 **stupnjeva Celzijevih**.

Notification labeled RAW is the original notification generated by the event while the label COR corresponds to the correct and final form of notification, when all the required services have processed the RAW notification. Notifications labeled as RAW+ represent cases where one of the required NLG services failed to process the notification. The differences in grammatical forms as opposed to initial RAW form are emphasized with bolded words. In this sense, the evaluation was used to determine whether the wrong grammatical form affects the intelligibility of the notification. This, in turn, shows us whether it is important to have all the required NLG services proposed in this paper.

After forming the notification, it is then passed on to the speech synthesis subsystem. Prior to this step, numerical tokens (e.g. time in Table 1) are substituted with their lexical counterparts by the text normalization service. Finally, the notification is transformed to a sequence of phonemes, prosodic features are added, and a waveform is generated.

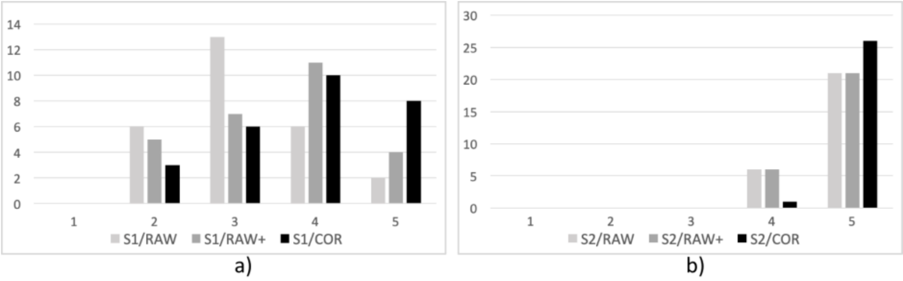
4.1. Results and Discussion

The evaluation survey was performed on 27 participants, 20 males and 7 females. Most of the participants were in their 20’s (19 users), with 6 participants from 30-40, one participant over 40 and one participant over 60 years of age. In the survey, participants were required to evaluate synthesized speech samples produced by the previously described system. Speech samples were graded on a 5-grade scale, with intelligibility, grammatical correctness, and overall quality as separate grading criteria. The goal of this evaluation is to provide feedback regarding the course of research in the future.

The evaluation dataset consisted of notifications produced by the NLG subsystem, generated from synthetic system events. The notification set contained three variations of the same message, as presented in Table 1. The first notification corresponds to the raw result generated from template (RAW). The second notification had only slight grammatical errors, corresponding to the case when one of the NLG services failed to process the notification (RAW+). The third notification was grammatically completely correct, corresponding to the case when all services were working (COR).

Furthermore, we wanted to evaluate whether grammatical correctness of the notification corresponds to the user perceived quality of the synthesized audio notification. This was done in order to see how important it is to have all the required services working and if the users would be willing to trade grammatically incorrect notifications for more human-like notifications.

For this purpose, two explored speech generation approaches were used: one based on statistical parametric speech synthesis (S1) and the other based on generative neural network approach (S2), which produced more natural, human-like speech. All notifications were synthesized using a female voice.



a) b)

Fig. 5. Intelligibility – survey results comparing S1 (a) and S2 (b)

Evaluation results regarding intelligibility are shown in Fig. 5. The obvious conclusion is that perceived intelligibility is greater in case of high-quality speech synthesis method (S2). However, it is interesting to notice that perceived intelligibility is greater in case of S2 even when grammatical errors were present. This draws us to conclusion that users value quality of synthesized speech over grammatical correctness. This is best shown in comparison of S1/RAW and S2/RAW examples, where the lowest grade for S2 was 4, while S1 received more grades in range from 2 to 3.

Results related to grammatical correctness are displayed in Fig. 6. Despite the fact that differences in grammatical correctness were recognized (rising grades between RAW, RAW+ and COR) for both speech synthesis methods, users graded the high-quality synthesizer (S2) as more grammatically correct. Interestingly, this was not the case since both synthesizers had exactly the same inputs. This is a clear indication that quality of synthesized speech is more important for the user perceived quality.

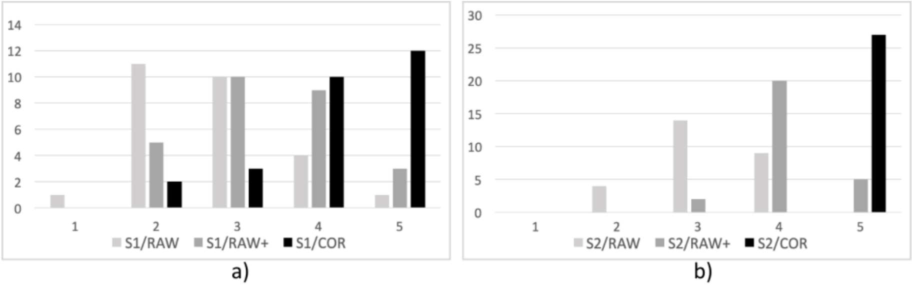


Fig. 6. Grammatical correctness - survey results comparing S1 (a) and S2 (b)

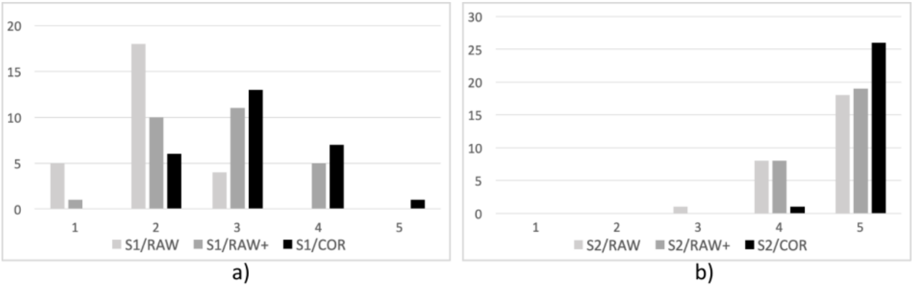


Fig. 7. Overall quality - survey results comparing S1 (a) and S2 (b)

Overall quality results are shown in Fig. 7. As expected, users regarded S2 as a far better speech synthesis method in terms of overall quality. Even in case of grammatically incorrect notification (S2/RAW), there were almost no grades below 4, which is unexpected and confirms the conclusion that users are willing to trade grammatical correctness for a more natural-sounding (or human-like) synthesized speech. On the other hand, grades regarding low-quality speech synthesis method (S1) are distributed across the entire range, with average perceived quality grade for best case (S1/COR) being 3.85, compared to 5.0 for high-quality synthesis (S2/COR).

The summary results for evaluated speech synthesis methods and notification variations are displayed in Table 2. The most interesting information is the grade difference between the two synthesis methods.

Table 2. Evaluation - NLG related services and impact on perceived quality of spoken notifications

	Intelligibility			**Grammatical Correctness**			**Overall Quality**		
	RAW	RAW+	COR	RAW	RAW+	COR	RAW	RAW+	COR
S1	3.15	3.52	3.85	2.74	3.37	4.19	1.96	2.74	3.11
S2	4.78	4.78	4.96	3.19	4.11	5.0	4.63	4.70	4.96
Diff	1.63	1.26	1.11	0.44	0.74	0.81	2.67	1.96	1.85

The intelligibility criterion directly reflects the quality of synthesized speech. Here, a considerable grade discrepancy between synthesizer S1 and S2 is present, as expected. It is interesting to notice that the smallest difference regarding intelligibility is related to the grammatically most correct notification form. This points us to a conclusion that grammatical correctness is important for intelligibility. For example, if it was not possible to develop a speech synthesizer capable of producing human-like speech for complex languages, it would definitely be required to ensure the grammatical correctness for spoken notifications, which would increase the quality perceived by users by approximately one grade.

The grammatical correctness criterion confirms that, despite low quality in case of speech synthesis method S1, the grade in its totally correct grammatical form (S1/COR) is one grade higher than the grade achieved by speech synthesis method S2 in its totally incorrect grammatical form (S2/RAW). This confirms that users noticed grammatical mistakes and corrections in case of low-quality speech synthesis method. From this we can conclude that all services described in the previous text should be present and functional when constructing the notification, regardless of the speech synthesis method which will be employed.

The differences between evaluated speech synthesis methods regarding overall quality reveal some interesting discoveries, as well. For instance, the grade range for low-quality speech synthesis method (S1) is from 1.96 to 3.11 (grade span of 1.15), while high-quality synthesis method grade span was 0.33. This again confirms that grammatical correctness affects users' quality of experience, even though in lesser extent than intelligibility.

The conclusion based on analyzed results is that both overall quality and grammatical correctness significantly affect user experience regarding evaluated spoken notifications. However, speech synthesis quality has somewhat greater influence. In case of morphologically rich languages, such as Croatian, we can conclude that grammatical correctness is especially important if there is no high-quality speech synthesis method available, since it ensures better user satisfaction in extent of an entire grade.

5. Conclusion

From the perspective of a minority language such as Croatian, designing and developing a system which can enable spoken interaction in a smart environment is a challenging endeavor. A lot of resources are required, yet there are few resources available. Additionally, it is inevitable to tackle with modeling of complex cognitive processes which are also language-related. In this paper we proposed an approach which enables us to develop a system as an orchestration of independent subsystems and their related service ecosystem. The initial results are not comparable to commercial products available for English language, but show promise.

Survey results showed that overall quality (i.e. naturalness, similarity to human speech) was the dominant factor regarding users' quality of experience. The generative neural network approach provided more natural sounding results and received better grades in all presented cases. According to grade comparison, we assume that morphological errors were perceived clearer for the better speech synthesis method, as well. Therefore, the proposed natural language generation services have an important role regarding quality of experience when differences between proper and improper morphological forms can be perceived.

Regarding future work, implementation of an improved speech synthesis service based on deep neural networks is of highest priority. The evaluated speech synthesis subsystem was created for test purposes using limited corpora and therefore has limitations regarding a broader range of applications. Future work will be focused on improving models by using larger corpora suitable for deep learning and evaluation with more example notifications and participants. After developing a fully functional system for generating natural sounding speech, the next step would be addition of speech recognition and natural language understanding subsystems to enable spoken interaction between a human user and the system. This addition would also require significant extensions to context resolution and natural language generation subsystems in order to support more meaningful discourse.

References

1. Alexakis, G., Panagiotakis, S., Fragkakis, A., Markakis, E., Vassilakis, K.: Control of Smart Home Operations Using Natural Language Processing, Voice Recognition and IoT Technologies in a Multi-Tier Architecture. Designs 3(3), 32. (2019)
2. Tadić, M., Brozović-Rončević, D., Kapetanović, A.: Hrvatski jezik u digitalnom dobu. META-NET White Paper Series. Springer, Heidelberg etc. (2012)
3. Cooper, E., Li, E.: Characteristics of text-to-speech and other corpora. In International Conference on Speech Prosody. 690–694. (2018)
4. Martinčić Ipšić, S., Matešić, M., Ipšić, I.: Korpus hrvatskoga govora. Govor, Vol. 21, No. 2, 135-150. (2004)
5. Hržica, G., Kuvač Kraljević, J.: Croatian adult spoken language corpus (HrAL). FLUMINENSIA: časopis za filološka istraživanja, Vol. 28, No. 2, 87-102. (2016)
6. Kominek, J., Black, A.W.: The CMU Arctic speech databases. In Fifth ISCA workshop on speech synthesis. (2004)
7. Ito, K.: The LJ speech dataset. https://keithito.com/LJ-Speech-Dataset/. (2020)
8. Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F.M. and Weber, G.; Common Voice: A Massively-Multilingual Speech Corpus. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), ELRA, 4218–4222. (2020)
9. Vasić, D., Brajković, E.: Development and Evaluation of Word Embeddings for Morphologically Rich Languages. In 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, 1-5. (2018)
10. Cohen, P.R., Oviatt, S.L.: The role of voice input for human-machine communication. In Proceedings of the National Academy of Sciences, 92(22), 9921-9927. (1995)
11. Berdasco, A., López, G., Diaz, I., Quesada, L., Guerrero, L.A.: User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and Cortana. In Multidisciplinary Digital Publishing Institute Proceedings, Vol. 31, No. 1, 51-59. (2019)
12. Edu, J.S., Such, J.M., Suarez-Tangil, G.: Smart Home Personal Assistants: A Security and Privacy Review. arXiv:1903.05593. (2019)
13. Ford, M., Palmer, W.: Alexa, are you listening to me? An analysis of Alexa voice service network traffic. Personal and Ubiquitous Computing, Vol. 23, No. 1, 67-79. (2019)

14. Hamdan, O., Shanableh, H., Zaki, I., Al-Ali, A.R., Shanableh, T.: IoT-based interactive dual mode smart home automation. In IEEE International Conference on Consumer Electronics (ICCE). IEEE, 1-2. (2019)
15. Fedotov, D., Matsuda, Y., Minker, W.: From Smart to Personal Environment: Integrating Emotion Recognition into Smart Houses. In IEEE International Conference on Pervasive Computing and Communications Workshops. IEEE, 943-948. (2019)
16. Petnik, J., Vanus, J.: Design of smart home implementation within IoT with natural language interface. IFAC-PapersOnLine, Vol. 51, No. 6, 174-179. (2018)
17. Lovrek, I: Context Awareness in Mobile Software Agent Network. RAD, Croatian Academy of Sciences and Arts. Technical Sciences. Vol. 513, 7-28. (2012)
18. Soic, R., Skocir, P., Jezic, G.: Agent-based system for context-aware human-computer interaction. In KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications. Springer, Cham., 34-43. (2018)
19. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y.: Exploring the limits of language modeling. arXiv:1602.02410. (2016)
20. Wang, X., Lorenzo-Trueba, J., Takaki, S., Juvela, L., Yamagishi, J.: A comparison of recent waveform generation and acoustic modeling methods for neural-network-based speech synthesis. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 4804-4808. (2018)
21. Newton Dictate, https://www.newtontech.net/en/newton-dictate/, accessed in September 2020.
22. AlfaNum, http://www.alfanum.co.rs/, accessed in September 2020.
23. Martinčić-Ipšić, S., Pobar, M., Ipšić, I.: Croatian large vocabulary automatic speech recognition. Automatika, Vol. 52, No. 2, 147-57. (2011)
24. Pobar, M., Ipšić, I.: Development of Croatian unit selection and statistical parametric speech synthesis. In Proceedings of the 34th International Convention MIPRO. IEEE, 913-918. (2011)
25. Beliga, S., Martinčić-Ipšić, S.: Text normalization for croatian speech synthesis. In Proceedings of the 34th International Convention MIPRO. IEEE, 1664-1669. (2011)
26. Načinović, L., Pobar, M., Ipšić, I., Martinčić-Ipšić, S.: Grapheme-to-Phoneme Conversion for Croatian Speech Synthesis. In 32nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2009), 318-323. (2009)
27. Zen, H., Tokuda, K., Black, A.W.: Statistical parametric speech synthesis. Speech Communication, Vol. 51, No. 11, 1039-1064. (2009)
28. Oord, A.V., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv:1609.03499. (2016)
29. Oord, A.V., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G.V., Lockhart, E., Cobo, L.C., Stimberg, F., Casagrande, N.: Parallel wavenet: Fast high-fidelity speech synthesis. arXiv:1711.10433. (2017)
30. Wang, Y., Skerry-Ryan, R.J., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q.: Tacotron: Towards end-to-end speech synthesis. arXiv:1703.10135. (2017)
31. Prenger, R., Valle, R., Catanzaro, B.: Waveglow: A flow-based generative network for speech synthesis. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 3617-3621. (2019)
32. Rouvier, M., Meignier, S.: A global optimization framework for speaker diarization. In Odyssey 2012 - The Speaker and Language Recognition Workshop. (2012)
33. Bates, M.: Models of natural language understanding. In Proceedings of the National Academy of Sciences, Vol. 92, No. 22, 9977-9982. (1995)

34. Gledec, G., Šoić, R., Dembitz, Š.: Dynamic N-Gram System Based on an Online Croatian Spellchecking Service. IEEE Access 7, 149988-149995. (2019)

Renato Šoić is a research assistant at the Department of Telecommunications of the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He received the master's degree from the University of Zagreb, in 2010. He participated in many industrial projects from different domains, including monitoring and control systems in satellite industry, mobile payment services and large-scale analytics and recommendation systems. Renato Šoić has co-authored seven conference articles and three journal articles. His research interests include speech technologies and human–computer interaction in smart environments.

Gordan Ježić is a professor at the Department of Telecommunications of the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He received the Ph.D. from the University of Zagreb in 2003. He actively participates in numerous international conferences as a paper author, speaker, member of organizing and program committees or reviewer. Gordan Ježić co-authored over 100 scientific and professional papers, book chapters and articles in journals and conference proceedings. His research interest includes telecommunication networks and services focusing on parallel and distributed systems, Machine-to-Machine (M2M) and Internet of Things (IoT) systems, mobile software agents and multi-agent systems. He is a senior member of IEEE Communication Society, IEEE FIPA, KES International, member of technical committee of IEEE SMC on Computational Collective Intelligence, and leader of technical committee of KES Focus Group on Agent and Multi-agent Systems.

Marin Vuković is an assistant professor at the Department of Telecommunications of the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He received the Ph.D. from the University of Zagreb in 2011. Marin Vuković has co-authored over 40 journal and conference papers and reviewed a number of papers for international conferences and journals. He actively participated in panels, round tables and held invited lectures with the goal of popularization of science and profession. He is a co-author of the patent at the Croatian Institute for Intellectual Property. Marin Vuković is a deputy director of "Laboratory for Security and Privacy (SPL)" and "Laboratory for Assistive Technology and Alternative and Augmentative Communication (ICT-AAC)" at the University of Zagreb, Faculty of Electrical Engineering and Computing. He is a senior member of IEEE Communications Society.

Received: April 24, 2020; Accepted: October 06, 2020

Computer Science and Information Systems 18(1):251–283 https://doi.org/10.2298/CSIS200901054S

Students' Preferences in Selection of Computer Science and Informatics Studies – A Comprehensive Empirical Case Study

Miloš Savić[1]*, Mirjana Ivanović[1], Ivan Luković[2] *, Boris Delibašić[3], Jelica Protić[4], and Dragan Janković[5]

[1] University of Novi Sad, Faculty of Sciences
Trg Dositeja Obradovića 4, Novi Sad, Serbia
{svc, mira}@dmi.uns.ac.rs

[2] University of Novi Sad, Faculty of Technical Sciences
Trg Dositeja Obradovića 6, Novi Sad, Serbia
ivan@uns.ac.rs

[3] University of Belgrade, Faculty of Organizational Sciences
Jove Ilića 154, Belgrade, Serbia
boris.delibasic@fon.bg.ac.rs

[4] University of Belgrade, School of Electrical Engineering
Bulevar Kralja Aleksandra 73, Belgrade, Serbia
jelica.protic@etf.bg.ac.rs

[5] University of Niš, Faculty of Electronic Engineering
Aleksandra Medvedeva 14, Niš, Serbia
dragan.jankovic@elfak.ni.ac.rs

Abstract. A selection of Computer Science, Informatics or similar study programs for academic studies evidently becomes an emerging choice of a vast number of students in recent years. To address some of the open questions, we performed an empirical study based on a survey, with a goal to find out the main motivating factors directing students to select computer science, informatics or similar programs for studying in a much greater extent. The survey was conducted on a sample of 1517 students from five well established, and most traditional faculties of computer science and informatics at three largest university cities in Serbia: Belgrade, Novi Sad, and Niš. The created sample is representative enough to illustrate the current situation and trends common for many similar societies. Our first analysis shows that the main motivating factor to select computer science or informatics study program at all faculties is the students' motivation to study just that topic, while at management faculty it is significantly less important. However, we also noticed that significant number of students wished to study something else but chose computer science and informatics due to possibility of finding jobs easier and of earning higher salaries in industry. The most important influential factors to choose a computer science or informatics major come from family members, and close relatives. The perceived brand and reputation of a faculty also plays a significant role. Students being examined prevalently tend to be satisfied with the faculty they have chosen. However, many of them see themselves leaving the country in a near or far future.

Keywords: students' preferences, computing education, Serbia.

*Corresponding authors

1. Introduction

Nowadays, we are the witnesses of an evident social phenomenon of an extremely raised interest of pupils from secondary schools for studying disciplines such as Computer Science (CS), Informatics, Information Technologies (IT), Information Communication Technologies (ICT), Information Systems, Software Engineering, or Computer Engineering [7]. To avoid repetitions in the text, in this paper, we denote all those disciplines with the CSISE acronym, for short. Despite the fact that these disciplines are young and rapidly developing comparing to majority of other disciplines, they have proven to be extremely popular. Many research questions arise about the main motives of newcomer students' behavior and its short and long term influences on the educational process, quality of acquired knowledge and skills, and various social impacts. We notice a similar trend in many countries all over the world with some possible positive and negative social impacts of such a trend.

What we can notice in recent years is a constant increase of figures about the number of applicants for CSISE programs in the calls for first year of B.Sc. studies. In our environment, we often notice even four or more candidates applied for one available position in CSISE study programs where candidates also apply at the same time at multiple study programs. On the other hand, there are many study programs at non CSISE areas that remain without students. Due to these circumstances, many research questions arise about the main motives and factors that influence students' selection of CSISE study programs. To identify and address the most important ones, we have performed a comprehensive empirical study at particular faculties and cities in Serbia.

The goals of our research and the performed empirical study are to: a) reveal what are the main reasons why students choose a CSISE study program; b) to what extent they are satisfied about the selected study program; and c) which sources of information they have used to make their decisions. An important question is also to find out whether the competing faculties are actually competitive, i.e. whether students from different study programs within the same university have the same attitudes towards their study programs and quality of educational process, as well as acquired knowledge and skills. This reveals how much CSISE study programs are overlapping within the same university, and consequently attract the students of the same or similar profile. This would indicate some recommendations for better profiling study programs and mitigate the similarity and competitiveness between CSISE study programs. Results of our study are additionally useful for planning effective faculties' marketing activities and admission procedures to recruit CSISE students. Despite that we completed our empirical study on a sample of Serbian students, we designed it in a way to be representative as much as possible to illustrate a current situation and trends common for many societies. As Serbia is one of South-East European and ex-communist countries with similar or almost identical higher education systems, we believe that the results presented in this paper may be even more useful in a wider area with similar higher education systems, showing practically the same problems, but not only for Serbian universities and faculties with CSISE programs.

In such a context, in the paper, we discuss on the example of Serbia some common open questions that are to be addressed in a more systematic way.

Q1: What is a real impact of the increased number of CSISE students to the local software industry?

Q2: Can academic institutions keep to the satisfactory level of quality with drastically increased number of CSISE students?

Q3: How academic institutions can preserve a sustainable education process of CSISE students?

Q4: How academic institutions can prevent a significant drop-off of education staff, and retain the majority of students at master level studies?

Q5: How to overcome or even temper significant differences in a position of academic institutions in the main city centers, compared to the academic institutions from other, usually less developed regions of the same country?

Q6: How to raise the level of motivation of CSISE students, keeping in mind that not all of them selected such study programs as their primary wish, but as a consequence of strong economical reasons?

To create a basis for addressing some of the aforementioned questions, our empirical study is based on a survey, with a goal to address the main motivating factors directing the students in Serbia to select CSISE programs for studying in a much greater extent, than many other academic disciplines. The survey was conducted on a sample of 1517 CSISE students from five well established and recognized faculties of the three largest, public universities in Serbia. By this, we selected: School of Electrical Engineering (SEE-BG) and Faculty of Organizational Sciences (FOS-BG) from University of Belgrade; Faculty of Technical Sciences (FTS-NS) and Faculty of Sciences (FS-NS) from University of Novi Sad; and Faculty of Electronic Engineering (FEE-NI) from University of Niš. 46.28% of respondents come from the University of Belgrade, where Belgrade is the largest city and capital of Serbia. 34.34% of respondents are from the University of Novi Sad, where Novi Sad is the second largest city located in northern Serbia, i.e. Autonomous Province of Vojvodina, while the rest 19.38% of them come from the University of Niš, as the third largest city, located in southern Serbia.

To the best of our knowledge, there are still no systematic and quantitatively based analyses that examined this phenomenon, and its possible impacts, not only in Serbia but also in other countries. To address some of the open questions at various levels of the society, typically just speculative, descriptive formulations are used to justify a positive impact of such trend for the overall industry development of the country, notifying how promising and influential a development of our software industry for the whole society is.

As our empirical study is performed over a sample of Serbian students, here we give some basic facts about Serbia and its higher education system. There are almost 7 million citizens by population estimates of Statistical Office of the Republic of Serbia for 2019. Serbia is a country with a system of high academic freedom at the level of universities, faculties or departments, and study programs. This means in practice that similar study programs can be found on different faculties, as faculties even within the same university develop independently similar study programs. In Serbia, we have fully profiled study programs in CSISE that typically exist in all major Serbian universities. As it is typical for many other countries, they are implemented at faculties profiled as engineering, management, or science. In some cases, programs in CSISE are even not independent ones in terms of accreditation but blended with some other disciplines. E.g., at School of Electrical Engineering from University of Belgrade, the main study program is profiled in Electrical and Computer Engineering, where CSISE is a module profiled from the second year of B.Sc. studies.

Serbia is a South-East European country, as well as post-communist, and Ex-Yugoslav country with an emerging economy, dramatic past and a challenging and long-lasting road towards joining the European Union. The government of the Republic of Serbia perceives information technologies and software industry as a strategic goal towards achieving economic sustainability and stopping brain drain that prevents Serbia to develop more rapidly. Serbia has a great potential in its universities, as there are several universities among the top 1000 ranked in the world. According to UNIVERSITAS21 [1], it was even No. 1 country in the world in 2017 in terms of their ranking when adjusted for economic development. Although CSISE study programs are well established throughout Serbian faculties, currently we could not find a systematic and reliable study analyzing why students choose a specific CSISE study program at particular faculty, and how they see their future jobs and perspectives afterwards. We have identified several difficulties preventing an easy planning and performing such analyses in a wider scope. Some of them are the following. In practice, different CSISE study programs are independently developed at almost all Serbian universities and their faculties. Even, apart from independent CSISE study programs, we can find just CSISE modules blended with some other study programs, not primarily related to CSISE, and by this it is not easy to differentiate exact data about CSISE students. The faculties develop their own staff, where resources among faculties are mostly not shared, even at the level of the same university. More importantly, these CSISE study programs at one university are actually always competitors trying to attract the best students for themselves. Students normally apply in the same call for several study programs at several institutions, and there is no any unified register of the candidate applications in the enrollment process. Each faculty organizes its own entrance exam, which is mandatory. As a rule, such entrance exams are not of the same level of content, requirements, and rigor, across different institutions. One of predominant factors for students in a selection of a study program to enroll is if a student will be granted a budget-financed place that eliminates a scholarship fee for a student. We believe that it can cover up other significant motivation factors for a selection of study program.

By addressing the aforementioned research questions, we intend to contribute to mitigating some hot problems in CSISE education process in many world-wide countries, such as how to provide satisfactory amount of well educated CSISE professionals, as there is a significant deficit of human resources in software industry, all over the world. Also, one of such problems is how to improve a selection process at universities keeping to the students' affinities and their main motivation factors, so as to attract enough high quality students for CSISE programs. Finally we intend to transfer a message that academic education of CSISE students requires a clear and sustainable strategy, rather than to be seen as a purely 'spontaneous' process, if it is recognized as one of the main pillars of digital society transformation and rapid economic development of the society.

The rest of the paper is organized as follows. Related work is presented in the subsequent section. The main method and ways of collecting data, description of sample and distribution of respondents per faculties are explained in Section 3. In Section 4 we present obtained results, their analysis and comprehensive discussion. Concluding remarks and lessons learned are given in Section 5.

2. Related Work

Understanding how pupils from secondary schools select college/faculty study programs is a well-studied area all over the world. Especially in the last two decades a lot of papers have been published analyzing gender differences and motivational factors in studying Science, Technology, Engineering, and Mathematics (STEM) disciplines [2,4,5,13]. The study of factors that influence computer science studies is a hot topic in science. It has been analyzed from many aspects, like dropout [10], career counselling [16] among others.

Also, different research studies present analysis of a wide range of factors and aspects that influence students' intentions to study CSISE programs and even aspirations to particular SC and ICT courses, e.g. [6]. On the other hand, some studies try to discover general students' satisfaction with selected study programs and faculties/collages/universities.

Choosing faculty study programs has been considered from different perspectives in a lot of countries and schools, and some of these results are presented in this section.

Soria and Stebleton [18] analyzed several aspects of students' satisfaction about selected study programs in different disciplines. First of all, they tried to discover the relationship between students' motivations for selecting faculty study programs, satisfaction with educational experience, and also their satisfaction of belonging on campus. They conducted a multi-institutional survey with students from several big public universities in US. Obtained results showed that "external extrinsic motivations for selecting a study program tend to be negatively associated with students' satisfaction and sense of belonging. Intrinsic motivations and internal extrinsic motivations tend to be positively related to students' satisfaction and sense of belonging".

Phelps et al. [15] studied the role of high-school course selection on choosing college STEM study programs. They showed that students who earned three credits in high school engineering and engineering technology courses were 1.60 times more likely to enroll in STEM study programs in four-year institutions than students who did not earn high school credits in those courses.

Yu et al. [21] used self-determination theory to study the choice of college study programs. They used SEM - Structural equation modeling, and showed that self-determined motivation, parenting styles and individual differences, motivation to study all have specific influences on choosing a major. This conclusion holds both for Eastern and Western populations.

Wang [19] identified several factors why students choose STEM study programs. Factors such as achievement in mathematics, motivation to study STEM study program, financial support in the beginning of the studies, exposure to science courses show a big impact on choosing STEM study programs.

Wegemer and Eccles [20] analyzed how gender and altruism influence the STEM career choice. They showed that altruism mediates the relationship between femininity and STEM career choice.

Malgwi et al. [12] reported several influencing factors on students choosing a faculty study program. Interest in the subject was the most important factor for both genders. For female students, aptitude in the subject is the next most important factor, where for male students the potential for career advancement, job opportunities and the salaries level were more significant.

Putnik et al. [17] presented the interesting students' opinions and correlations on satisfaction and views about computer science studies and their ambitions and expectations for future careers and jobs. Authors statistically processed data collected from extensive survey, where questionnaire contained more than 120 questions and options, conducted on a considerable sample of students from several Balkan countries.

Leppel et al. [11] studied the impact of parental occupation and socioeconomic status on choice of faculty study program. Having a father in a professional or executive occupation has a larger effect on female students than does having a mother in a similar occupation. The opposite holds for males. On the other hand, females from families with high socioeconomic status are less likely to study program in business; the opposite holds for males. Students who believe that being very well off financially is very important are more likely to choose study program in business than other students.

Montmarquette et al. [14] developed a model that showed that the expected earning mostly influences the choice of a study program. This general conclusion varies by race and gender.

Giannakos [6] shifted a little bit focus of his research to explore students' intentions to study computer science and additionally to find out the differences between programming and ICT courses. He combined several theories, including Social Cognitive Theory, Unified Theory of Acceptance, and Use of Technology, as motivating factors in students' attitude towards CS courses in a Greek university. His expectations were similar as ours: that such research can open new ways of understanding students' intentions to pursue computing and IT related careers and motivations to enroll CSISE studies.

Kori et al. [9] conducted a research rather similar to ours. They studied the reasons why students choose to study informatics. Their main intention was to analyze data and find useful guidelines on how to improve future students' recruitment and retention in informatics studies at three universities in Estonia. Main conclusions of the research were: ”the most frequent reasons for studying informatics were general interest in ICT, previous experience in the field, need for personal professional development, and importance of the field in the future“. Additionally, ”analysis showed that candidates were accepted with higher probability if they found informatics to be suitable for them, or expressed good opportunities in the labor market.“

All aforementioned research works show a plethora of different motives and factors influencing a selection of some CSISE study program. Some of the motives are highly dependent of local conditions of a particular university, country, a wider region, or a profile of target population. In Serbia, as an emerging economy, post-communist country, CSISE study programs are spread across many schools/faculties which have often very similar study programs that are allocated in different scientific fields, e.g. mathematics, engineering, or even social sciences. In such circumstances it is not always clear what are the main reasons and motives that students choose specific study programs. Additionally, for several competing influential faculties that offer different study programs in CSISE in Serbia, it is extremely important and challenging to discover these reasons and motivational factors. Comprehensive analysis and obtained results can play an essential role in attracting new generations of students to particular faculty and study program. Presented results can be also used by different universities and governmental educational policy stakeholders to carefully consider position of educational staff, properly plan marketing campaigns to attract more CSISE students, and adequately support improvement of study programs to adjust them to emerging needs of labor market and local numerous ICT companies.

3. Data Collection about CSISE Studies in Serbia

3.1. Methods

Before we preset methodology and design of questionnaire to investigate students' opinions we will briefly present school system in Serbia. It can help readers to better understand obtained results. In Serbia compulsory primary education lasts eight years. After that, pupils enter secondary education level, which consists of: 1) general grammar schools, 2) specialized grammar schools, intended for education of highly talented students, and 3) vocational schools, oriented to one of 15 different areas. All grammar schools have four-year programs, and their students can enroll at almost any faculty. On the other hand, some vocational schools last for three years only, so their students need additional year of study in order to proceed to higher education system in their specialization field. Candidates are admitted to the faculty based on secondary school grades (40% weight in total score) and the entrance exam results (60% weight in total score). Entrance exam is organized by each faculty, but it is expected to be replaced by centralized State Matura exam in 2020.

In order to investigate students' opinions about faculties and study programs in CSISE, we designed a questionnaire shown in Table 1 based on our domain expertise in the field of CSISE education. Besides common demographic questions, the questionnaire contains questions addressing:

- Primary motivating factors to study CSISE and factors for choosing a concrete CSISE faculty;
- Students' expectations and satisfaction with chosen study programs and faculties;
- How students informed themselves about CSISE faculties and study programs before enrollment and questions asking whether someone recommended the chosen faculty; and
- Students' future short-term and long-term plans and career opportunities.

Table 1. The questionnaire used to obtain students' opinions about CSISE faculties in Serbia.

Item	Question	Comments
1	Please specify (a) University, faculty and study program/module/direction (b) Study year (c) Have you renewed the current study year and how many times? (d) Gender (Male/Female) (e) City and country where you finished elementary school (f) City and country where you finished secondary school (g) Which type of secondary school have you attended? (h) Secondary school average grade (5 – excellent, 4 – very good, 3 – good, 2 – satisfactory)	Demographic questions

Continued on next page

Table 1 – *Continued from previous page*

Item	Question	Comments
2	After finishing your studies you plan to (a) continue with master studies at the same faculty (b) continue with master studies at some other faculty in Serbia (c) continue with master studies abroad (d) find an IT job in Serbia (e) find an IT job abroad (f) something else (please specify)	Multiple-choice question addressing future short-term plans
3	Have you considered enrolling informatics studies abroad before enrolling the faculty in Serbia? If yes please specify country and university.	Yes-no question. Additional comments are possible for the "Yes" answer.
4	Why have you chosen to study informatics and computer science? Please rate the relevance of the following factors. 1) Informatics has always attracted me and I feel it as my life's calling 2) I wanted to study something else, but I did not see any perspective of that profession in Serbia 3) I have chosen to study informatics since the IT industry is expanding globally and everyone talks about IT	Five-points Likert scale questions (from 1 – strongly disagree to 5 – strongly agree) addressing primary motivating factors for studying informatics
5	Did someone recommend you to enroll the chosen faculty? Please indicate whether the following persons recommended you to enroll the chosen faculty and whether their recommendation strongly influenced your faculty choice. 1) Secondary school teachers 2) Secondary school friends 3) Parents and close family 4) Current students of your faculty 5) Current students of some other faculty 6) Someone who finished your faculty 7) Someone who finished some other faculty 8) Persons working in IT sector 9) Persons working at my faculty Please indicate whether there are some other persons who recommended you to enroll your faculty. Also please indicate whether there are persons who recommended you not to enroll the chosen faculty and what were their key arguments.	Students answer by selecting one of three given answers: 1. No 2. Yes, but the recommendation did not have a major impact on my choice 3. Yes, the recommendation had a big impact on my choice

Continued on next page

Table 1 – *Continued from previous page*

Item	Question	Comments
6	Why have you chosen your current faculty to study informatics and computer science? Please rate the relevance of the following factors. 1) The study programs offered by your faculty better suit your interests and processional aspirations compared to study programs at other faculties in Serbia 2) Informatics studies at other faculties in Serbia are much more demanding and harder compared to informatics studies at your faculty 3) You heard that students of your faculty easily get well paid IT jobs 4) You thought that it would be easier to obtain state financing at your faculty than at some other faculty in Serbia 5) You were informed about possibilities to get an internship practice in IT companies during studies at your faculty 6) You heard that students of your faculty easily get jobs in foreign IT companies or go abroad for master studies 7) You heard that teachers of your faculty are competent and that the teaching content follows modern trends 8) The faculty you enrolled is considered more respectable compared to other faculties in Serbia offering informatics studies 9) Persons you consider competent nicely spoke about the faculty you enrolled 10) You heard that courses at your faculty are of better quality than courses at other informatics faculties in Serbia 11) You thought that you will be in a better position at the labor market after finishing studies at your faculty Please specify if there are additional reasons for choosing the faculty you currently study.	Five-points Likert scale (from 1 – strongly disagree to 5 – strongly agree) questions addressing factors for choosing a concrete faculty
7	Have you considered enrolling some other faculty offering informatics and computer science study programs beside the faculty you currently study? If yes please specify which faculty and why did you choose the faculty you currently study.	Yes-No question. Additional comments are possible for the "Yes" answer.
8	How and how often did you inform and get information about your faculty and study program before enrollment? Please indicate the relevance of the following information sources. 1) Secondary school friends 2) Secondary school teachers 3) Current students of my faculty 4) Former students of my faculty 5) Social media (Facebook, Twitter, etc.) 6) Faculty web site 7) Official faculty profiles on social media 8) Classic media (TV, newspapers, etc.) 9) Faculty promotional and advertising campaigns 10) Science popularization lectures 11) Preparatory lectures for student competitions 12) Seminars, courses and other extracurricular activities 13) Educational fairs	Students answer by selecting one of three given answers: 1) Never 2) Rarely 3) Frequently

Continued on next page

Table 1 – *Continued from previous page*

Item	Question	Comments
9	What are your expectations from the chosen faculty and study program? 1) To obtain knowledge enabling easier adaptations to labor market needs 2) To obtain a broad education in the study field necessary for further academic advancement (master and doctoral studies) 3) To master practical techniques and tools used in IT companies 4) To learn how to solve problems from real IT practice 5) To learn knowledge that can help me to start my own IT business 6) To obtain advice from my professors regarding my further professional development 7) To obtain also knowledge from other scientific fields that is applicable in real IT practice 8) To obtain theoretical knowledge necessary for understating and solving problems from real IT practice If you have some other expectations please specify them.	Five-points Likert scale questions (from 1 – strongly disagree to 5 – strongly agree) addressing expectations from the chosen faculty and study program
10	Has your opinion about your faculty and study program changed during your studies?	Students answer by selecting one of three given answers: 1. Yes, to better 2. No 3. Yes, to worse
11	According to your experience, how satisfied are you with the chosen faculty and study program?	Five-point Likert scale question (from 1 – completely dissatisfied to 5 – completely satisfied)
12	If you could go to the past, would you enroll the same faculty? If no please explain which faculty would you enroll.	Yes-No question. Additional comments are possible for the "No" answer.
13	Please indicate key advantages and key disadvantages of your faculty compared to other faculties offering informatics and computer science program.	Open-ended question
14	How do you see yourself in the IT sector for a long-term? 1) Freelancer 2) An IT expert working in a non-IT company 3) Employee in a Serbian IT company that makes software/hardware products for the Serbian market 4) Employee in a Serbian IT company outsourcing software/hardware products for foreign IT companies 5) Employee in a Serbian IT company designing and developing its owns software/hardware product for the global market 6) Employee in a large multinational company having a development center in Serbia 7) Employee in a small/medium foreign IT company 8) Employee in a large foreign IT company 9) IT entrepreneur home 10) IT entrepreneur abroad	Multiple-choice question addressing future long-term plans

The questionnaire was disseminated using Google Forms among students conducting CSISE study programs. The Google Forms platform allows pollsters to send a link to a questionnaire to potential respondents. The link to the questionnaire was disseminated to our students using institutional learning management systems and mailing lists, and they filled-in questionnaire voluntarily and anonymously. The questions were formulated in

Serbian. The questionnaire was live for approximately 3 months (from January to March 2018).

The reliability of collected data was assessed using the Cronbach's alpha coefficient. Then, collected students' responses were analyzed by using descriptive statistics, Pearson's correlation coefficients and non-parametric statistical tests. Non-parametric statistical tests were utilized to compare responses to a questionnaire item considering two or more independent subsamples. Subsamples were formed by various criteria: enrolled faculty, study year, gender and primary motivating factors to study informatics. The Mann-Whitney U (MWU) test and the Kolmogorov-Smirnov (KS) test were instrumented to compare two independent subsamples. MWU is a test of stochastic superiority and it examines the null hypothesis that responses in one sample do not tend to be neither higher nor lower than responses in another sample. This test is suitable for questionnaire items to which respondents provide answers on the Likert scale (e.g. question 11 and question groups 4, 6 and 9 in Table 1). To quantify the degree of a difference between two subsamples S_1 and S_2 considering responses to a Likert-scale questionnaire item Q, we examine two probabilities of superiority: the probability that a randomly selected response from S_1 is strictly higher than a randomly selected response from S_2 and the inverse probability, i.e. the probability that a randomly selected response from S_2 is strictly higher than a randomly selected response from S_1. The KS test was utilized to examine the null hypothesis that the distributions of responses of two independent samples to a questionnaire item are not significantly different. To compare more than two independent subsamples we employed the Kruskal-Wallis (KW) ANOVA test with a post-hoc pairwise comparison based on the MWU test with the Bonferroni adjustment for the p-value.

3.2. Sample

The questionnaire aimed at collecting students' opinions about Serbian faculties offering CSISE study programs was disseminated at five faculties listed in Table 2. The table also shows the number of respondents from each institution. A total of 1517 respondents is distributed as follows: 46.28% from the University of Belgrade as the largest state university in Serbia, 34.34% from the University of Novi Sad, the second largest state university in Serbia, and 19.38% from the University of Niš, the third largest state university in Serbia.

Table 2. Serbian faculties which participated in the survey.

Faculty	University	Abbrv.	#respondents
School of Electrical Engineering	University of Belgrade	SEE-Bg	434
Faculty of Organizational Sciences	University of Belgrade	FOS-Bg	268
Faculty of Electronic Engineering	University of Niš	FEE-Ni	294
Faculty of Technical Sciences	University of Novi Sad	FTS-NS	302
Faculty of Sciences	University of Novi Sad	FS-NS	219

The basic demographic characteristics of the respondents are summarized in Table 3 that shows the distribution of respondents by study year and gender. It can be seen that the distribution of respondents by study year, excluding final year students, is fairly balanced. Final year students constitute the smallest fraction of the sample, less than 5%. The distribution of respondents by gender is also relatively balanced: approximately 60% of respondents are male students and 40% of respondents are female students. Having in mind that in Serbia and several other Balkan countries, number of female students in CSISE disciplines is bigger than in other West European countries, number of female respondents in this research is more than satisfactory [17,8].

Table 3. The distribution of respondents by study year and gender.

	Study year					Gender	
	1st [%]	2nd [%]	3rd [%]	4th [%]	5th [%]	Male [%]	Female [%]
FEE-Ni	23.81	31.63	21.43	20.07	3.06	69.05	30.95
SEE-Bg	10.37	30.88	30.88	26.04	1.84	68.43	31.57
FOS-Bg	33.96	22.01	21.64	19.78	2.61	36.57	63.43
FTS-NS	29.8	27.48	14.24	20.86	7.62	60.6	39.4
FS-NS	28.31	24.2	17.81	20.09	9.59	55.71	44.29
Total [%]	23.6	27.82	22.21	21.89	4.48	59.53	40.47

A large majority of the respondents (93.21%) finished secondary school in Serbia. Our sample also contains students that finished secondary school in several other Balkan countries: Bosnia and Herzegovina (5.34%), Montenegro (1.05%), and Croatia (0.4%).

The top 10 most frequent Serbian cities our respondents come from are: Belgrade, Novi Sad, Niš, Leskovac, Šabac, Vranje, Pirot, Kruševac, Užice and Valjevo, indicating that our sample is also geographically fairly dispersed through the whole Serbia with one exception – students coming from Kragujevac as the fourth largest city in Serbia are not significantly present in our sample. This can be explained by the fact that CSISE faculties from the University of Kragujevac, have not participated in our survey.

Regarding secondary-level education, the largest fraction of our respondents finished secondary grammar school (78.78%), 11.67% of respondents finished secondary school in electrical engineering, while 9.55% of respondents obtained diploma from other vocational schools. We asked our respondents to indicate their secondary school average grade, where the grade scale is: 5 – excellent, 4 – very good, 3 – good, and 2 – sufficient. More than 86% of our respondents had excellent grades in their secondary schools suggesting that the best secondary school pupils enroll the CSISE faculties.

3.3. Reliability of Collected Data

The reliability of collected responses to our questionnaire was examined using the Cronbach's alpha coefficient [3]. This coefficient reflects the internal consistency of responses to different questions covering the same theoretical construct. The alpha coefficient higher than 0.7 signifies an acceptable level of internal consistency. Our questionnaire contains four large groups of questions reflecting four different constructs (items 5, 6, 8 and 9 in Table 1). Thus, we have computed four Cronbach's alpha coefficients:

1. α_1 – the internal consistency of responses to questions addressing faculty recommendations;
2. α_2 – the internal consistency of responses to questions assessing factors for enrolling a particular faculty;
3. α_3 – the internal consistency of responses to questions related to how students were informed about the chosen faculty prior to enrollment; and
4. α_4 – the internal consistency of responses to questions eliciting expectations from the enrolled faculty.

The obtained alpha values, $\alpha_1 = 0.6946$, $\alpha_2 = 0.7963$, $\alpha_3 = 0.7972$ and $\alpha_4 = 0.8813$, imply that the reliability of collected responses is at an acceptable level for further statistical analyses.

4. Results and Discussion

4.1. Motivation

The questionnaire contains three questions asking respondents why they have chosen to study different CSISE study programs (questionnaire item 4 in Table 1). The distributions of answers to those three questions are given in Tables 4, 5 and 6. We notice that a majority of respondents, more than 60%, feel or strongly feel informatics as their life's calling. The application of the KW ANOVA test showed that there are statistically significant differences in the distribution of responses of students from different institutions (H = 33.15, $p < 0.0001$). MWU post-hoc tests revealed that students from FEE-NI and SEE-BG more strongly feel informatics as their professional career than students from FOS-BG. Approximately 70% of FEE-NI/SEE-BG students are strongly attracted to informatics, while approximately 50% of FOS-BG students feel informatics as their life's calling.

Table 4. The distribution of responses to questionnaire item "Informatics has always attracted me and I feel it as my life's calling". SD – strongly disagree (1), D – disagree (2), N – neutral (3), A – agree (4), SA – strongly agree (5).

	SD [%]	D [%]	N [%]	A [%]	SA [%]	Mean [%]	Median [%]
FEE-Ni	6.8	5.78	16.67	36.39	34.35	3.85	4
SEE-Bg	3.69	6.68	21.89	34.33	33.41	3.87	4
FOS-Bg	5.6	14.18	27.24	35.45	17.54	3.45	4
FTS-NS	6.29	9.27	24.5	32.12	27.81	3.65	4
FS-NS	4.11	9.13	26.94	31.51	28.31	3.71	4
Total	5.21	8.7	23.07	34.08	28.94	3.72	4

Nearly half of the respondents from SEE-BG (47%) and more than half of the respondents from the rest of the institutions consider the global expansion of the IT industry as an important or very important motivating factor to study informatics (Table 6). The KW

Table 5. The distribution of responses to questionnaire item "I wanted to study something else, but I did not see any perspective of that profession in Serbia". SD – strongly disagree (1), D – disagree (2), N – neutral (3), A – agree (4), SA – strongly agree (5).

	SD [%]	D [%]	N [%]	A [%]	SA [%]	Mean [%]	Median [%]
FEE-Ni	37.41	20.75	14.29	13.27	14.29	2.46	2
SEE-Bg	47.24	16.59	12.9	16.82	6.45	2.19	2
FOS-Bg	41.79	16.42	13.43	11.94	16.42	2.45	2
FTS-NS	48.34	14.24	11.59	12.58	13.25	2.28	2
FS-NS	39.27	22.37	13.7	14.16	10.5	2.34	2
Total	43.44	17.73	13.12	14.04	11.67	2.33	2

Table 6. The distribution of responses to questionnaire item "I have chosen to study informatics since the IT industry is expanding globally and everyone talks about IT". SD – strongly disagree (1), D – disagree (2), N – neutral (3), A – agree (4), SA – strongly agree (5).

	SD [%]	D [%]	N [%]	A [%]	SA [%]	Mean [%]	Median [%]
FEE-Ni	11.56	11.56	18.03	32.31	26.53	3.51	4
SEE-Bg	12.9	10.83	29.26	34.33	12.67	3.23	3
FOS-Bg	8.96	8.58	16.42	38.81	27.24	3.67	4
FTS-NS	14.9	10.93	20.2	29.8	24.17	3.37	4
FS-NS	13.7	6.85	25.57	36.99	16.89	3.37	4
Total	12.46	10.02	22.48	34.21	20.83	3.41	4

ANOVA test indicates that there are statistically significant differences between the institutions (H = 28.43, $p < 0.0001$).The global expansion of the IT industry is significantly stronger motivating factor to study CSISE for students from FEE-NI and FOS-BG than for students from SEE-BG which incline towards a neutral opinion regarding this factor.

The most alarming finding we obtained by analyzing responses to the questionnaire Item 4. It is the percentage of CSISE students who wanted to study something else but enrolled CSISE faculties. Approximately one quarter of respondents from each institution, without statistically significant differences among institutions, wanted to study something else but they have not seen any perspective of desired professions in Serbia. This result indicates CSISE as a popular substitute for less paid university degree professions or professions that are not in demand in the local community.

All the aforementioned findings lead to the conclusion that it is essential for key education stakeholders to provide adequate strategy of higher education in the CSISE domain and work continuously on its improvement. Such strategy is to be tightly coupled with a general strategy of the society digitalization, as well as the Strategy of the development of Artificial Intelligence in the Republic of Serbia, which is recently published at the time of writing this article.

4.2. Future Plans

Two questionnaire Items 2 and 14 asked respondents about their future plans. Future short-term plans of our students are summarized in Table 7. It can be seen that the largest fraction of respondents in each institution plan to continue with master studies at the same faculty, while the fraction of those students who plan to continue with master studies at some other faculty in Serbia is significantly lower. This result indicates that students are generally satisfied with CSISE studies in Serbia and their faculty choices. However, there is a relatively large fraction of students who see their future abroad: 18.79% of respondents want to apply for a master's degree abroad, while 7.45% respondents want to find a job abroad, which is nearly a quarter of the total number of respondents. Regarding the questionnaire Item 3, 14.5% of respondents considered to study abroad before enrolling a faculty in Serbia, which additionally signifies the fact that a relatively large fraction of Serbian CSISE students see their short-term future abroad. A similar situation can be also observed with long-term plans of Serbian CSISE students (Table 8) – more than a quarter of all respondents (26.44% of the total number) see their long-term future career abroad, where 10.94% of respondents want to work in a large IT company abroad, 8.9% want to start their own business abroad, and 6.6% see themselves as employees in small/medium IT companies abroad. The most dominant students in all five institutions are those who want to pursue professional careers in development centers of multinational IT companies that are located in the local community.

Table 7. The distribution of responses given in percentages to the questionnaire Item 2 (future short-term plans).

	All	FEE-Ni	SEE-Bg	FOS-Bg	FTS-NS	FS-NS
Master studies at the same faculty	41.33	37.07	42.17	41.04	47.35	37.44
IT job in Serbia	23.4	19.73	24.65	22.01	21.85	29.68
Master studies abroad	18.79	19.05	19.12	25	16.23	13.7
IT job abroad	7.45	10.2	6.91	7.46	5.63	7.31
Master studies in Serbia at other faculty	3.36	8.16	2.53	1.49	1.66	3.2
Do not know	2.77	2.72	2.53	1.49	3.64	3.65
Something else	3.5	3.07	2.09	1.51	3.64	5.02

Apart from having a strategy for higher education in the CSISE domain, a strong necessity is to improve the current structural characteristics and maturity of ICT industry. Despite that we have strong Research & Development (R&D) ICT companies, nowadays, local ICT companies are predominantly outsourcing profiled in regard to the business model being applied. However, it is crucial to shift a focus towards the improved structure and quality of job offers at labor markets. More creative and challenging jobs are needed to keep high-quality young professionals staying in the country. Consequently, it will lead to improvements of the common values recognized in the whole society, as a crucial requirement of young professionals to stay in the country.

Table 8. The distribution of responses given in percentages to the questionnaire Item 14 (future long-term plans).

	All	FEE-Ni	SEE-Bg	FOS-Bg	FTS-NS	FS-NS
IT job, MNC dev. center in Serbia	20.83	18.71	25.58	26.49	17.55	11.87
Entrepreneur, home	11.87	13.61	11.98	10.07	12.58	10.5
IT job abroad, large company	10.94	11.56	10.37	13.43	10.26	9.13
IT job home, outsourcing	9.23	11.9	8.99	6.72	8.28	10.5
Entrepreneur, abroad	8.9	10.88	8.53	7.09	9.93	7.76
IT job home, global market	8.17	7.14	8.29	7.84	9.93	7.31
IT expert in non-IT firms	7.71	4.08	4.61	13.81	7.62	11.42
IT job abroad, small/medium company	6.6	7.14	4.61	4.85	8.94	9.13
IT job home, local market	4.88	6.12	3.92	4.85	3.97	6.39
Freelancer	4.22	5.1	4.61	1.87	4.64	4.57
Other	6.65	3.76	8.51	2.98	6.3	11.42

4.3. Recommenders and Information Sources

When planning marketing activities to attract new students, it is useful to know who actually recommends a faculty and which information sources students use to be informed about the faculty and available study programs before enrollment. Thus, we asked students about faculty recommenders and information sources (questionnaire Items 5 and 9). Faculty recommenders sorted by their impact are shown in Table 9. It can be seen that parents and close family are the most important influencers when making faculty choice decisions for students from all five institutions. More than a quarter of the respondents enrolled faculty followed the advice from their parents and close family. Former students, current students, and persons working in the IT sector are also very influential recommenders ranked in the top 4 positions among students from all five faculties. Additionally, they have more than two times higher impact compared to the fifth ranked recommenders that are secondary school teachers.

Table 9. Faculty recommenders. W – the percentage of respondents for which the corresponding recommender strongly influenced faculty choice, r(F) – the rank of the corresponding recommender for students attending faculty F.

	W	r(FEE-Ni)	r(SEE-Bg)	r(FOS-Bg)	r(FTS-NS)	r(FS-NS)
Parents & close family	25.51	1	1	1	1	1
Former students of my faculty	22.35	2	2	2	2	3
Persons working in the IT sector	19.51	3	3	4	3	4
Current students of my faculty	19.25	4	4	3	4	2
Secondary school teachers	9.23	5	6	7	6	5
Secondary school friends	8.5	6	5	8	5	8
Former students of some other faculty	6.99	7	7	6	7	7
Current students of some other faculty	5.47	8	8	5	9	9
Teachers working at my faculty	4.8	9	9	9	8	6

Table 10 shows information sources sorted by their importance to students. It can be seen that faculty web pages are the most frequently used source to inform about a faculty and study programs. This is the top ranked information source for respondents from all five faculties. Current students of a faculty are also very important information source for novice students, ranked as the second most important by respondents from 3 faculties, and among the top five information sources in all five institutions. Interesting to notice is that respondents from different faculties differently value information sources. For example, students from FTS-NS find social media and official faculty accounts on social media very important (ranked as the 2nd and 3rd), while students from FEE-NI consider those two information sources significantly less important (ranked as 6th and 7th).

Table 10. Information sources. W – the percentage of respondents which were frequently informed by the corresponding information source or frequently used it to get information about the chosen faculty, r(F) – the rank of the corresponding information source for students attending faculty F.

	W	r(FEE-Ni)	r(SEE-Bg)	r(FOS-Bg)	r(FTS-NS)	r(FS-NS)
Faculty web page	42.58	1	1	1	1	1
Current students of my faculty	30.19	2	2	2	4	3
Social media	26.83	7	4	3	2	5
Official faculty accounts on social media	24.72	6	7	7	3	2
Former students of my faculty	24.65	3	3	4	6	4
High school friends	20.96	4	5	5	5	7
High school teachers	16.35	5	6	9	7	6
Classic media (TV, newspapers)	11.14	10	9	6	11	9
Educational fairs	9.62	8	12	8	9	8
Preparatory lectures for student competitions	8.64	9	8	13	10	9
Faculty promotional and advertising campaigns	7.84	11	13	10	8	9
Public science popularization lectures	5.87	12	10	11	13	12
Seminars, courses and extracurricular activities	5.01	13	11	12	12	13

An evident conclusion is that faculties, particularly those educating ICT professionals, must improve their promotional activities in order to attract more high quality newcomers. Before all, they must provide high quality information availability on their digital/Internet platforms. Then, they need to handle a high quality alumni section, and also further strengthen a good faculty's reputation, in general.

4.4. Faculty Choice Factors

We asked respondents to indicate the relevance of various faculty choice factors (questionnaire Item 6). The obtained results are summarized in Table 11. The table shows examined factors sorted by their relevance at the level of the whole sample, where respondents evaluated factors on the Likert scale from 1 to 5.The average value of responses is taken as the measure of factor relevance, where a higher average value indicates a more relevant factor. The table also shows the rank of each factor and the median response for each institution. We notice that the importance of faculty choice factors varies between students from different institutions:

- Students from FEE-NI, FOS-BG and FTS-NS indicate the possibility to obtain well paid IT jobs after graduation as the most important factor to enroll those three faculties. This factor is also highly ranked by SEE-BG students (the 4th most important factor) and FS-NS students (the 3rd most important factor).
- Students from SEE-BG indicate the respectability of the institution as the most important reason to enroll this faculty followed by a better position at the labor market after graduation. The respectability of an institution is also important for FEE-NI and FTS-NS students, which indicate this factor as the second most important faculty choice factor.
- The most important faculty choice factor for FS-NS students is the study program and recommendation of the faculty from competent persons. The possibility to obtain well paid IT jobs is ranked as the third most important reason to enroll this faculty.

Table 11. Faculty choice factors. R – rank, M – median, from 1 meaning strongly disagree to 5 meaning strongly agree.

	All	FEE-Ni		SEE-Bg		FOS-Bg		FTN-NS		FS-NS	
	Avg.	R	M	R	M	R	M	R	M	R	M
well paid IT jobs	3.93	1	4	4	4	1	5	1	4	3	4
study programs	3.84	5	4	3	4	2	5	3	4	1	4
respectable institution	3.84	2	4	1	5	4	4	2	4	5	3
position at labor market	3.76	6	3	2	4	3	4	4	4	4	4
recommended by competent persons	3.59	2	4	5	4	5	4	5	4	2	4
job/master abroad	3.43	4	4	7	4	7	4	7	4	8	3
high quality courses	3.36	9	3	5	4	9	3	8	3	6	3
student practice	3.25	8	3	8	3	5	4	6	4	9	3
professional teachers & IT trends	3.14	7	3	9	3	8	4	9	3	7	3
easy to finish	2.07	11	2	11	1	10	3	10	2	11	2
state financing	1.91	10	2	10	1	11	1	11	1	10	3

Aforementioned results obviously indicate that a primary goal of each faculty is to increase its reputation in the following areas: a) reaching high quality and modern study programs; b) adjustments of the programs so as to follow the local industry needs and requirements; and c) attracting young, high-quality staff capable of creating better and well-motivating study conditions in the future. On the other hand, the main problem is to provide high-quality teaching staff in circumstances of significantly low salaries in academia compared to industry, while rigorous requirements for defending PhD theses and further promotions are to be met.

4.5. Expectations

Respondents were also asked to indicate their expectations from enrolled faculties and study programs (questionnaire Item 9). The obtained responses are summarized in Table 12. The table shows the examined expectations sorted by their importance that is determined according to the average response, i.e. higher values of average responses indicate more important expectations for students. It can be seen that knowledge enabling an easier adaptation to the labor market needs is the most important for students from all five institutions. Students tend to agree or strongly agree with a large majority of statements listed in the questionnaire Item 9 with one exception: SEE-BG, FTS-NS and FS-NS students have a neutral opinion about learning entrepreneurial knowledge and skills.

Table 12. Students' expectations from enrolled faculties. R – rank, M – median, from 1 meaning strongly disagree to 5 meaning strongly agree.

	All	FEE-Ni		SEE-Bg		FOS-Bg		FTN-NS		FS-NS	
	Avg.	R	M	R	M	R	M	R	M	R	M
To obtain knowledge enabling easier adaptations to labor market needs	4.28	1	4	1	4	1	5	1	4	1	4
To learn how to solve problems from real IT practice	4.1	2	4	4	4	5	4	3	4	2	4
To obtain theoretical knowledge necessary for understanding and solving problems from real IT practice	4.1	5	4	2	4	6	4	4	4	4	4
To master practical techniques and tools used in IT companies	4.09	3	4	5	4	3	5	2	4	3	4
To obtain a broad education in the field necessary for further academic advancement (master & doctoral studies)	4.07	4	4	3	4	4	4	5	4	5	4
To obtain also knowledge from other scientific fields applicable in IT practice	3.89	7	4	6	4	2	5	7	4	7	4
To obtain advice from my professors regarding my further professional development	3.77	6	4	7	3	7	4	6	4	6	4
To obtain knowledge that can help starting own IT businesses	3.36	8	4	8	3	8	4	8	3	8	3

A common problem in the CSISE education all over the world is to find a good balance between current labor market needs and ICT industry requirements on one hand side, and long-lasting fundamental knowledge that students are to gain during their study, on the other hand side. In this very dynamic area and rapid technological changes it is not easy always to cope with more technical and technological requirements of industry. Students predominantly perceive that just current technology knowledge is important for them, often neglecting long-lasting, theoretical and fundamental knowledge. Also, we notice the lack of students' entrepreneurship knowledge, which is also important for those who like to act in a proactive way in the labor market.

The phenomena of neglecting long-lasting fundamental knowledge is negatively influenced by students' and companies' expectations expressed that faculties must adjust their study programs just to highly support emerging technologies, while limiting the fundamental, theoretical knowledge and disciplines. Nowadays, educators must cope with quite complicated role of creating a good balance between salable and conceptual knowledge, as they have to prepare students to quickly and easily switch to new technologies and emergent software tools.

4.6. Satisfaction

Three questions in questionnaire address students' satisfaction with enrolled faculties and study programs. When asked whether they would enroll the same faculty again, 88.07% of respondents answered "yes", while 11.93% of them answered "no". The distribution of responses considering individual faculties is shown in Table 13. We notice that a vast majority of students from all five institutions would again enroll the same faculty. This result is the first indicator that students are generally satisfied with CSISE faculties.

Table 13. The distribution of responses given in percentages to the question "If you could go to the past, would you enroll the same faculty?"

	FEE-Ni	SEE-Bg	FOS-Bg	FTS-NS	FS-NS
Yes	86.05	87.56	91.42	90.73	84.02
No	13.95	12.44	8.58	9.27	15.98

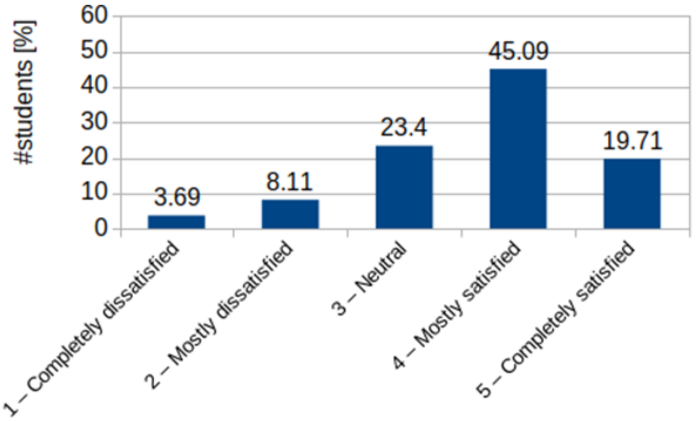
Respondents were also asked to indicate whether their opinion about enrolled faculties changed during studies. The distribution of responses is given in Table 14. There are more students whose opinion about enrolled faculties and study programs changed to better (31.05%) than those whose opinion changed to worse (26.99%). Also, it is important that in all five faculties the percentage of students whose opinion about the enrolled faculty changed to worse is higher than the percentage of students who would not again enroll the chosen faculty. Considering the whole sample, 18.39% of respondents are students who would again enroll the same faculty although their opinion about it has changed to worse. This means that at all five institutions we have a significant fraction of unsatisfied students who still think that they made the best possible choice of the faculty, i.e. they do not see any better alternative to the enrolled faculties.

Table 14. The distribution of responses given in percentages to the question "Has your opinion about your faculty and study program changed during your studies?"

	FEE-Ni	SEE-Bg	FOS-Bg	FTS-NS	FS-NS	All
Yes, to better	23.47	22.35	39.55	35.1	42.47	31.05
No	48.3	43.78	38.81	40.07	36.53	41.99
Yes, to worse	28.23	33.87	21.64	24.83	21	26.99

Students were also asked to indicate the degree of their satisfaction with chosen faculties and study programs on the scale from 1 (completely dissatisfied) to 5 (completely satisfied). The distribution of responses is shown in Figure 1. It can be noticed that the largest fraction of respondents are students who are mostly satisfied with the Serbian CSISE faculties followed by students who are neither satisfied, nor dissatisfied. Every fifth respondent is completely satisfied with CSISE studies at the chosen faculty. Less than 12% of students are mostly or completely dissatisfied with our faculties, while it is the same percentage of students that would not enroll the chosen faculty if they could go back to the past. The trend observed for the whole sample is present at the level of individual faculties (Table 15): a majority of students are mostly or completely satisfied, while 15% of students or less are completely or mostly dissatisfied with chosen faculties. Consequently, we conclude that our students generally tend to be satisfied with Serbian CSISE faculties.

Fig. 1. Students' satisfaction with chosen faculties and study programs – the distribution of responses to the questionnaire Item 11, given in percentages.



We also examined whether students' satisfaction with chosen faculties and study programs depends on the study year. The obtained results are summarized in Table 16. The table shows the average satisfaction on the scale-from 1 (completely dissatisfied) to 5 (completely satisfied) for students from different study years, as well as the results of the statistical comparison by the KW ANOVA test followed by post-hoc MWU tests for pairwise comparison. We notice that for 4 out of 5 faculties (all except FS-NS) there are statistically significant differences in satisfaction with the enrolled faculty between students from different study years. Statistically significant differences are also present at the level of the whole sample. Thus, students' satisfaction with chosen faculties is not independent of the study year. The post-hoc testing revealed that students of lower study years tend to be significantly more satisfied than students of higher study years.

Table 15. Students' satisfaction with chosen faculties and study programs per institutions, given in percentages. "Mean" shows the average satisfaction on the scale from 1 (completely dissatisfied) to 5 (completely satisfied).

	FEE-Ni	SEE-Bg	FOS-Bg	FTS-NS	FS-NS
Completely dissatisfied	7.14	3.69	0.37	3.64	3.2
Dissatisfied	8.5	9.22	9.7	5.96	6.39
Neutral	26.19	25.35	21.27	22.52	19.63
Mostly satisfied	46.26	47.93	41.42	43.71	44.29
Completely satisfied	11.9	13.82	27.24	24.17	26.48
Mean	3.47	3.59	3.85	3.79	3.84

Table 16. Statistical comparison of students from different study years regarding their satisfaction with chosen faculties and study programs (questionnaire Item 11). The column "SSD" indicates whether there are statistically significant differences. $P > Q$ in the column "Post-hoc testing" means that P-th year students tend to be significantly more satisfied than Q-th year students.

Year	1	2	3	4	5	KW ANOVA	SSD	Post-hoc testing
All	4.05	3.73	3.56	3.41	3.59	H = 86.2, $p < 10^{-4}$	yes	$1 > 2, 1 > 3, 1 > 4, 2 > 4$
FEE-Ni	3.91	3.5	3.39	2.96	3.55	H = 25.11, $p < 10^{-5}$	yes	$1 > 2, 1 > 3, 1 > 4$
SEE-Bg	4.07	3.69	3.52	3.38	3.25	H = 22.01, $p < 10^{-4}$	yes	$1 > 3, 1 > 4, 2 > 4$
FOS-Bg	4.12	3.96	3.6	3.58	3.57	H = 16.82, p = 0.002	yes	$1 > 3, 1 > 4$
FTS-NS	4.06	3.88	3.86	3.28	3.61	H = 30.3, $p < 10^{-6}$	yes	$1 > 4, 2 > 4, 3 > 4$
FS-NS	4.04	3.73	3.56	4	3.71	H = 6.37, p = 0.17	no	

The results of statistical comparison of male and female students regarding their satisfaction with chosen faculties and study programs are presented in Table 17. The table shows the average satisfaction of male and female students, results of the MWU and KS tests, as well as the values of probabilities of superiority PS(M) and PS(F). PS(M) is the probability that a randomly selected male respondent is more satisfied with the chosen faculty and study program than a randomly selected female respondent. Oppositely, PS(F) is the probability of superiority of female respondents, considering responses to the questionnaire Item 11. The null hypothesis of MWU and KS tests can be accepted considering the whole sample: $p(U) > 0.05$, $p(D) > 0.05$, PS(M) $\approx$ PS(F). The null hypothesis of the MWU test can be also accepted for all faculties except for SEE-BG, where male students express a significantly more positive opinion about the faculty compared to female students. The difference in the probabilities of superiority, PS(M) $-$ PS(F), for SEE-BG is 0.12 and it is slightly higher than the second largest difference in the probabilities of superiority (0.09 at FTS-NS) implying that the observed difference between male and female students, although statistically significant, is not too drastic. Thus, we conclude that significant differences between Serbian male and female students regarding their satisfaction with the Serbian CSISE faculties and study programs are absent.

Table 17. Statistical comparison of male and female students regarding their satisfaction with chosen faculties and study programs (questionnaire Item 11). M(Male) – the average satisfaction of male respondents, M(Female) – the average satisfaction of female respondents, U – the MWU test statistic, $p(U)$ – the p-value of U, D – the KS test statistic, $p(D)$ – the p-value of D, PS(M) – the probability of superiority of male students, PS(F) – the probability of superiority of female students. The column "SSD" indicates whether there are statistically significant differences.

	M(Male)	M(Female)	U	$p(U)$	D	$p(D)$	PS(M)	PS(F)	SSD
All	3.65	3.73	269315	0.32	0.03	0.77	0.33	0.36	no
FEE-Ni	3.42	3.59	8480.5	0.23	0.06	0.94	0.30	0.38	no
SEE-Bg	3.64	3.46	17915.5	0.03	0.10	0.33	0.40	0.28	yes
FOS-Bg	3.83	3.86	8229.5	0.86	0.07	0.92	0.36	0.35	no
FTS-NS	3.73	3.88	9871	0.15	0.10	0.43	0.30	0.39	no
FS-NS	3.84	3.84	5789	0.77	0.03	0.99	0.36	0.33	no

In general, the expressed satisfaction of the Serbian CSISE students with their selection of study programs and faculties is quite positive. However, we are facing an evident issue of a decreased level of satisfaction by study progress through higher study years. To discover exact reasons of such trend, we need to perform a deeper analysis and try to identify exact causes of the problem. The problems might be in students' greater expectations concerning study courses and covered topics, non-adequate knowledge of educators, non-adequate motivation of educators or students, general study conditions, etc. One of strongly influential reasons is the fact that many students join companies even for full time work during the third and fourth year of Bachelor studies. Evidently, we face here with numerous questions requiring an additional comprehensive analysis.

4.7. Primary Motivating Factors

Our respondents can be divided into two groups concerning primary motivating factors for enrolling the CSISE studies:

- Students strongly attracted by CSISE (denoted by G_1), i.e. students who responded with "agree" or "strongly agree" to the questionnaire item "Informatics has always attracted me and I feel it as my life's calling", and
- Students weakly attracted to CSISE (denoted by G_2), i.e. students who responded with "strongly disagree", "disagree" or "neither agree nor disagree" to the previously mentioned questionnaire item.

The results of statistical comparison of G_1 and G_2 regarding their satisfaction with chosen faculties are given in Table 18. We notice statistically significant differences between the groups at all faculties except FTS-NS. FTS-NS students strongly attracted to CSISE tend to be more satisfied with the faculty compared to FTS-NS students weakly attracted to CSISE ($M(G_1) = 3.83$, $M(G_2) = 3.72$, PS(G_1) > PS(G_2)), but the difference is not statistically significant by both employed non-parametric statistical tests

($p(U) = 0.44$, $p(D) = 0.95$). Statistically significant differences are also present between groups G_1 and G_2 at the level of the whole sample. Thus, the students strongly attracted to CSISE tend to be significantly more satisfied with Serbian CSISE faculties than students weakly attracted to CSISE.

Table 18. Statistical comparison of students strongly attracted to CSISE (group G_1) and students weakly attracted to CSISE (group G_2) regarding their satisfaction with chosen faculties and study programs. $M(G_i)$ – the average satisfaction of group G_i (i = 1 or i = 2).

	$M(G_1)$	$M(G_2)$	U	$p(U)$	D	$p(D)$	PS(G_1)	PS(G_2)	SSD
All	3.78	3.53	230978.5	0.00	0.10	0.00	0.42	0.28	yes
FEE-Ni	3.58	3.22	7057	0.00	0.21	0.01	0.46	0.25	yes
SEE-Bg	3.67	3.41	17511	0.01	0.11	0.18	0.42	0.27	yes
FOS-Bg	3.97	3.72	7435.5	0.01	0.17	0.04	0.44	0.27	yes
FTS-NS	3.83	3.72	10404.5	0.44	0.06	0.95	0.37	0.32	no
FS-NS	4.07	3.51	4102.5	0.00	0.20	0.03	0.50	0.21	yes

We additionally examined the relationship between students' satisfaction with chosen faculties and primary motivating factors for studying CSISE by comparing responses to questionnaire Item 11 between students from the following two groups:

- Group G_1 – students for which CSISE was not a desired career choice. A respondent is included in G_1 if she/he responded with "agree" or "strongly agree" to the questionnaire item "I wanted to study something else, but I did not see any perspective of that profession in Serbia".
- Group G_2 – students for which CSISE studies were the first option or one among equally desired options. G_2 encompasses students who responded with "strongly disagree", "disagree" and "neither agree nor disagree" to the previously mentioned questionnaire item.

The results of statistical comparison of G_1 and G_2 regarding their satisfaction with enrolled faculties are given in Table 19. The average satisfaction of students from G_1 is less than the average satisfaction of students from G_1 at all five faculties ($M(G_1) < M(G_2)$, PS(G_1) < PS(G_2)). Statistically significant differences between G_1 and G_2 are present at the two faculties: SEE-BG and FS-NS, while absent at the other three faculties. Statistically significant differences between G_1 and G_2 are also present at the level of the whole sample.

Thus, students who wanted to study something else but enrolled CSISE tend to be less satisfied with Serbian IT/CS compared to students who wanted to study CSISE. Taking into account the previous result, i.e. the comparison between students strongly attracted to CSISE and students weakly attracted to CSISE, we conclude that students' satisfaction with chosen faculties and study programs is not independent of primary motivating factors for enrolling CSISE.

Table 19. Statistical comparison of students who wanted to study something else, but enrolled CSISE (group G_1) and students for which CSISE studies were either the first option or one among equally desired options (group G_2) regarding their satisfaction with chosen faculties and study programs. $M(G_i)$ – the average satisfaction of group G_i (i = 1 or i = 2).

	$M(G_1)$	$M(G_2)$	U	$p(U)$	D	$p(D)$	PS(G_1)	PS(G_2)	SSD
All	3.48	3.76	184657.5	0.00	0.14	0.00	0.27	0.43	yes
FEE-Ni	3.36	3.52	7783.5	0.17	0.09	0.72	0.30	0.40	no
SEE-Bg	3.14	3.73	11371.5	0.00	0.26	0.00	0.20	0.52	yes
FOS-Bg	3.75	3.89	6501	0.14	0.10	0.67	0.30	0.41	no
FTS-NS	3.68	3.82	8214.5	0.41	0.05	1.00	0.32	0.38	no
FS-NS	3.63	3.92	3642	0.03	0.15	0.26	0.26	0.45	yes

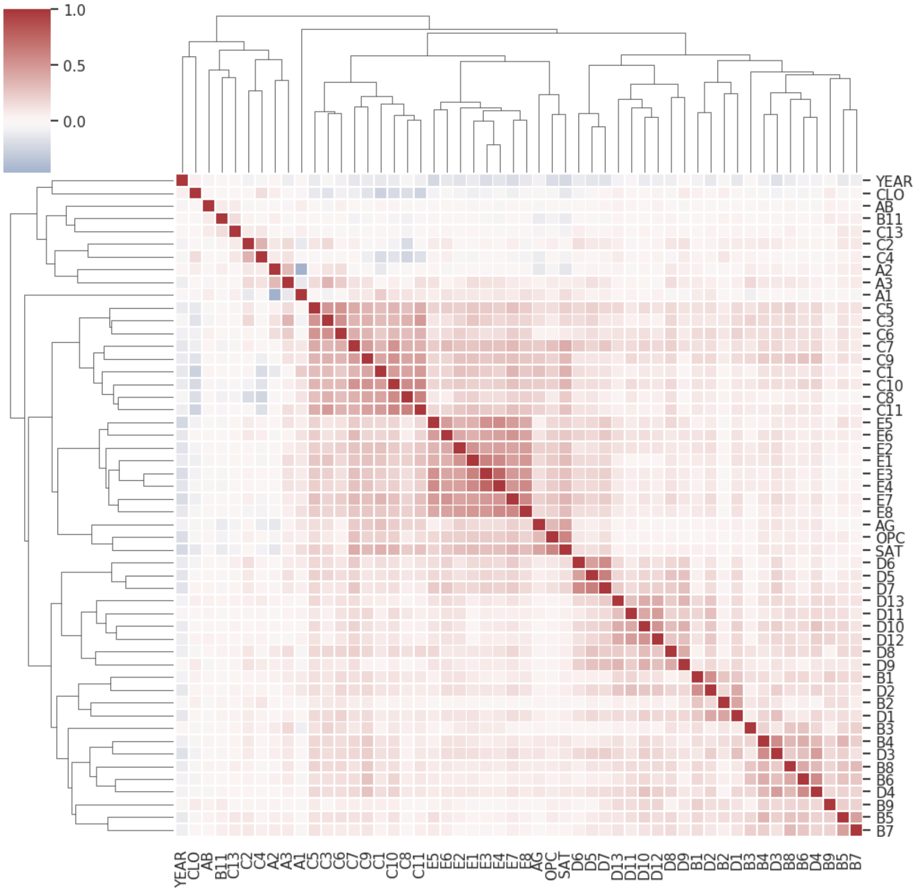
4.8. Correlation Analysis

For each pair of questions expressed on a numeric scale or on an a scale that can be converted to numeric (e.g., yes-no and yes-neutral-no questions) we have computed the Person's correlation coefficient considering given students' responses. The clustered heatmap plot of the correlation matrix is shown in Figure 2. The clusters of highly correlated responses were determined by the complete-linkage hierarchical agglomerative clustering procedure. The labeling of questions on the plot is as follows:

1. YEAR is the study year;
2. AB is questionnaire item 3 (IT-related studies abroad);
3. AG is item 12 (enrolling the same faculty again);
4. OPC is item 10 (the change of opinion about the chosen faculty during studies);
5. SAT is item 11 (satisfaction with the chosen faculty);
6. A labels mark questions given within item 4 (primary motivating factors to study informatics);
7. B labels correspond to questions given within item 5 (faculty recommenders);
8. C labels represent questions from item 6 (factors for enrolling the chosen faculty);
9. D labels are questions addressing information sources prior to faculty enrollment (item 8); and
10. E labels represent questions assessing expectations from the chosen faculty (item 9).

It can be seen that there are moderate to strong correlations within responses related to factors for enrolling the chosen faculty (C labels), within responses related to students' expectations from the chosen faculty (E labels) and within responses related to information sources (D labels). This result additionally confirms the reliability of collected data. Considering response variables belonging to different categories, moderate to strong correlations are present for B_1 and D_2 ($r = 0.54$), B_4 and D_3 ($r = 0.55$), and B_6 and D_4 ($r = 0.56$), where B_i denotes the i-th question in the category B (questionnaire item 5 related to faculty recommenders) and D_i denotes the i-th question in the category D (questionnaire item 8 addressing information sources). Those correlations indicate that secondary school teachers (B_1 and D_2), current students (B_4 and D_3) and former students (B_6 and D_4) providing a large amount of information about prospective faculties were the most influential recommenders of chosen faculties.

Fig. 2. The clustered heatmap plot of the correlation matrix for students' responses to questionnaire items.



5. Conclusion

The performed analysis shows that in majority of cases the main motivating factor to select CSISE study programs at almost all faculties is the students' motivation to study just that topic, while in FOS-BG it is significantly less important. We notify a significant number of students who initially wished to study something else but chose CSISE due to a possibility of finding easier well-paid jobs in software industry. The most important influential factors for a selection of CSISE study programs are firstly originating from family members, and close relatives, and then from current and past students of the same faculty. The perceived brand and reputation of a faculty also has a notable influence on particular selection. Students prevalently tend to be satisfied with the institutions and study programs they have chosen. However, it is worrying and important alert for key educational stakeholders that many of them see themselves leaving the country in a near (19%) or far future (26%).

Having in mind the main findings of our analysis, as well as all our previous long-year experience in the problem domain, we further discuss lessons learned in the context of research questions Q1 – Q6, given in Introduction section.

Q1: What is a real impact of the increased number of CSISE students to local software industry?

As it is a case in many emerging societies, Serbian government perceives information technologies and software industry as a strategic goal towards achieving economic sustainability and stopping brain drain. In the last decade, numerous software companies from abroad recognized a high potential of information technology and software industry in Serbia. In this way, they established their branch companies here or acquired local software companies, and by this outsourced software development activities in Serbia. It raises future expectations for constantly increasing needs for software and IT specialists in the next decade. In such circumstances, local universities are trying to find adequate ways to cope with such demands. We see that a much better approach in the future is to motivate their cooperation, instead of simple competition, as it is demanding profession with rapid development of new knowledge and technologies that require adequate education of high-quality specialists.

The outsourcing business model of software companies that seems to be predominant in economies as Serbia is, nevertheless whether a software company is R&D or just service oriented one, opens new and emergent questions:

a) Do such companies, and in what extent, really need high-quality educated professionals having fundamental knowledge and capabilities of applying critical thinking and problem solving skills necessary for long-lasting career?
b) Alternatively, do they just need employees with a deep knowledge of a particular currently popular technology?
c) What are the ratios of numbers of companies and the needs for professionals of a profile a) per numbers of companies and the needs for professionals of a profile b)?
d) How to adapt professionals of a profile b) to the new technologies, after several years when current technologies become outdated?
e) How many of university capacities are to be assigned to academic studies, and how many to professional studies to adequately address the needs for professionals of the profiles a) and b)?

Those are very sensitive and still poorly analyzed questions in such economies that lead to the conclusion of the necessity of having a sustainable and long-lasting educational strategy that will provide a maximization of positive outcomes of the local software industry for the society.

Q2: Can academic institutions keep to the satisfactory level of quality with drastically increased number of CSISE students?

As we already face the increasing number of students enrolling CSISE study programs, and as it is a dynamically changing profession, we need increasing number of teachers who are ready to constantly update their knowledge and courses to be in line with current technological and professional trends and industry needs. However, this is a very demanding activity for several reasons:

- Teachers at Serbian CSISE faculties usually have enormous number of classes and handle huge numbers of students, often significantly over predefined quota. Besides, they cope with the diversity of courses that should be delivered to students. They have been working for many years with maximal or often over maximal number of classes per week, which is 2 to 4 times higher than in majority of recognized world-wide universities.
- The process of developing teaching assistants and assistant professors is time consuming both for candidates and supervisors. For assistant professors, typically, it takes more than 10 years from the time a candidate approaches undergraduate studies to the time of a promotion to the level of assistant professor. Moreover, there is a necessity to further support a candidate during the period of an assistant professor, to gain valuable experience in teaching and research. However, in the last decade in Serbia we are facing a significant brain drain problem, where young staff leaves academia soon after obtaining their Ph.D. degrees, or even during their Ph.D. studies, as they collect their first teaching experiences as teaching assistants. One of the reasons is in significantly lower salaries comparing to the salaries of professionals in local software companies, or generally abroad. Moreover, most of them are additionally demotivated with constantly increasing and more severe requirements for promotions or even keeping current positions at Serbian faculties, from year to year. Besides, as our faculties with CSISE programs are not dedicated as "pure" computer science faculties, often criteria for promotions are much stronger or different in nature, as they are tailored from the other research disciplines, while the teaching staff from other disciplines, as a rule, is not as charged with teaching hours, as the CSISE staff.
- Increasing number of students require significantly more classes and teachers' time. They cannot manage to innovate teaching materials and include in them emergent topics, technologies, or tools.

Q3: How academic institutions can preserve a sustainable education process of CSISE students?

Evidently, strong to even radical changes in the whole educational ecosystem in a society as Serbia are necessary. One of the primary steps is to establish a sustainable system that will provide a significant increase of salaries for all education staff at faculties, as majority of education staff nowadays need to be employed in other additional jobs to increase their living standard. Consequently, they could not give an appropriate contribution in all academic activities and addressing the requirements of modern software industry. In the near future a brain drain of academic staff is expected to increase, which will make the situation even worse. Keeping in mind that budget level financing system supported by Ministry of Education, Science, and Technological Development of Republic of Serbia is of a limited capacity, a stronger and systematic involvement of interested (software) companies can contribute to the changes that will improve a position of the university teaching staff.

Q4: How academic institutions can prevent a significant drop-off of education staff, and retain the majority of students at master level studies?

This is a complex problem requiring considerable and long-lasting efforts to be solved, starting from a strategy of higher education in the CSISE domain. As a consequence of rather unfavorable economic situation in developing economies, as Serbia is, students of CSISE study programs get opportunity and decide to find a job during studies, even on the second and mostly on the third year. Being satisfied with salaries, working conditions and having no demand of employers to finish master studies, majority of them decide not to enroll master studies, and some of them even drop out of bachelor studies. The best students incline to continue with master studies abroad, expecting more academic and professional ambitions and opportunities there. Much better economic situation and living standard in well-developed countries motivate them to move and live abroad.

Q5: How to overcome or even temper significant differences in a position of academic institutions in the main city centers, compared to the academic institutions from other, usually less developed regions of the same country?

Considering example of Serbia, despite that there are three big universities in Serbia, Belgrade, Novi Sad and Niš, as well as smaller ones in other Serbian cities, it is noticeable that majority of new students incline to study in Belgrade, as a capital of Serbia. On the other hand, Belgrade is the most expensive city in Serbia, and accordingly not always convenient for young people to begin their future academic careers. The similar situation is in other countries of the region, as well as in many other developing countries.

Again, it should be a strategic goal of the government and educational policy makers to provide sustainability and better financial conditions for teaching staff at all universities throughout the country. Additionally, it is important to strengthen higher education capacities, initiate propositions of new and more attractive study programs that will support the better diversification of students at whole educational space of the country.

Q6: How to raise the level of motivation of CSISE students, keeping in mind that not all of them selected such study programs as their primary wish, but as a consequence of strong economical reasons?

Several aspects of this question are already identified in previous findings. Increasing students' motivation, generally, is not an easy task. Different technology enhanced learning tools, attractive presentations, challenging tasks, teamwork on real-life projects, practical placement in companies and real-working environments are among the numerous ways to motivate students. Obviously, all such efforts are highly time consuming and majority of educational staff is not always highly motivated to cope with them. Despite some negative factors, the results of our research indicate that students are generally satisfied with CSISE studies and their faculty choices. Besides, students see that just the knowledge enabling an easier and early adaptation to the labor market needs is the most important for them.

To conclude, a good balance between the following issues is very important:

- Higher salaries for faculty educational staff;
- Improvement of teaching methods, a balance between fundamental and technology knowledge, and continuous adjustment of teaching materials to the software industry needs;

- Cooperation and exchanges of teaching experiences among staff from all CSISE faculties, as well as the involvement of industry experts in the teaching process in a smaller extent;
- Regular meetings with company representatives and alumni to hear about their needs, expectations, wishes, to achieve continuous changing and improving education in the CSISE area; and
- Initiating and constant improving cooperation and strengthening educational and scientific networking with high quality European and world-wide universities, to improve a visibility of local universities in the area of CSISE.

Finally, if we analyze the situation in Serbia that can be generalized to the similar, particularly neighboring economies, we can say that potentials of Serbian software industry and ICT market are evident, quite strong, and constantly growing. Officially published data about the export of software products and services of Serbian software industry show that, expressed in €, it was about 100M in 2008, 300M in 2013, and even 1,1B in 2018. In 2011, there were 1,704 software companies with almost 15,000 employed and business revenue of 1,3B €, while in 2018 there were 2,349 software companies, with about 28,500 employed, and almost the doubled business revenue of 2,5B €. All of this can influence a general improvement of the society. All identified problems are seen mostly as a consequence of non-strategic decisions, or some kind of chaotic movements and actions. Therefore, a sustainable and long-lasting educational strategy is needed that will utilize the potentials of the local software industry and academic institutions to maximize the positive effects for the society. To come to the successful and sustainable strategy, more extensive analyses of not only Serbian academic education in the CSISE area, as well a local software industry, are needed, to give a better justification of the research questions discussed in this section of the paper.

References

1. UNIVERSITAS21 (U21). Available online (2019) at: https://universitas21.com/what-we-do/u21-rankings/u21-ranking-national-higher-education-systems-2019/comparison-table
2. Cheryan, S., Plaut, V.C., Handron, C., Hudson, L.: The stereotypical computer scientist: Gendered media representations as a barrier to inclusion for women. Sex Roles 69, 1573–2762 (2013), https://doi.org/10.1007/s11199-013-0296-x
3. Cronbach, L.J.: Coefficient alpha and the internal structure of tests. Psychometrika 16, 297–334 (1951)
4. Diekman, A.B., Brown, E.R., Johnston, A.M., Clark, E.K.: Seeking congruity between goals and roles: A new look at why women opt out of science, technology, engineering, and mathematics careers. Psychological Science 21(8), 1051–1057 (2010), https://doi.org/10.1177/0956797610377342
5. Eccles, J.: Who am I and what am I going to do with my life? Personal and collective identities as motivators of action. Educational Psychologist 44(2), 78–89 (2009), https://doi.org/10.1080/00461520902832368
6. Giannakos, M.: Exploring students intentions to study computer science and identifying the differences among ict and programming based courses. Turkish Online Journal of Educational Technology 13, 68–78 (07 2014)

7. Groth, D.P., MacKie-Mason, J.K.: Why an informatics degree? Commun. ACM 53(2), 26–28 (Feb 2010), https://doi.org/10.1145/1646353.1646364
8. Ivanović, M., Putnik, Z., Budimac, Z., Bothe, K., Zdravkova, K.: Gender influences on studying computer science: Non-EU Balkan case. In: Proceedings of the 6th Balkan Conference in Informatics. p. 171–178. BCI '13, Association for Computing Machinery, New York, NY, USA (2013), https://doi.org/10.1145/2490257.2490286
9. Kori, K., Pedaste, M., Niitsoo, M., Kuusik, R., Altin, H., Tonisson, E., Vau, I., Leijen, l., Mäeots, M., Siiman, L., Murtazin, K., Paluoja, R.: Why do students choose to study information and communications technology? Procedia - Social and Behavioral Sciences 191, 2867–2872 (06 2015)
10. Lacave, C., Molina, A.I., Cruz-Lemus, J.A.: Learning analytics to identify dropout factors of computer science studies through bayesian networks. Behaviour & Information Technology 37(10-11), 993–1007 (2018)
11. Leppel, K., Williams, M.L., Waldauer, C.: The impact of parental occupation and socioeconomic status on choice of college major. Journal of Family and Economic Issues 22, 373–394 (2001), https://doi.org/10.1023/A:1012716828901
12. Malgwi, C.A., Howe, M.A., Burnaby, P.A.: Influences on students' choice of college major. Journal of Education for Business 80(5), 275–282 (2005), https://doi.org/10.3200/JOEB.80.5.275-282
13. Maltese, A.V., Tai, R.H.: Pipeline persistence: Examining the association of educational experiences with earned degrees in STEM among U.S. students. Science Education 95(5), 877–907 (2011), https://onlinelibrary.wiley.com/doi/abs/10.1002/sce.20441
14. Montmarquette, C., Cannings, K., Mahseredjian, S.: How do young people choose college majors? Economics of Education Review 21(6), 543 – 556 (2002), http://www.sciencedirect.com/science/article/pii/S0272775701000541
15. Phelps, L., Camburn, E., Min, S.: Choosing stem college majors: Exploring the role of pre-college engineering courses. Journal of Pre-College Engineering Education Research (J-PEER) 8(1), Article no. 1 (2018)
16. Pordelan, N., Hosseinian, S.: Design and development of the online career counselling: a tool for better career decision-making. Behaviour & Information Technology 0(0), 1–21 (2020)
17. Putnik, Z., Štajner Papuga, I., Ivanović, M., Budimac, Z., Zdravkova, K.: Gender related correlations of computer science students. Computers in Human Behavior 69, 91 – 97 (2017), http://www.sciencedirect.com/science/article/pii/S0747563216308287
18. Soria, K.M., Stebleton, M.: Major decisions: Motivations for selecting a major, satisfaction, and belonging. NACADA Journal 33(2), 29–43 (2013), https://doi.org/10.12930/NACADA-13-018
19. Wang, X.: Why students choose STEM majors: Motivation, high school learning, and postsecondary context of support. American Educational Research Journal 50(5), 1081–1121 (2013), https://doi.org/10.3102/0002831213488622
20. Wegemer, C.M., Eccles, J.S.: Gendered stem career choices: Altruistic values, beliefs, and identity. Journal of Vocational Behavior 110, 28 – 42 (2019), http://www.sciencedirect.com/science/article/pii/S0001879118301301
21. Yu, S., Zhang, F., Nunes, L.D., Levesque-Bristol, C.: Self-determined motivation to choose college majors, its antecedents, and outcomes: A cross-cultural investigation. Journal of Vocational Behavior 108, 132 – 150 (2018), http://www.sciencedirect.com/science/article/pii/S0001879118300782

Miloš Savić is an associate professor at the Faculty of Sciences, Department of Mathematics and Informatics, University of Novi Sad, where he received his BSc, MSc and PhD degrees in computer science in 2010, 2011 and 2015, respectively. His research interests are primarily in the areas of complex network analysis and machine learning.

Mirjana Ivanović Since 2002 holds position of full professor at Faculty of Sciences, University of Novi Sad, Serbia. She is member of University Council for informatics for more than 10 years. Author or co-author is, of 13 textbooks, 13 edited proceedings, 3 monographs, and of more than 440 research papers on multi-agent systems, e-learning and web-based learning, applications of intelligent techniques, software engineering education, and most of which are published in international journals and proceedings of high-quality international conferences. She is/was a member of Program Committees of more than 200 international Conferences and General Chair and Program Committee Chair of numerous international conferences. Also she has been invited speaker at several international conferences and visiting lecturer in Australia, North Macedonia, Thailand and China. As leader and researcher she has been participated in numerous international projects. Currently she is Editor-in-Chief of Computer Science and Information Systems Journal and Associated Editor of several international Journals.

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

Boris Delibašić is a full professor at the University of Belgrade - Faculty of Organizational Sciences, Republic of Serbia. His research interests lie in data science, machine learning, business intelligence, multicriteria decision analysis, and decision support systems. He is a coordinator of the EWG-DSS. He was guest lecturer on the Friedrich Schiller University of Jena, Germany, 2006 - 2011. He was awarded with the Fulbright Visiting Scholar Grant in 2011. He has been granted projects from several research agencies (Swiss National Science Foundation, German academic exchange service, Office for Naval Research, Serbian Ministry of Science).

Jelica Protić received the Ph.D. degree in electrical engineering from the University of Belgrade in 1999. She is currently a Full Professor and the Head of Department of Computer Engineering and Informatics with University of Belgrade, the School of Electrical Engineering. With Milo Tomasevic and Veljko Milutinovic, she co-authored Distributed Shared Memory: Concepts and Systems (IEEE CS Press, 1997) and presented numerous pre-conference tutorials on this subject. She has long term experience in teaching a diversity of courses in programming languages, as well as the development of various educational software tools. Her research interests include distributed systems, consistency models, complex networks, and all aspects of computer-based quantitative performance analysis and modeling.

Dragan Janković received B.Sc., M.Sc., and a Ph.D. degree in Computer Science from the Faculty of Electronic Engineering, University of Niš, Serbia, in 1991, 1995, and 2001, respectively. Currently, he works as a full professor at the Department of Computer Science, Faculty of Electronic Engineering. His research interest includes logic design, software development, medical informatics, and blockchain technology. He was a participant and project leader for a number of research and development projects. Author or co-author more than 350 scientific papers and 10 technical solutions. He has participated in the realization of more than 30 national and international projects. He was a researching fellow of Siemens AG (Munich, Germany), Infineon Technologies AG (Munich, Germany), and ABB (Switzerland).

Received: September 01, 2020; Accepted: December 28, 2020.

Computer Science and Information Systems 18(1):285–301 https://doi.org/10.2298/CSIS200531056H

Hypothetical Tensor-based Multi-criteria Recommender System for New Users with Partial Preferences

Minsung Hong[1] and Jason J. Jung[2]*

[1] Western Norway Research Institute
Box 163, NO-6851 Sogndal, Norway
msh@vestforsk.no
[2] Department of Computer Engineering
Chung-Ang University
84 Heukseok-ro, Seoul, Korea
j3ung@cau.ac.kr, j2jung@gmail.com

Abstract. Multi-Criteria Recommender Systems (MCRSs) have been developed to improve the accuracy of single-criterion rating-based recommender systems that could not express and reflect users' fine-grained rating behaviors. In most MCRSs, new users are asked to express their preferences on multi-criteria of items, to address the cold-start problem. However, some of the users' preferences collected are usually not complete due to users' cognitive limitation and/or unfamiliarity on item domains, which is called 'partial preferences'. The fundamental challenge and then negatively affects to accurately recommend items according to users' preferences through MCRSs. In this paper, we propose a Hypothetical Tensor Model (HTM) to leverage auxiliary data complemented through three intuitive rules dealing with user's unfamiliarity. First, we find four patterns of partial preferences that are caused by users' unfamiliarity. And then the rules are defined by considering relationships between multi-criteria. Lastly, complemented preferences are modeled by a tensor to maintain an inherent structure of and correlations between the multi-criteria. Experiments on a TripAdvisor dataset showed that HTM improves MSE performances from 40 to 47% by comparing with other baseline methods. In particular, effectivenesses of each rule regarding multi-criteria on HTM are clearly revealed.

Keywords: Cold-start problem, Partial preferences, Multi-criteria recommender system, Tensor factorization.

1. Introduction

The amount of valuable data available on the Internet and the number of its users have hugely increased in the last decades. Although the data can be helpful to the users who try to find useful information such as restaurants, hotels, and museums appropriated to their interests, results provided by a search engine may be overwhelming on the Web or relevant applications [3,19]. Therefore, recommender systems have been broadly studied to cope with the information overload by providing personalized recommendations, content, and services. For instance, such systems automatically extract tourists' preferences from their explicit or implicit feedback and match features of tourism items with their needs [7,29,31].

* Corresponding author

Conventional recommendation techniques such as Collaborative Filtering (CF), as one of the most well-known and frequently adopted methods to recommend items in various fields, are typically developed based on a single rating type. However, such single-criterion recommender systems can not express and reflect fine-grained user rating behaviors, and the accuracy of those systems is often low [2]. For example, in some cases (e.g., restaurant or hotel recommendations), multiple ratings (e.g., overall, staff, or atmosphere) can often be collected to reflect various aspects of restaurants and hotels. Such multiple-criteria data would be a source of rich intelligence on item recommendations, if the data is appropriately analyzed and applied.

However, it is a non-trivial task to exploit the multiple ratings into recommendation services due to the cold-start problem that becomes more severe in the context of MCRSs. In most of the systems, new users are asked to present their preferences on some criteria of items in order to address the cold-start problem, and it could be lots of burden to users. Furthermore, some of the preferences collected are incompletely answered due to the users' cognitive limitation and/or unfamiliarity on item domains, which are called 'partial preferences' [23]. The fundamental problem thus results in low performances of MCRSs.

In this paper, we find four patterns of the 'partial preference' via data analysis in the context of MCRSs. And then a Hypothetical Tensor Model (HTM) based on three rules managing unknown users' preferences in the four patterns is proposed and is used to predict users' unobserved ratings through the Higher-Order Singular Value Decomposition (HOSVD). Simultaneously, the model keeps inherent correlations between multiple criteria. It is important using a tensor to model multiple user preferences and apply the defined rules into the model since the intuitive rules are introduced by considering relationships between users' unknown and known rating scores (i.e., multiple criteria). Experiments with a real-world dataset from Tripadvisor, which is one of the famous web review services for restaurant and hotel in tourism, show that HTM significantly outperforms than other baseline methods. Furthermore, we reveal the effectivenesses of three rules for each criterion rating. Therefore, our contributions of this paper are as follows.

- We find four patterns of partial preferences that are caused by new users' unfamiliarity in MCRSs.
- An intuitive rule set based on relationships between multi-criteria is defined to address the negative impact of unknown user preferences in the recognized patterns.
- We propose a rule-based hypothetical tensor model to improve the performance of MCRSs along with to maintain a structure of and correlations between multi-criteria
- Experimental results show better performances of the proposed method than baseline methods as well as effectivenesses of the proposed rules for each criterion.

The rest of this paper is organized as follows: In Section 2, we reviews relevant works associated with MCRSs. Section 3 introduces three intuitive rules to manage four patterns of partial preferences and proposes the hypothetical tensor model. In Section 4, we present experimental setup and evaluation protocol, while Section 5 describes in detail our empirical studies and discusses experimental results. Finally, Section 6 concludes with directions of future work.

2. Related work

Typical CF methods exploit a single rating as elements of a item-user matrix. Such techniques focus on one type (i.e., overall) of rating provided by users and suggest items to them based on preferences of their neighbors who have similar rating behaviors. Although such single rating-based approaches show a smooth and satisfying performance, as the appearance of multi-criteria recommendation techniques, one has been perceived that single criteria systems have relatively less accurate [18,5,28]. Thus, many researchers have studied to propose a new model for MCRSs [4,30,17,27].

The multi-criteria approach can be classified into memory-based and model-based methods, like CF techniques. In memory-based approach, similarities are mainly computed in two ways: one combines traditional similarity values for each criterion into a single similarity through aggregation methods (e.g., average and weighted sum) [1]. The other approach calculates distances between multi-criteria directly via multi-dimensional distance metrics (e.g., Euclidean and Manhattan). The model-based approach builds a model to predict unknown ratings and is based on the assumption that an item rating doesn't independent with other ratings and there exist relations between multi-criteria ratings. In this regard, various techniques have been used such as probabilistic modeling [25], support vector regression, multi-linear singular value decomposition [9], and genetic algorithm [10], deep neural network [22].

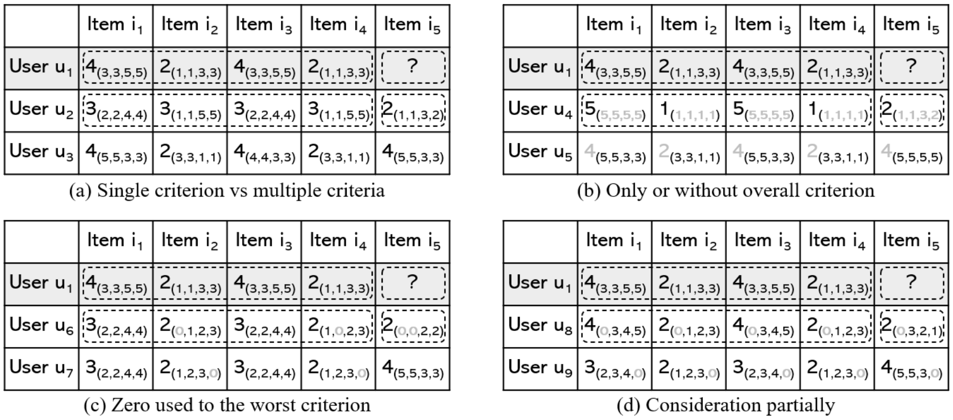
However, although an expected improvement of MCRSs could be achieved under the idea that such systems can obtain abundantly ratings for multi-criteria [21,2], it is generally difficult to get complete preferences due to users' unfamiliarity [23], such as four patterns we found in this paper, for rating scheme in real-world systems. Furthermore, it is a non-trivial task to exploit multiple ratings because of correlations between them. Therefore, MCRSs need to maintain a structure of multi-criteria and correlations between them when such systems model user preferences.

Therefore, in this paper, we find patterns of unknown rating class that are caused by new users' unfamiliarity on a rating scheme and define intuitive rules to complement the incomplete data by considering relationships between multi-criteria. And then the user preferences are expressed by a tensor model to keep the inherent structure of and the relationship between multiple ratings. Note that the proposed model is able to be applied into various domains having multiple criteria, such as restaurant, hotel and POI recommendation services.

3. Partial preference and hypothetical tensor factorization

3.1. Complementation with unknown class rules

As above-mentioned, in the context of MCRSs, users are often asked to fill multiple criteria for evaluating items based on their experiences. However, such tasks are burdens for users and result in incomplete multiple ratings due to users' unfamiliarity on the evaluation scheme. To alleviate the above problem, we intuitively define three rules for unknown rating class by an analysis of users' preferences on multiple criteria with examples. Figure 1 illustrates the reason how MCRSs can be more accurate than single-criterion based recommendations and presents potential patterns caused by users who are unfamiliar to



	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	?
User u_2	$3_{(2,2,4,4)}$	$3_{(1,1,5,5)}$	$3_{(2,2,4,4)}$	$3_{(1,1,5,5)}$	$2_{(1,1,3,2)}$
User u_3	$4_{(5,5,3,3)}$	$2_{(3,3,1,1)}$	$4_{(4,4,3,3)}$	$2_{(3,3,1,1)}$	$4_{(5,5,3,3)}$

(a) Single criterion vs multiple criteria

	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	?
User u_4	$5_{(5,5,5,5)}$	$1_{(1,1,1,1)}$	$5_{(5,5,5,5)}$	$1_{(1,1,1,1)}$	$2_{(1,1,3,2)}$
User u_5	$4_{(5,5,3,3)}$	$2_{(3,3,1,1)}$	$4_{(5,5,3,3)}$	$2_{(3,3,1,1)}$	$4_{(5,5,5,5)}$

(b) Only or without overall criterion

	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	?
User u_6	$3_{(2,2,4,4)}$	$2_{(0,1,2,3)}$	$3_{(2,2,4,4)}$	$2_{(1,0,2,3)}$	$2_{(0,0,2,2)}$
User u_7	$3_{(2,2,4,4)}$	$2_{(1,2,3,0)}$	$3_{(2,2,4,4)}$	$2_{(1,2,3,0)}$	$4_{(5,5,3,3)}$

(c) Zero used to the worst criterion

	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	$4_{(3,3,5,5)}$	$2_{(1,1,3,3)}$	?
User u_8	$4_{(0,3,4,5)}$	$2_{(0,1,2,3)}$	$4_{(0,3,4,5)}$	$2_{(0,1,2,3)}$	$2_{(0,3,2,1)}$
User u_9	$3_{(2,3,4,0)}$	$2_{(1,2,3,0)}$	$3_{(2,3,4,0)}$	$2_{(1,2,3,0)}$	$4_{(5,5,3,0)}$

(d) Consideration partially

Fig. 1. Partial preference patterns

the rating scheme of MCRSs. Given a rating scheme ranged from 1 (worst) to 5 (best), let's assume that five ratings including overall rating are obtained for multi-criteria. In the (a) of the figure, a recommender system using only overall rating will select the user u_3, who has similar rating behaviors to the target user u_1. However, the user u_3 oppositely rates multi-criteria of the items by comparing with the target user actually. On the other hand, user u_2 has more similar behaviors with of user u_1, than the user u_3 in the context of MCRSs. Thus, the accuracy of MCRSs is often higher than of single-criterion based recommendations. Although this improvement can be achieved under the assumption that such systems can obtain abundantly ratings for multi-criteria, it is difficult to get complete preferences due to users' unfamiliarity for rating scheme in fact. Even though, these unknown ratings will be predicted by MCRSs, it makes sparsity problem severe.

As shown in the Figure, we found four patterns, which frequently caused by the unfamiliarity issue, via data analysis of users' multiple ratings. The patterns illustrated by the examples (b), (c), and (d) in Figure 1 are as follows. In the examples, ratings with gray color in brackets indicate unknown values.

- The first and second patterns are that users often input only or without overall ratings. There are two sub-cases of the input case of only overall ratings: unambiguous and ambiguous ratings. The former includes ratings 1 and 5 considered as clearly worst and best evaluations. The other contains other ratings (i.e., 2, 3, and 4 in the above-mentioned rating scheme). For instance, it could be clearly considered that the unknown ratings of user u_4 are 5 or 1. In this regard, the user u_4 will be used to predict unknown ratings of user u_1.
- The other cases are less occurred than the above one, but it affects the accuracy of MCRSs also. The third one is that users sometimes do not input any ratings to express their worst preference. This pattern has mainly occurred with low overall ratings. In the case of example (c), let's assume that the users u_6 and u_7 used zero (not input any ratings) to show their worst experience. As a result, although u_7 has different multiple ratings with the target user u_1, both users selected on the recommendation process if a MCRS does not take this pattern into account.

- The last one relates to users who do not consider some of multi-criteria. It could happen due to individual users' criteria or ambiguous experience. That is, a user may do not enter some ratings because she/he does not care the relevant criteria or needs too much effort because of difficult decisions. In this regard, if users u_8 and u_9 do not care the first and last criterion respectively, selecting only the user u_8 would be appropriate to predict preferences of the target user u_1 to item i_5.

To alleviate negative effects of the above patterns, we propose three rules that are based on other filled multiple ratings (i.e., relationships between multi-criteria). Given overall rating r_0 and n number of the other multiple ratings $r_{k\in\{1,2,\ldots,n\}}$ with a rating scheme (1 to N), a generalized rule function $\mathcal{RU}$ is defined by

$$\mathcal{RU}() = \begin{cases} r_0 & \mathcal{R}_1\text{: if } \sum_{k=1}^{n}[r_k{=}\emptyset]{=}n\wedge(r_0{=}1\vee r_0{=}N) \\ 1 & \mathcal{R}_2\text{: if } \sum_{k=0}^{n}[r_k{=}\emptyset]{\leq}(n{+}1)/2\wedge\arg(r_{\forall k}){\leq}N{\times}0.25 \\ N & \mathcal{R}_3\text{: if } \sum_{k=0}^{n}[r_k{=}\emptyset]{\leq}(n{+}1)/2\wedge\arg(r_{\forall k}){\geq}N{\times}0.75, \end{cases} \tag{1}$$

where the $[\mathcal{P}]$ indicates the "Inverson bracket notation" that returns 1 if the condition $\mathcal{P}$ is true. As a summary, the $\mathcal{R}_1$ is mainly applied to the first pattern. While the second and last patterns are handled by the $\mathcal{R}_2$ and $\mathcal{R}_3$, the third one is managed by $\mathcal{R}_2$.

3.2. Hypothetical tensor model

This section describes a structure of the proposed HTM based on the rule function above-defined. Traditional Matrix Factorization (MF) techniques based on a two-dimensional user-item matrix are based on the idea that the overall ratings are generated by users' and items' latent factors. However, the assumption may fail to comprehensively represent a structure of and relationships between the latent factors [6], since it neglects considering multiple factors. Whereas, tensor models as a matrix generalization have been used to predict missing ratings along with maintaining a multi-dimensional structure of data, as it can consider the interdependency between multiple factors such as users, items, contexts, and so on in the research field of recommender systems [24,13,12,11].

In this paper, the context factors indicate multiple-criteria (i.e., rating types). Therefore, the HTM simply has three orders (i.e., user × item × rating type) as shown in (a) of Figure 2. The illustrations (b) and (c) represent a normal model and a HTM based on the proposed rule for unknown ratings in four patterns of partial preferences. In these examples, users and items are equal to of patterns in Figure 1 and the number of multiple ratings except for overall rating is 4. Note that we only illustrate users relate to the proposed rules to save space. White color represents unknown ratings. Also, light and dark gray colors indicate ratings filled by users and ratings complemented by the proposed rules, respectively. As a result, the density of the proposed hypothetical tensor model becomes higher than that of the normal tensor, and it helps to improve the accuracy of MCRSs as will be showed in Section 5.

3.3. Tensor factorization

This section defines factorization problem of the proposed HTM on prediction of unobserved users' preferences to items. Given I users, J items, and K rating types (i.e.,

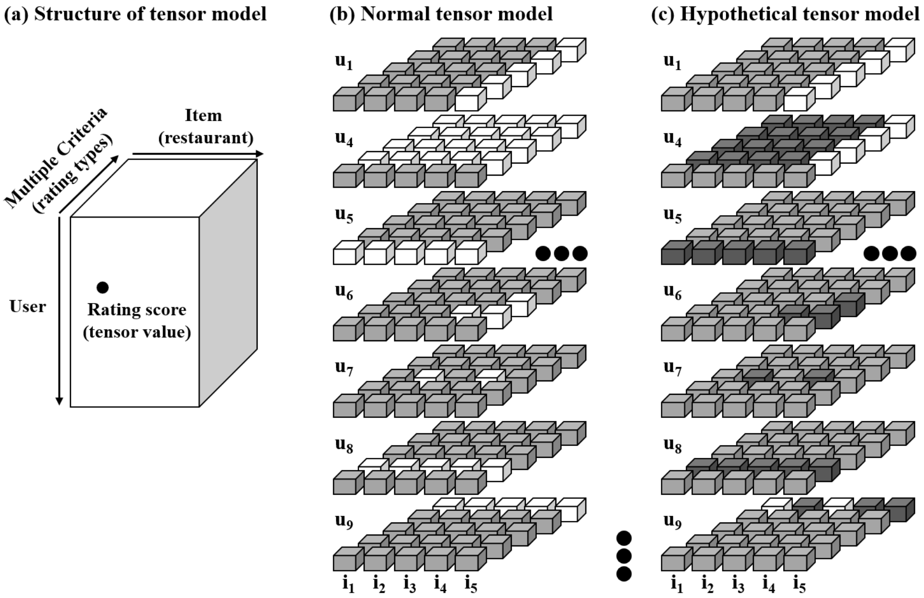

(a) Structure of tensor model
(b) Normal tensor model
(c) Hypothetical tensor model
Multiple Criteria (rating types)
Item (restaurant)
User
Rating score (tensor value)
u1
u4
u5
u6
u7
u8
u9
i1 i2 i3 i4 i5

Fig. 2. Structure of rule-based hypothetical tensor model

multi-criteria including overall criterion), the proposed model $\mathcal{H}$ is defined as follows:

$$\mathcal{H} = \{H_{ijk}\} \in \mathbb{R}^{I \times J \times K}, \tag{2}$$

where the value H_{ijk} indicates a k^{th} rating of i^{th} user for j^{th} item.

Like a conventional tensor factorization, our problem is how to minimize loss between observed and approximate tensors with considering regularization risks. Therefore, we aim to minimize a loss function $L(H_{ijk}, \hat{H}_{ijk})$, where the original and approximate users' ratings for k^{th} criterion of items are H_{ijk} and $\hat{H}_{ijk}$. For better generalization performances, a regularization term $\Omega(H_{ijk})$ is also added to the loss function. Thus, a final loss function is $L(H_{ijk}, \hat{H}_{ijk}) + \Omega(H_{ijk})$ along with least squares loss function $L()$ and Frobenius norm Ω as standard choices. Also, $\times_U$ represents a tensor-matrix multiplication operator, where the subscript indicates a direction of the tensor on which the matrix is multiplied. Additionally, entries of a i^{th} row of the matrix U are denoted by U_{i*}. Therefore, the loss function of the HTM for tensor factorization is defined by

$$\begin{aligned} F(\mathcal{H}, \mathcal{S}, U, I, T) \quad &= 1/2 \|\mathcal{S} \times_U U \times_I I \times_C C - \mathcal{H}\|_F^2 \\ &+1/2 \left[\lambda_U \|U\|_F^2 + \lambda_I \|I\|_F^2 + \lambda_C \|C\|_F^2\right], \end{aligned} \tag{3}$$

where $\mathcal{S} \in \mathbb{R}^{d_U \times d_I \times d_C}$ represents a central tensor; $\|\cdot\|_F^2$ indicates the Frobenius norm; $U \in \mathbb{R}^{I \times d_U}$, $I \in \mathbb{R}^{J \times d_I}$, and $C \in \mathbb{R}^{K \times d_C}$ are matrices of users, items, and criteria; d_U, d_I, and d_C are parameters adjusting the dimensionality of latent factors; λ_U, λ_I, and λ_C are the regularization parameters.

Because of the absence of a closed-form solution for the minimization of Eq. (3), the loss function is minimized by Stochastic Gradient Descent (SGD). Algorithm 1 shows the procedures of tensor factorization by using Higher Order Singular Value Decomposition (HOSVD) [13] for the proposed HTM, where the gradients of our objective function can be calculated as follows:

$$\begin{aligned}\eta\partial_{U_{i*}}F^L &= (\hat{\mathcal{H}}_{ijk} - \mathcal{H}_{ijk}) \times \mathcal{S} \times_I I_{j*} \times_C C_{k*},\\ \eta\partial_{I_{j*}}F^L &= (\hat{\mathcal{H}}_{ijk} - \mathcal{H}_{ijk}) \times \mathcal{S} \times_U U_{i*} \times_C C_{k*}, \text{ and}\\ \eta\partial_{C_{k*}}F^L &= (\hat{\mathcal{H}}_{ijk} - \mathcal{H}_{ijk}) \times \mathcal{S} \times_U U_{i*} \times_I I_{j*}.\end{aligned} \tag{4}$$

This algorithm linearly scales to the number of rating values R and iteration number L

Algorithm 1: Factorization of hypothetical tensor model

```
h Data: observed tensor $\mathcal{H}$, learning rate $t_0$, torelance $tol$, maxEpoch $maxEpo$
        regularization parameters $\lambda = \lambda_U = \lambda_R = \lambda_C$
Result: approximate tensor $\hat{\mathcal{H}}$
Initialize $\mathcal{H}, \mathcal{S}, U, I, C$ with zero ;
Set $l = 0$, $t = t_0$, $tol = 5$, and $maxEpo = 10$;
while not converged and $l < maxEpo$ do
    $\eta = 1/\sqrt{t}$ and $t = t + 1$ ;
    for each $\mathcal{H}_{ijk} \neq 0$ do
        Update the $U_{i*}$, $I_{j*}$, and $C_{k*}$ by Eq. (4) ;
        Compute the objective function $F^L$ by Eq. (3) ;
    end
    Compute training loss $tl_l$ in $l^{th}$ iteration ;
    Compute change rate $cRate = (tl_{l-1} - tl_l)/tl_{l-1} * 100$ ;
    if $cRate < tol$ then
        break;
    end
    $l = l + 1$;
end
Return $\hat{\mathcal{H}} = \mathcal{S} \times_U U \times_R R \times_C C$ ;
```

and the dimensionalities I, J, and K of user, item and criterion factors. Therefore, a time complexity of the proposed algorithm is $\mathcal{O}(LRIJK)$. It is worth to mention that tensor factorization of our models are faster than conventional ones because the R and K are constants and the iteration number is less than $L = 10$. Therefore, the final complexity of our approach thus is $\mathcal{O}(FIJ)$ with constant $F \leqq 10 \times RK$ actually.

4. Evaluation protocol and metrics

4.1. Dataset and data analysis

The dataset used in our experiments contains 44,217 multiple ratings of 19,970 users to 2,484 restaurants gathered from Tripadvisor. Note that we used all data in the dataset without any filtering in order to consider new users with partial preferences. Since this study considers four rating criteria, sparsities of the proposed models is higher than other compared methods based on a user-item matrix. For example, if the tensor model consists of the users, restaurants, and multiple ratings, its sparsity is very high, as 99.978% (density is 0.022%). According to [26], this sparsity is natural in real-world situations but hampers the accuracy of recommendation systems.

Indeed, let's look at the dataset in details to briefly discuss how many unknown ratings are collected in terms of MCRSs. Table 1 presents the distribution of multiple ratings in the dataset. The percentage expression in brackets indicates a ratio of rating number divided by the total number, and the ratios in the last row are between the total numbers of the overall type and the other rating types. Note that we also found that there are some users, who input only other ratings without overall one (i.e., the second pattern), but they are very few (around 89) in the dataset. Therefore, we can glance at the problem of new users with partial preferences in the dataset via ratios and distributions showed in the table above. Brief but meaningful analysis results are as follows:

Table 1. Rating distribution for multi-criteria

Rating	Overall (%)	Food (%)	Price (%)	Service (%)
1	1,027 (5.96)	485 (5.49)	640 (7.09)	641 (7.02)
2	1,144 (6.64)	532 (6.02)	700 (7.76)	505 (5.53)
3	2,122 (12.32)	1,113 (12.59)	1,597 (17.70)	1,103 (12.08)
4	5,287 (30.70)	2,562 (28.98)	2,876 (31.87)	2,436 (26.68)
5	7,642 (44.37)	4,148 (46.92)	3,210 (35.58)	4,446 (48.69)
Total	17,222	8,840 (51.33)	9,023 (52.39)	9,131 (53.02)

- In terms of rating types, the overall rating is usually filled by most of the users, while half of the other types have the unknown values. In the context of MCRSs, it emphasizes why the unknown class patterns caused by users' unfamiliarity need to be addressed. It will be discussed more in Section 5.2.
- There are much more positive ratings (around 75 %) than negative ones, and it is important to analyze the effects of proposed rules, especially $\mathcal{R}_1$ and $\mathcal{R}_3$. In other words, the effectiveness of $\mathcal{R}_2$ could be relatively decreased because of the small number of low ratings.

On the other hand, if a multi-dimensional model such as matrix or tensor maintains inherent relations between multiple types of ratings, correlations between them are also significant for MCRSs. Table 2 shows the correlations between rating types in two datasets. One consists of fully filled data only, and the other fills empty values by zero as all data to compute correlations. For the fully filled data, each criterion shows high correlations with the others. In particular, all other criteria have higher correlations with overall criterion than 0.79. However, when we consider all data without any complementary ways, the correlations are decreased as around 0.2. It will significantly and negatively affect the performance of recommender systems based on the multi-dimensional models. Therefore, we need to carefully deal with new users with partial preferences in such systems. One other interesting is that the correlations between other rating types except for overall type are higher than 0.92, in the case of all data. It is because of that half of the other rating

Table 2. Correlation between multi-criteria

	Fullly filled data (8,612)				All data (17,222)			
	Overall	Food	Price	Service	Overall	Food	Price	Service
Overall	1.000	0.873	0.839	0.794	1.000	0.197	0.207	0.181
Food	0.873	1.000	0.813	0.687	0.197	1.000	0.941	0.926
Price	0.839	0.813	1.000	0.693	0.207	0.941	1.000	0.938
Service	0.794	0.687	0.693	1.000	0.181	0.926	0.938	1.000

types are empty and filled by zero. It emphasizes that a model must distinguish between unknown and lowest ratings to avoid this kind of bias. In this paper, we only used known ratings and unknown ratings complemented by the proposed rules to train models.

4.2. Experimental protocol

To compare prediction performances of the proposed method and other techniques, three measures (i.e., Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE)) are exploited as follows:

$$RMSE = \sqrt{\sum\nolimits_{i=1}^{N} (\hat{r}_i - r_i)^2/N}, \tag{5}$$

$$MSE = \sum\nolimits_{i=1}^{N} (\hat{r}_i - r_i)^2/N, and \tag{6}$$

$$MAE = \sum\nolimits_{i=1}^{N} |\hat{r}_i - r_i|/N, \tag{7}$$

where $\hat{a}_i$ and a_i are predicted and observed rating scores, respectively. The N indicates the number of compared ratings.

We use k-fold cross-validation scheme to compare errors between predicted and observed ratings with avoiding overfitting impacts in all the following experiments. In this regard, it is significant to select a proper value for k since a poorly chosen k value could cause a misrepresentation (e.g., overestimation or high variance). According to "Typically, $k = 5$ or $k = 10$ have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor very high variance [16]," we set the k as 5.

4.3. Baseline Models

This section explains other baselines. The two-dimensional techniques using user-item matrix and basic MCRSs based on the techniques are as follows:

K-Nearest Neighbors-based CF (KNN): is one of the basic CF algorithms. The predicted value $\hat{r}_{ui}$ of user u to item i is defined by $\hat{r}_{ui} = \sum_{v\in\sum_i^n(u)} sim(u,v) \cdot r_{vi}/\sum_{v\in\sum_i^k(u)} sim(u,v)$, where n is the number of neighbors and $sim()$ denotes a similarity function.

KNN BaseLine-based CF (KNN-BL): inspired by [15] models neighborhood relations by minimizing a global cost function. The prediction $\hat{r}_{ui}$ is defined as $\hat{r}_{ui} =$

$b_{u,i}+\sum_{v\in\sum_{N_i^k(u)}} sim(u,v)\cdot(r_{vi}-b_{vi})/\sum_{v\in\sum_{N_i^k(u)}} sim(u,v)$, where $b_{u,i}=\mu+b_u+b_i$ indicates baseline estimates with an overall average rating μ and observed deviations b_u and b_i of users and items. The n and $sim()$ are equal to those of KNN.

Co-clustering-based CF (COC) [8]: assigns users and items into some user and item clusters. Rating scores of a target user are then predicted based on ratings of users and items belonging to the same cluster of the target user. An approximate rating $\hat{r}_{ui}$ is computed as follows: $\hat{r}_{ui} = \overline{C_{ui}} + (\eta_u - \overline{C_u}) + (\eta_i - \overline{C_i})$, where the $\overline{C_{ui}}$, $\overline{C_u}$ and $\overline{C_i}$ are average ratings of co-cluster C_{ui}, u's cluster and i's cluster, respectively.

Singular Value Decomposition-based MF (SVD): has been popularized by Simon Funk during the Netflix Prize. The prediction $\hat{r}_{ui}$ is set as: $\hat{r}_{ui} = \eta+b_u+b_i+q_i^T p_u$. If user u or item i is unknown, then the biases b_u or b_i and the factors p_u or q_i are assumed to be zero. To estimate all unknown values, a loss function is usually minimized by stochastic gradient descent.

SVD++-based MF (SVD++) [14]: is an extension of the SVD considering implicit ratings. The prediction $\hat{r}_{ui}$ is calculated by $\hat{r}_{ui} = \eta+b_u+b_i+q_i^T*(p_u+|I_u|^{-1/2}\sum_{j\in I_u} y_j)$, where the y_j term indicates a new set of item factors capturing implicit ratings. Also, the implicit rating means that a user u rated an item j, regardless of the rating value.

Non-negative Matrix Factorization (NMF) [20]: is similar to SVD. The predicted rating $\hat{r}_{ui}$ is set as: $\hat{r}_{ui} = q_i^T p_u$, where user and item factors keep positive. For optimization, a (regularized) stochastic gradient descent is used.

Aggregation-based Multi-criteria Recommendation (AMR): On the other hand, we also compare our method with conventional MCRS methods that has been proposed in [1]. Among the MCRS techniques, we implemented aggregation-function-based approach based on the above-mentioned two-dimensional techniques. Such approaches have three steps. The first stage predicts missing rating values via a single-criterion method based on the user-item matrix for each criterion. Second stage aims to estimate relationships between the overall rating and other criteria, so we used linear, Ridge, and Lasso regressions in order to get best coefficients between overall rating with the others (i.e., food, price, and service ratings). Lastly, the approximated values are aggregated into overall ratings with the coefficients obtained as weights in this aggregation stage.

We implemented the two-dimensional recommendation techniques by using surprise library and conducted a grid-search to find optimal parameters. The AMR was developed by using the techniques and regression methods of Scikit-learn library. All the experiments including parallel parameter searches conducted in the same computation environment consisting of 40 CPUs and RAM 128GB.

5. Evaluation and discussion

This section compares the proposed Hypothetical Tensor Model (HTM) with other baseline methods and discusses effectiveness of the proposed rules to alleviate the problem of new users with partial preferences in the context of MCRSs.

5.1. Performance comparison

Figure 3 shows the RMSE and MAE of two-dimensional techniques and HTM. The compared methods have much lower performances than HTM in terms of both measures. The

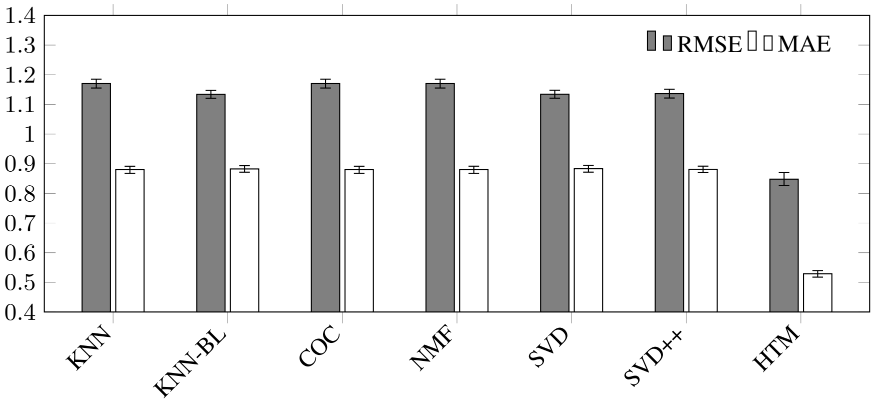


Fig. 3. Performance comparison with two dimensional-based techniques

two-dimensional methods for the dataset including new users with partial ratings are average RMSE 1.1525 and MAE 0.8810. It means that they are negatively affected by the new users as one of cold-start problems. However, the proposed HTM outperform them, with average RMSE 0.8480 and MAE 0.5286. These results support that our proposed model can improve the recommendation performance and alleviate the new user problem.

The HTM is also compared with basic MCRSs (i.e., AMRs). Figure 4 presents performances of the various AMRs with different two-dimensional techniques and HTM. Experimental results show that the proposed HTM significantly outperforms than the AMRs. Because the AMRs are based on the two-dimensional methods, and such methods show low performances on the dataset including new users, the performance of AMRs are also affected negatively. Even the AMRs with SVD and SVD++ show bad performances than two-dimensional methods SVD and SVD++. It emphasizes that proper handling of new users with partial preferences is significantly important to MCRSs.

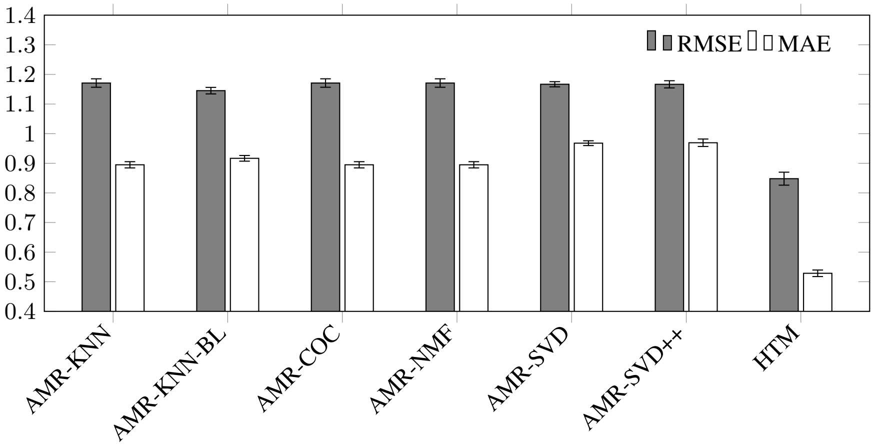


Fig. 4. Performance comparison with basic MCRS techniques

To more discussion, we selected KNN-BL, COC, SVDpp, and ARMs based on them, because they show better performance than the other two-dimensional techniques. Table 3 lists performances of all the methods and MSE ratios of HTM to the other techniques. We use the MSE ratio to see how much is a better performance of the proposed method than baseline methods. The ratio is defined by $(1 - MSE_{targ}/MSE_{comp}) \times 100$, where the MSE_{targ} and MSE_{comp} indicate MSE errors of target and compared methods. Note that all the techniques were tested with the same number 3,344 of test samples with 5-fold cross-validation to avoid some bias which can happen by different sizes of test set. If the ratio is a positive value 50, it means that a target method is better as more 50 percentage than a compared one. As a result, the HTM improves the minimum 40% (maximum 47%) than the baseline methods in terms of the MSE measure, see the last row. In general, the more deviations between RMSE and MAE, the more substantial error variance. It means that errors of predicted ratings are unstable. In this regard, HTM shows slightly higher standard deviations of MSE, RMSE and MAE than the ARMs, while it has similar ones with two-dimensional methods. Therefore, HTM's instability on the preference prediction is acceptable.

Table 3. Comparison of HTM with other techniques

Measure		two-dimensional techniques			ARMs			HTM
		KNN-BL	COC	SVD++	KNN-BL	COC	SVD++	
Mean	MSE	1.2855	1.3693	1.2912	1.3114	1.3711	1.3615	**0.7192**
	RMSE	1.1338	1.1702	1.1363	1.1452	1.1709	1.1668	**0.8480**
	MAE	0.8824	0.8798	0.8809	0.9167	0.8949	0.9679	**0.5286**
Std	MSE	0.0302	0.0348	0.0333	0.0252	0.0333	**0.0203**	0.0372
	RMSE	0.0133	0.0149	0.0147	0.0110	0.0143	**0.0087**	0.0219
	MAE	0.0108	0.0119	0.0110	0.0096	0.0105	**0.0081**	0.0110
MSE comparison (%)		44.06	47.48	44.30	45.16	47.55	47.18	

5.2. Effectiveness of unknown class rules

This section debates the efficacy of unknown class rules. Table 4 lists performances of ARMs with the above three two-dimensional techniques and the proposed rules. The notation RARM indicates ARM based on the proposed rules. As a result, RARMs based on KNN-BL and SVD++ have better performance than the ARMs, except for COC base.

Table 4. Effectiveness comparison of AMRs by proposed rules

Measure		KNN-BL		COC		SVD++	
		RARM	ARM	RARM	ARM	RARM	ARM
Mean	RMSE	1.1374	1.1452	1.1814	1.1709	1.1429	1.1666
	MAE	0.8729	0.9167	0.8980	0.8949	0.8675	0.9693
Std	RMSE	0.0152	0.0110	0.0174	0.0143	0.0181	0.0121
	MAE	0.0113	0.0096	0.0122	0.0105	0.0122	0.0126

Moreover, the two methods show slightly low RMSE and MAE errors than two-dimensional techniques (see KNN-BL and SVD++ in Table 3) despite the standard deviation of them is similar. Thus, experimental results support that there is the potential effectiveness for the basic MCRSs. However, there were no high improvements, since it might be difficult to effectively keep inherent relationships between multi-criteria via such methods based two-dimensional matrix.

Table 5 shows performances of the HTM for each multi-criterion and all ratings by the proposed rules. Bold and underline font styles represent first and second-best performances, respectively. The CTF denotes a conventional tensor factorization that isn't applied by the proposed rules. Performances of the HTM with all rules is marked by italic font style. To fairly compare between model performances, optimal parameters for CTF and HTMs were equally set as regularization parameter $\lambda = 0.01$, learning rate $t_o = 0.001$, latent factors '3-2-2' of users, items, and rating types.

Table 5. Effectiveness comparison of HTM for multi-criteria

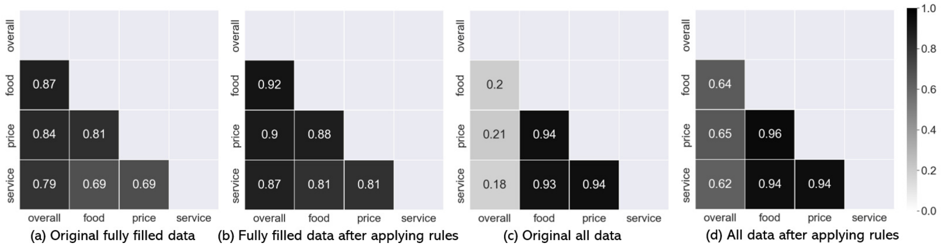
		Mean					Std				
		CTF	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	HTM	CTF	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	HTM
All	MSE	1.0412	**0.6376**	$\underline{1.0355}$	1.0368	*0.6158*	0.0251	**0.0063**	0.0270	$\underline{0.0155}$	*0.0142*
	RMSE	1.0203	**0.7985**	$\underline{1.0175}$	1.0182	*0.7847*	0.0123	**0.0039**	0.0132	$\underline{0.0076}$	*0.0091*
	MAE	0.6790	**0.5307**	$\underline{0.6763}$	0.6772	*0.5221*	0.0074	**0.0048**	0.0070	$\underline{0.0048}$	*0.0055*
Overall	MSE	1.8015	**0.7370**	$\underline{1.7979}$	1.8181	*0.7196*	0.0449	**0.0129**	0.0487	$\underline{0.0442}$	*0.0372*
	RMSE	1.3421	**0.8585**	$\underline{1.3407}$	1.3483	*0.8480*	0.0168	**0.0075**	0.0181	$\underline{0.0163}$	*0.0219*
	MAE	0.9237	**0.5403**	$\underline{0.9205}$	0.9248	*0.5286*	$\underline{0.0082}$	**0.0054**	0.0133	0.0112	*0.0110*
Food	MSE	0.4561	0.4723	$\underline{0.4493}$	**0.4396**	*0.4453*	0.0246	0.0279	**0.0172**	$\underline{0.0234}$	*0.0265*
	RMSE	0.6751	0.6869	$\underline{0.6702}$	**0.6628**	*0.6670*	0.0181	0.0204	**0.0128**	$\underline{0.0180}$	*0.0199*
	MAE	0.4535	0.4667	$\underline{0.4511}$	**0.4472**	*0.4576*	0.0111	$\underline{0.0104}$	0.0115	**0.0089**	*0.0104*
Price	MSE	0.4994	0.5148	$\underline{0.4948}$	**0.4825**	*0.4939*	$\underline{0.0162}$	0.0290	0.0187	**0.0120**	*0.0044*
	RMSE	0.7066	0.7172	$\underline{0.7033}$	**0.6946**	*0.7028*	$\underline{0.0115}$	0.0205	0.0133	**0.0086**	*0.0032*
	MAE	0.5421	**0.5279**	0.5420	$\underline{0.5371}$	*0.5247*	**0.0035**	0.0104	0.0062	$\underline{0.0062}$	*0.0030*
Service	MSE	0.6945	0.7129	$\underline{0.6884}$	**0.6834**	*0.6876*	**0.0242**	0.0451	0.0321	$\underline{0.0297}$	*0.0332*
	RMSE	0.8333	0.8439	$\underline{0.8295}$	**0.8265**	*0.8290*	**0.0145**	0.0269	0.0193	$\underline{0.0180}$	*0.0199*
	MAE	0.5648	0.5751	**0.5600**	$\underline{0.5623}$	*0.5664*	**0.0101**	0.0157	0.0145	$\underline{0.0116}$	*0.0106*

In terms of rating types (i.e., each criterion), $\mathcal{R}_1$ positively affects the "Overall" rating, and $\mathcal{R}_2$ and $\mathcal{R}_3$ show their effectiveness to other extra rating types (i.e., "Food", "Price", and "Service"). However, the magnitudes of their impacts differ. As we glanced in Table 1, it is because of the different numbers of data that can be applied by each rule. Indeed, rating numbers of data applied by $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$ were $4,345$, $1,582$, and $2,371$. In this regard, three rules have acceptable efficiencies to improve the performance of HTM.

It is worthy to mention that although the CTF for dataset including new users had worse performance than two-dimensional methods (see only results from "Overall" rating to compare fairly), other extra rating types showed improved performances by comparing with the "Overall" rating. As a result, performances of CTF for all data show improvements than the two-dimensional techniques (see the row "All") in terms of both mean and standard deviation of MSE, RMSE, and MAE. It means that representation via tensor models can improve stably performance because of their characteristics maintaining an inherent structure of and relationships between multi-criteria.

On the other hand, our HTM outperform than the other techniques including CTF in terms of “Overall” rating types. Furthermore, the standard deviations of RMSE and MAE of HTM for most of the rating types (except for “Service”) are similar to or smaller than the CTF. Thus, HTM has better stability to predict user preferences than conventional tensor factorization methods. Furthermore, HTM show 40.86% performance improvement from the CTF by MSE comparison for “All” ratings. As a result, experimental results verified that handling of the problem new users in MCRSs is one of significant tasks, and the HTM solves the problem well.

Lastly, we discuss correlations between multi-criteria by the proposed rules ($\mathcal{RU}()$). As afore-discussed, there are two kinds of dataset (i.e., fully filled data and all data) to compare correlations. Figure 5 shows the correlation heat-maps between multi-criteria of original data and data complemented by the proposed rules. For the ‘fully filled data’ of (a) and (b) in the figure, $\mathcal{RU}()$ increases correlations between multiple ratings. Even, for the ‘all data’ including new users, overall rating’s correlations with the other ratings are increased than original all data (see (c) and (d) in the figure above). It would be the reason why our HTM outperforms the CTF.



(a) Original fully filled data (b) Fully filled data after applying rules (c) Original all data (d) All data after applying rules

Fig. 5. Correlation between multi-criteria with rules

6. Conclusion

MCRSs have been developed to improve the accuracy of traditional single-criterion recommender systems that cannot express and reflect fine-grained users’ rating behaviors. However, in some MCRSs, new users would be asked to express their preferences on some criteria of items in order to address the cold-start problem. Such collected preferences are often incomplete because of the users’ unfamiliarity on the rating scheme and recommendation domain, which are called ‘partial preferences’. The issue of new users as one of the cold-start problems thus decrease the accuracy performance of MCRSs.

To address the problem in the context of MCRSs, we found four patterns of partial preferences that are caused by new users’ unfamiliarity via data analysis. And then, an intuitive rule set based on relationships between multi-criteria is defined to alleviate the negative impact of partial preferences. As a result, the Hypothetical Tensor Model (HTM) based on the rules is proposed to maintain a structure of and correlations between multi-criteria and to improve the performance of MCRSs. Experiments on a TripAdvisor dataset showed the better performances of the HTM than baseline methods as well as the effectiveness of the proposed rules for each criterion.

Since there are still some limitations in the proposed model, we plan two future pieces of research. One relates to $\mathcal{R}_1$ which is currently applied to the minimum or maximum rating. As the rule can be applied many data than the other rules, we will find an appropriate machine learning technique to complete other values of ratings. The other is relevant to the high computational cost of the proposed method. Even though, as afore-discussed in Section 3.3, some parts of computational complexity are constant, but the cost is still high. Therefore, we will leverage a clustering method to reduce the computational cost.

Acknowledgement. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2020R1A2B5B01002207).

References

1. Adomavicius, G., Kwon, Y.: New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems 22(3), 48–55 (2007)
2. Al-Ghuribi, S.M., Noah, S.A.M.: Multi-criteria review-based recommender system-the state of the art. IEEE Access 7, 169446–169468 (2019)
3. Borràs, J., Moreno, A., Valls, A.: Intelligent tourism recommender systems: A survey. Expert Systems with Applications 41(16), 7370–7389 (2014)
4. Ebadi, A., Krzyzak, A.: A hybrid multi-criteria hotel recommender system using explicit and implicit feedbacks. International Journal of Computer and Information Engineering 10(8), 1377–1385 (2016)
5. Farokhi, N., Vahid, M., Nilashi, M., Ibrahim, O.: A multi-criteria recommender system for tourism using fuzzy approach. Journal of Soft Computing and Decision Support Systems 3(4), 19–29 (2016)
6. Fu, Y., Liu, B., Ge, Y., Yao, Z., Xiong, H.: User preference learning with multiple information fusion for restaurant recommendation. In: Zaki, M.J., Obradovic, Z., Tan, P., Banerjee, A., Kamath, C., Parthasarathy, S. (eds.) In Proceedings of the 2014 SIAM International Conference on Data Mining. pp. 470–478. SIAM, Philadelphia, Pennsylvania, USA (Apr 2014)
7. Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G.E.: Mobile recommender systems in tourism. Journal of Network and Computer Applications 39, 319–333 (2014)
8. George, T., Merugu, S.: A scalable collaborative filtering framework based on co-clustering. In: In Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005). pp. 625–628. IEEE Computer Society, Houston, Texas, USA (Nov 2005)
9. Hassan, M., Hamada, M.: A neural networks approach for improving the accuracy of multi-criteria recommender systems. Applied Sciences 7(9), 868 (2017)
10. Hassan, M., Hamada, M.: Genetic algorithm approaches for improving prediction accuracy of multi-criteria recommender systems. International Journal of Computational Intelligence Systems 11(1), 146–162 (2018)
11. Hong, M., Akerkar, R., Jung, J.J.: Improving explainability of recommendation system by multi-sided tensor factorization. Cybernetics and Systems 50(2), 97–117 (2019)
12. Hong, M., Jung, J.J.: Multi-sided recommendation based on social tensor factorization. Information Sciences 447, 140–156 (2018)
13. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Amatriain, X., Torrens, M., Resnick, P., Zanker, M. (eds.) In Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010. pp. 79–86. ACM, Barcelona, Spain (Sep 2010)

14. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Li, Y., Liu, B., Sarawagi, S. (eds.) In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 426–434. ACM, Las Vegas, Nevada, USA (Aug 2008)
15. Koren, Y.: Factor in the neighbors: Scalable and accurate collaborative filtering. ACM Transactions on Knowledge Discovery from Data 4(1), 1:1–1:24 (2010)
16. Kuhn, M., Johnson, K.: Applied predictive modeling, vol. 26. Springer (2013)
17. Kumar, G.: A survey on multi criteria decision making recommendation system using sentiment analysis. International Journal of Applied Engineering Research 13(15), 11724–11729 (2018)
18. Lakiotaki, K., Matsatsinis, N.F., Tsoukiàs, A.: Multicriteria user modeling in recommender systems. IEEE Intelligent Systems 26(2), 64–76 (2011)
19. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: A survey. Decision Support Systems 74, 12–32 (2015)
20. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Transactions on Industrial Informatics 10(2), 1273–1284 (2014)
21. Musto, C., de Gemmis, M., Semeraro, G., Lops, P.: A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In: Cremonesi, P., Ricci, F., Berkovsky, S., Tuzhilin, A. (eds.) In Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017. pp. 321–325. ACM, Como, Italy (Aug 2017)
22. Nassar, N., Jafar, A., Rahhal, Y.: A novel deep multi-criteria collaborative filtering model for recommendation system. Knowledge-Based Systems 187 (2020)
23. Pappas, N., Popescu-Belis, A.: Adaptive sentiment-aware one-class collaborative filtering. Expert Systems with Applications 43, 23–41 (2016)
24. Rendle, S., Marinho, L.B., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 727–736. ACM, Paris, France (Jun 2009)
25. Sahoo, N., Krishnan, R., Duncan, G.T., Callan, J.: Research note - the halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! movies. Information Systems Research 23(1), 231–246 (2012)
26. Singh, M.: Scalability and sparsity issues in recommender datasets: a survey. Knowledge and Information Systems pp. 1–43 (2018)
27. Wasid, M., Ali, R.: An improved recommender system based on multi-criteria clustering approach. Procedia Computer Science 131, 93–101 (2018)
28. Zhang, S., Salehan, M., Leung, A., Cabral, I., Aghakhani, N.: A recommender system for cultural restaurants based on review factors and review sentiment. In: In Proceedings of the 24th Americas Conference on Information Systems, AMCIS 2018. New Orleans, LA, USA (Aug 2018)
29. Zheng, X., Luo, Y., Sun, L., Zhang, J., Chen, F.: A tourism destination recommender system using users' sentiment and temporal dynamics. Journal of Intelligent Information Systems 51(3), 557–578 (2018)
30. Zheng, Y.: Criteria chains: A novel multi-criteria recommendation approach. In: Papadopoulos, G.A., Kuflik, T., Chen, F., Duarte, C., Fu, W. (eds.) In Proceedings of the 22nd International Conference on Intelligent User Interfaces, IUI 2017. pp. 29–33. ACM, Limassol, Cyprus (Mar 2017)
31. Zhongqin, B., Shuming, D., Zhe, L., Yongbin, L.: A recommendations model with multiaspect awareness and hierarchical user-product attention mechanisms. Computer Science and Information Systems 17(3), 849-865 (2020)

Minsung Hong as a postdoctoral researcher in Western Norway Research Institue, Norway, since February 2018 is participating in several EU Horizon2020 and Norway national projects. He received the B.S. and M.S. degrees in Computer Engineering from DanKook University in 2011 and 2014, respectively. He received the Ph.D. degree in Computer Engineering from Chung-Ang University in 2018. His research topics are recommender systems based on big data by using various the methodologies of artificial intelligence, data mining, and machine learning.

Jason J. Jung is a Full Professor in Chung-Ang University, Korea, since September 2014. Before joining CAU, he was an Assistant Professor in Yeungnam University, Korea since 2007. Also, he was a postdoctoral researcher in INRIA Rhone-Alpes, France in 2006, and a visiting scientist in Fraunhofer Institute (FIRST) in Berlin, Germany in 2004. He received the B.Eng. in Computer Science and Mechanical Engineering from Inha University in 1999. He received M.S. and Ph.D. degrees in Computer and Information Engineering from Inha University in 2002 and 2005, respectively. His research topics are knowledge engineering on social networks by using many types of AI methodologies, e.g., data mining, machine learning, and logical reasoning. Recently, he have been working on intelligent schemes to understand various social dynamics in large scale social media.

Received: May 31, 2020; Accepted: November 12, 2020.

Computer Science and Information Systems 18(1):303–321 https://doi.org/10.2298/CSIS201109059Z

Metaphor Research in the 21st Century: A Bibliographic Analysis

Dongyu Zhang[1], Minghao Zhang[1], Ciyuan Peng[2], Jason J. Jung[2], and Feng Xia[3]*

[1] School of Software, Dalian University of Technology, Dalian 116620, China
zhangdongyu@dlut.edu.cn, zhang.minghao@outlook.com
[2] Department of Computer Engineering, Chung-Ang University, Seoul 156-756, Korea
sayeon1995@gmail.com, j2jung@gmail.com
[3] School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat 3353, Australia
f.xia@ieee.org

Abstract. Metaphor is widely used in human communication. The cohort of scholars studying metaphor in various fields is continuously growing, but very few work has been done in bibliographical analysis of metaphor research. This paper examines the advancements in metaphor research from 2000 to 2017. Using data retrieved from Microsoft Academic Graph and Web of Science, this paper makes a macro analysis of metaphor research, and expounds the underlying patterns of its development. Taking into consideration sub-fields of metaphor research, the internal analysis of metaphor research is carried out from a micro perspective to reveal the evolution of research topics and the inherent relationships among them. This paper provides novel insights into the current state of the art of metaphor research as well as future trends in this field, which may spark new research interests in metaphor from both linguistic and interdisciplinary perspectives.

Keywords: metaphor, literature analysis, statistical analysis, scholarly big data.

1. Introduction

Metaphor is an indispensable part of human communication. According to empirical studies, every three sentences in natural language uses a metaphor [38,43]. It is not only a universal linguistic phenomenon but also a means for people to understand and cognize [29]. Humans frequently use one concept in metaphors to describe another concept for reasoning. For instance, in the metaphorical utterance: ‘experience is a treasure,’ we use ‘treasure’ to describe ‘experience’ to emphasize that ‘experience’ can be valuable. A metaphor has been viewed as a mapping system that conceptualizes one domain (target) in terms of another (source) [29]. In particular, along with the rapid explosion of social media applications such as Facebook and Twitter, metaphorical texts and information have increased dramatically. It seems to be very common for Internet users to use vivid and colorful metaphorical expressions on social media on a variety of topics including, products, services, public events, tidbits of their life, etc.

An increasing number of researchers have studied metaphor from different perspectives in fields like linguistics [47,41,37,52], psychology [35,27,33,20], neuroscience [1,22,15,17],

* Corresponding author

management [54,39,13,2], and computer science [48,36,55,19]. Since metaphor research has been developing dramatically, it is necessary to review the current situation, the development and trends of metaphor research, as well as studying how metaphor research has evolved through time. This may make contributions to some novel and interesting studies of metaphors from both linguistic and interdisciplinary perspectives as well as exploring the related underlying mechanism. Previous studies have shown that quantitative analysis can explain the nature of a particular discipline or field and changes in research focus over time [34,8]. Researchers can use some information platforms, such as AMiner [50], Google Scholar [10], Microsoft Academic Services [51], and many other scientific online systems [49]. These information platforms contain useful data, including but not limited to authors, papers, and references, and they can carry out statistical analysis. So far, based on the above academic systems, a large number of related works have applied quantitative analysis techniques in scientometrics. [25] used bibliographic analysis to summarize human interactions. [24] reviewed various studies using online social networks to identify personality, as reported in the literature. [8] made a quantitative assessment of mapping the intellectual structure and development of computer-supported cooperative work. [31] used complex network topology to study the evolution of artificial intelligence. Also, [45] made contributions to the research in the field of transportation.

Numerous theories and technologies of literature analysis based on big scholarly data have been proposed [53,32,56,26]. However, so far, few people have collected bibliometrics data to analyze metaphor quantitatively and to comprehend its internal structure as well as evolution. To fill this gap, in this paper, we carry out a bibliometric analysis of the development of metaphor research in the early 21st century, based on the following four aspects. First, we analyze the development of metaphor research by counting the increment of the number of publications over time. Second, we emphasize influence and citation patterns to distinguish the behavioral dynamics of citation. Third, we try to quantify milestones during this period through identification of the characteristics of influential papers, researchers, and institutions. Finally, we explore the internal structure of metaphor research by analyzing the evolution of themes and mutual attraction.

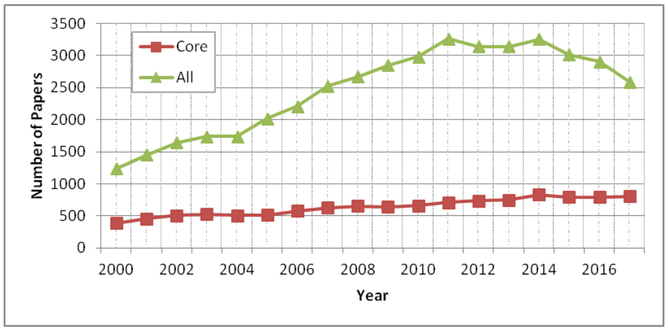


Fig. 1. Changes in the number of papers in Metaphor (every year) since 2000.

The scholarly dataset we use in our study consists of 11,564 papers from the Web of Science and 44,586 papers from Microsoft Academic Graph (MAG). The rest of the paper

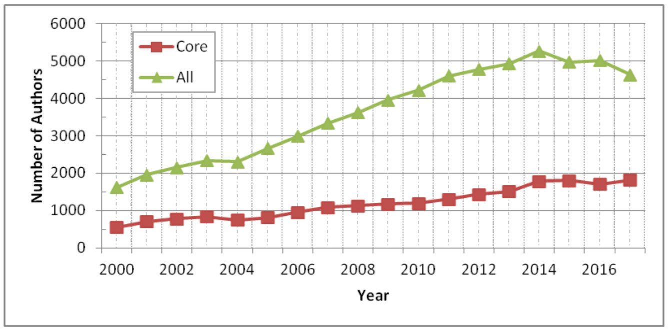


Fig. 2. The number of authors every years.

is organized as follows. Section 2 provides the methodologies and models we use for our analysis. Section 3 introduces the experimental results we obtained from our literature analysis. Section 4 concludes the paper and provides some directions for the future.

2. Methods

In this section, we first introduce the data set we use to analyze metaphors: core data sets and extended data sets. Then we introduce several indicators for measuring the importance of authors and institutions in the field of metaphor research and their calculation methods. Finally, we introduce the division of the field of metaphor research.

2.1. Datasets

For conducting experiment, we employ MAG data set (`http://research.microsoft.com/en-us/projects/mag/`)—a widely used and one of the best databases for empirical research in scientometrics and citation analysis [42,51]. Hence, to investigate the current state of metaphor research, we extract the papers from the MAG data set, which contains six entities: affiliations, authors, conferences, fields of study, journals, and papers. The new MAG data set contains new relationships in the field of study with papers. First, we limit the publication time of the articles to 2000 and beyond. Then, from these papers, we select articles that comprise at least one of the following six words in their title or abstract: metaphor, metaphorical, metaphorically, Metaphor, Metaphorical, or Metaphorically. We use all the extracted papers as our extended data set containing 44,586 articles, of which 1,872 are conference papers. Because the number of conference papers was inadequate, we do not consider its particularity, and we do not give it any special treatment.

Additionally, we found all the journals related to metaphors from the Web of Science database (`http://isiknowledge.com`), of which there are about a thousand. We extract these journals as a list of our core journals. Then, based on the list of core journals, we extract the articles published in the core journals from the papers of the extended data set as our core data set. It contains 11,564 articles.

We use the same statistics and calculations for both the core data set and the extended data set.

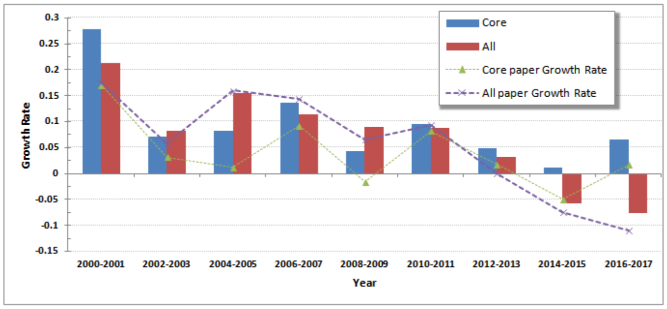


Fig. 3. The growth rate of authors as well as total publications every two years.

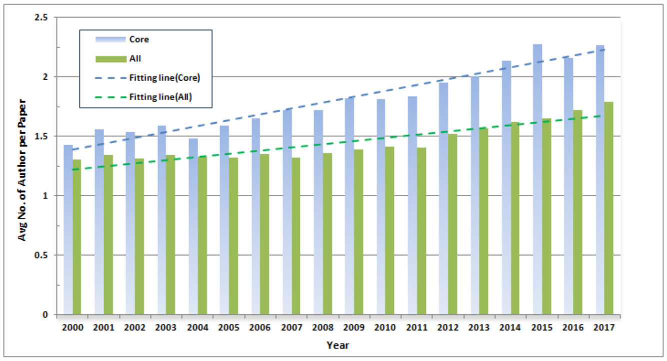


Fig. 4. The average number of authors per paper.

2.2. Indicators and calculation methods

We consider the following indicators to assess the relevance of authors as well as publications in this field.

- `Measuring research output by measurement`: We assume that the core data set or the extended data set is P, and we use statistical methods to calculate the total number of articles in the data set denoted as $|P|$, total number of authors $\sum_{p \in P} |A_p|$, total number of citations $\sum_{p \in P} |Ci_p|$, and total number of references $\sum_{p \in P} |R_p|$. We then calculate the average number of authors per paper $\frac{\sum_{p \in P} |A_p|}{|P|}$, the average number of citations per paper $\frac{\sum_{p \in P} |Ci_p|}{|P|}$, the average number of references per paper $\frac{\sum_{p \in P} |R_p|}{|P|}$, and the average number of papers per author $\frac{|P|}{\sum_{p \in P} |A_p|}$ ($|A_p|$ represents the total number of authors of the paper, $|Ci_p|$ represents the total number of citations of the paper, and $|R_p|$ represents the total number of references of the paper).
- `self-citation rate`: In addition, to reflect the dynamics of the researcher's reference behavior, we use the most rigorous self-guided definition as our evaluation height, that is, if both referenced papers have at least one mutual author, then there are

Table 1. Ranking of papers based on the total number of citations received in2000-2017 in core dataset papers.

No.	Title	Citations	Published Year
1	Knowledge and organization: A social-practice perspective[5]	2,044	2001
2	The network structure of social capital[6]	1,922	2000
3	Adaptive subgradient methods for online learning and stochastic optimization[12]	1,609	2011
4	From metaphor to measurement: Resilience of what to what?[7]	1,348	2001
5	Community resilience as a metaphor, theory, set of capacities, and strategy for disaster readiness[35]	1,279	2008
6	Social and psychological resources and adaptation[23]	1,234	2002
7	Relational frame theory: A post-Skinnerian account of human language and cognition[21]	1,185	2001
8	Modern social imaginaries[46]	1,150	2002
9	Where mathematics comes from: How the embodied mind brings mathematics into being[30]	993	2000
10	The evolution of foresight: What is mental time travel and is it unique to humans?[44]	800	2007
11	A thorough benchmark of density functional methods for general main group thermochemistry, kinetics, and noncovalent interactions[16]	724	2011
12	Self-control relies on glucose as a limited energy source: Willpower is more than a metaphor[14]	723	2007
13	Scale-free networks provide a unifying framework for the emergence of cooperation[40]	696	2005
14	The surveillant assemblage[18]	692	2000
15	Metaphoric structuring: Understanding time through spatial metaphors[4]	685	2000

two references between these references. The paper is self-cited by the author. It can be computed as $\frac{\sum_{r \in R} |A_r|}{|R|}$, where $|R|$ is the total number of references of the paper and $|A_r|$ is the number of author self-citation. Similarly, a self-journal (conference) is when the paper and one or more of its references are published in the same journal (conference). This can be computed as $\frac{\sum_{r \in R} |J_r|}{|J_r|}$, where $|R|$ is the total number of references in the paper and $|J_r|$ is the number of journal (conference) self-citations. Self-affiliation is when the paper and one or more of its references come from same affiliation. This can be computed as $\frac{\sum_{r \in R} |Aff_r|}{|R|}$, where $|R|$ is the total number of references in the paper and $|J_r|$ is the number of journal (conference) self-citations.

2.3. The inner structure of metaphor

- `Topic exploration`: The study of metaphor is not an independent discipline, but an interdisciplinary science. The MAG data set constructs the domain into a forest of six-layered tree structures. The new MAG data set also contains new relationships

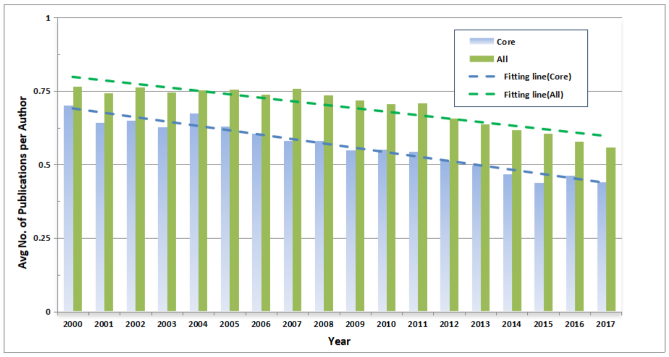


Fig. 5. The average productivity of Metaphor scientists.

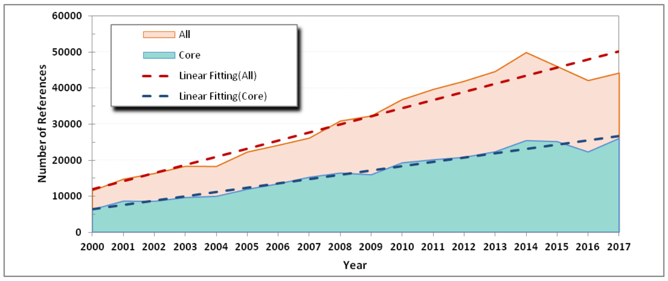


Fig. 6. Changes in references.

in the field of study. Therefore, we can easily divide the topic of metaphor research. In the end, we choose the root node of each tree as the topic of metaphor research. The core data set and the extended data set have the same 19 topics: psychology, sociology, computer science, economics, medicine, biology, mathematics, philosophy, engineering, business, history, physics, political science, chemistry, geography, geology, art, environmental science, and materials science. We also select the top 50 secondary fields with the largest number of articles for our subsequent analysis.

- `The relevance of the topics`: To investigate the relevance of the topics further, given the two topics A and B, we calculate the probability of B occurring given A's occurrence as follows.

  First, we compute the probability of Topics A and B's occurrence as $P_A = \frac{N_A}{N}$ and $P_B = \frac{N_B}{N}$, where $|N_A|$ and $|N_B|$ represent the total number of papers belong to Topic A and Topic B, respectively. $|N|$ is the total number of papers.

  Second, we calculate the probability of Topics A and B appearing simultaneously as $P_{AB} = \frac{N_{AB}}{N}$, where $|N_{AB}|$ is the number of articles simultaneously belongs to Topic A and Topic B, respectively.

  Finally, we obtain the probability that A appears under the condition that B appears by $P(A|B) = \frac{P_{AB}}{P_B}$.

  Using the above method, we calculate the probability relationships between the top 50 secondary fields containing the largest number of articles for later analysis work.

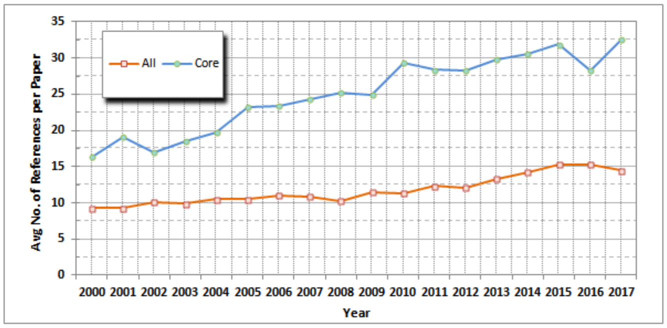


Fig. 7. Average number of references per paper.

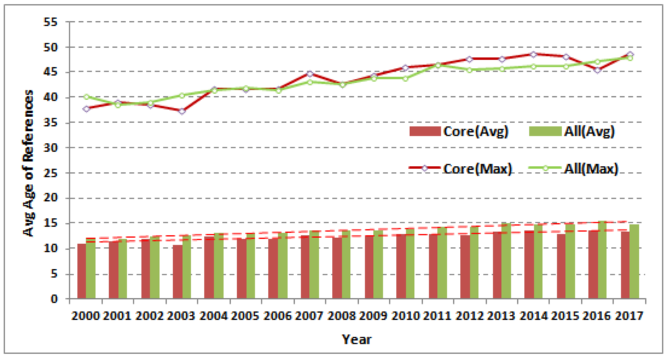


Fig. 8. The average age differences between the cited paper and the citing paper.

- `Proportion of the topic in different years`: To observe the evolution of the topic over time, we use $\theta_k^{[t]}$ [31] to represent the proportion of the topic k at t year. It can be seen that θ is the average topic distribution in all articles. This indicator allows us to quantify the importance of topics over a specific period. We compute the indicator in the root field.
- `Popular topics`: To measure the trend of a field over time, we calculate the variety index between two periods with $r_k = \frac{\sum_{t=2010}^{2017} \theta_k^{[t]}}{\sum_{t=2000}^{2009} \theta_k^{[t]}}$. We compute the indicator in the root field. When $r_k > 1$, this field is more popular in 2010-2017 than in 2000-2009. While $r_k < 1$, the research in this field reduced in 2010-2017. Further, when $r_k = 1$, there is no change.
- `Network of topic co-presence`: Article co-citation analysis is often used to identify developments in the field of research by exploring common citation relationships between references as a basis for assessing and planning scientific and technical research. In a visual network map, the lines in the document co-citation network represent the frequency with which other publications in the same data set refer to both publications. Based on the similarities of the research, the network can be divided into different groups. [31] conducted an experiment of joint citation analysis to reveal the evolution of the field of social simulation. Following their

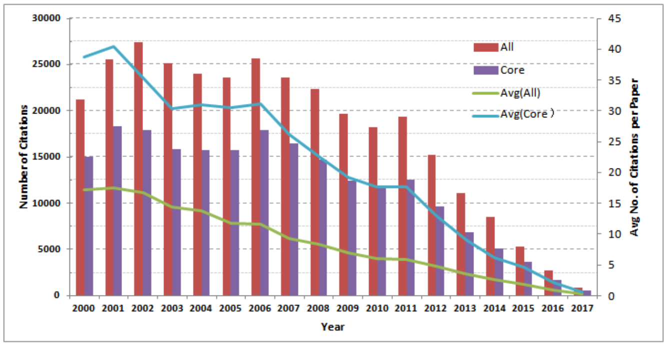


Fig. 9. The number of citations per year and the average number of papers cited.

steps, we use this method to build a topic coexistence network to discover the interconnection patterns between them. Based on the correlations of the subjects P_A, P_B, and P_{AB}, which we compute before, we calculate the coexistence coefficient $co = \frac{P_{AB}^2}{min(P_A,P_B)*mean(P_A,P_B)}$. Therefore, we choose themes with $co(A, B) > 0.1$ to build coexistence. We calculate the index between the top 50 secondary fields containing the largest number of articles for later analysis work.

3. Results

3.1. Evolution of metaphor

As Fig. 1 shows, in the whole development process of metaphor research, the number of papers on metaphor published every year continues to increase in both the core data set and the whole data set. This finding shows that metaphor research has become more and more popular in recent years. Could this be the result of an increase in the number of researchers? To verify this conjecture, as shown in Fig. 2, we analyze the number of authors in the data set, and we found that the growth rate of authors has the same trend as the number of papers, but it is slightly higher (Fig. 3). We conclude that the increase in the number of authors might stimulate an increase in the number of metaphorical papers.

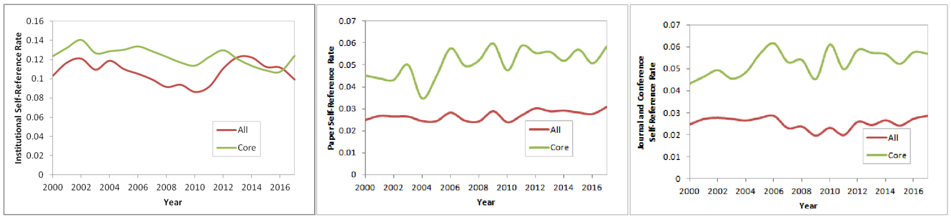


Fig. 10. Institutional self-citation rate, Paper self-citation rate, and Conference and journal self-citation rate.

Table 2. Ranking of papers based on the total number of citations received in 2000-2017 in all dataset papers.

No.	Title	Citations	Published Year
1	Knowledge and organization: A social-practice perspective [5]	2,044	2001
2	The network structure of social capital[6]	1,922	2000
3	Adaptive subgradient methods for online learning and stochastic optimization[12]	1,609	2011
4	From metaphor to measurement: Resilience of what to what?[7]	1,348	2001
5	Community resilience as a metaphor, theory, set of capacities, and strategy for disaster readiness[35]	1,279	2008
6	Social and psychological resources and adaptation[23]	1,234	2002
7	Relational frame theory: A post-Skinnerian account of human language and cognition[21]	1,185	2001
8	Modern social imaginaries[46]	1,150	2002
9	Negotiation as a metaphor for distributed problem solving[9]	1,092	2003
10	Where mathematics comes from: How the embodied mind brings mathematics into being[30]	993	2000
11	Animation: Can it facilitate?[3]	928	2002
12	Model-driven data acquisition in sensor networks[11]	848	2004
13	The evolution of foresight: What is mental time travel and is it unique to humans? [44]	800	2007
14	A thorough benchmark of density functional methods for general main group thermochemistry, kinetics, and noncovalent interactions. [16]	724	2011
15	Self-control relies on glucose as a limited energy source: Willpower is more than a metaphor [14]	723	2007

Since some conferences are held every two years, the number of papers and the overall results are affected. The primary purpose of conferences is to provide opportunities for scientists to communicate and to understand what others are doing. They can publish their research results as soon as possible, which is very important for timely subjects. Journal papers, by contrast, have a longer review cycle, which can lead to fluctuations in growth rates. Most of the data in our statistics come from journal papers, and a small part come from conference papers. Although we put the two types of papers together for statistical analysis, to explain the development of this discipline better, we analyze the growth rate by using the data from every two years as a unit.

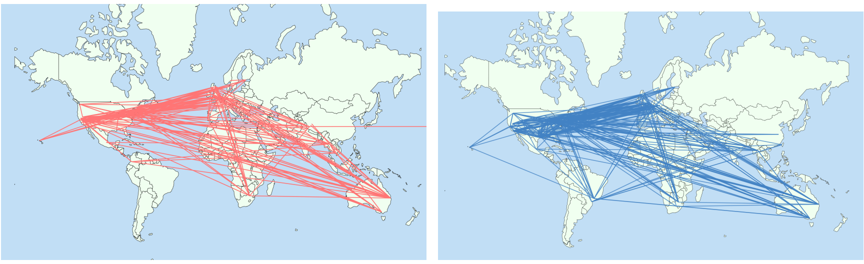
Besides, Fig. 4 plots the average number of authors per paper over time, and a clear upward trend can be seen from the fit curve, indicating that collaborative papers are becoming more common. We also observe that the average number of publications per author declined over time (Fig. 5), indicating that average productivity was weakening during this period.

Table 3. Ranking of authors based on the average number of citations per paper during 2000-2017 (Core dataset Author).

No.	Name	Organization	Citation	Paper	Citations per Paper
1	John Seely Brown	PARC	2044	1	2044
2	Paul Duguid	University of California, Berkeley	2044	1	2044
3	John C. Duchi	Stanford University	1609	1	1609
4	Elad Hazan	Princeton University	1609	1	1609
5	Yoram Singer	Hebrew University of Jerusalem	1609	1	1609
6	Steve Carpenter	University of Wisconsin-Madison	1348	1	1348
7	Nick Abel	Commonwealth Scientific and Industrial Research Organisation	1348	1	1348
8	J. Marty Anderies	Commonwealth Scientific and Industrial Research Organisation	1348	1	1348
9	Brian Walker	Commonwealth Scientific and Industrial Research Organisation	1348	1	1348
10	Rose L. Pfefferbaum	Phoenix College	1279	1	1279
11	Karen Fraser Wyche	University of Oklahoma Health Sciences Center	1279	1	1279
12	Betty Pfefferbaum	University of Oklahoma Health Sciences Center	1279	1	1279
13	Fran H. Norris	Dartmouth College	1279	1	1279
14	Susan P. Stevens	Dartmouth College	1279	1	1279
15	Stevan E. Hobfoll	Rush University Medical Center	1234	1	1234

3.2. Impact and citation analysis

A dramatic increase in the number of references (Fig. 6) indicates that researchers are more focused on the work of others. The reason for this phenomenon may be the increase in the number of references per paper and the increase in the number of published papers (Fig. 1). From Fig. 7, we can see the change in the average number of references for each paper from 2000 to 2017. In the core data set, the average number of references per paper increased from 16 in 2000 to 32 in 2017. Papers in all data sets have the same trend (from 9 in 2000 to 15 in 2015). Fig. 8 shows the average and the maximum age difference between the cited paper and the citation. We can see that the age difference in the reference cited by researchers shows a prolonged and tortuous growth trend. The main reason for this phenomenon is that scholars cite papers in different ways: one is mainly referring to classic papers, and another refers to the latest papers. In 2012, [28] first used deep learning to classify high-resolution images, confirming that deep convolutional neural networks are superior to traditional machine learning techniques. More and more scholars are trying to keep abreast of the latest developments, which reduces the average age difference between citations and cited papers, and which restricts the impact of reference classic papers.



(a) Total dataset

(b) Core dataset

Fig. 11. The overview of Metaphor citation relationships between 2000 and 2017. The lines represent the citation relationships among the top 50 most-cited institutions.

In general, the more recently published papers are, the fewer the people who read and cite them, the shorter the time of citations, and the lesser the impact. For example, if a paper published today is not known to anyone, it will not be cited. Therefore, the average number of citations per paper should decrease as publication time approaches. However, as shown in Figure 9, there are still years with increased citations, such as 2001, 2006, and 2011, indicating that the papers published in those years are more influential than other years.

3.3. Identifying influential Papers/ Researchers/ Institutions

Figure 10 shows three average self-citation rates: institutional self-citation, paper self-citation, and journal and conference self-citation. In recent years, there has been no obvious growth trend in these areas. As time goes by, scientists are increasingly citing self papers, which may be the reason for the increase in the number of references to a paper.

We use citations to quantify the importance of paper/res-earcher/institution in metaphor research. For example, we consider the most cited papers from 2000 to 2017 as the most influential papers. Table 1 and Table 2 show the ranking of papers from 2000 to 2017

Table 4. Ranking of authors based on the average number of citations per paper during 2000-2017 (All dataset Author).

No.	Name	Organization	Citation	Paper	Citations per Paper
1	John Seely Brown	PARC	2044	1	2044
2	Paul Duguid	University of California, Berkeley	2044	1	2044
3	John C. Duchi	Stanford University	1609	1	1609
4	Elad Hazan	Princeton University	1609	1	1609
5	Yoram Singer	Hebrew University of Jerusalem	1609	1	1609
6	Steve Carpenter	University of Wisconsin-Madison	1348	1	1348
7	Nick Abel	Commonwealth Scientific and Industrial Research Organisation	1348	1	1348
8	J. Marty Anderies	Commonwealth Scientific and Industrial Research Organisation	1348	1	1348
9	Rose L. Pfefferbaum	Phoenix College	1279	1	1279
10	Karen Fraser Wyche	University of Oklahoma Health Sciences Center	1279	1	1279
11	Betty Pfefferbaum	University of Oklahoma Health Sciences Center	1279	1	1279
12	Fran H. Norris	Dartmouth College	1279	1	1279
13	Susan P. Stevens	Dartmouth College	1279	1	1279
14	Stevan E. Hobfoll	Rush University Medical Center	1234	1	1234
15	Bryan Roche	Maynooth University	1185	1	1185

based on total citations. By ranking the papers of the two data sets, respectively, the comparison shows that the first eight papers are the same. From the ranking of these papers, we can identify the key issues and keywords in different periods. For example, many social studies in these papers indicate that scholars have invested a lot of time and energy in exploring the relationship between metaphor and society.

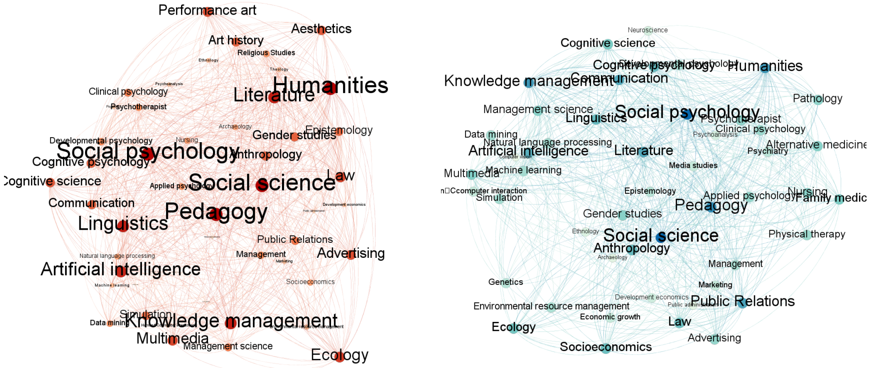
Table 3 and Table 4 list the top 15 researchers who cited the most times, as well as the total number of papers they published, the total number of citations, and their affiliations. Although some researchers have published very few papers, they have achieved high citation rates. The quantified top 15 authors with strong influence do not change much between the two data sets.

Research institutions can be seen as clusters of researchers. Table 5 lists the top 15 institutions based on average citations per paper, total number of authors who have published in top journals/conferences, total number of citations, and total number of articles published in top journals/conferences, in addition to standard deviation of citations (SD) per author and institution. The lower the SD value, the closer the point in the data set is to the average. This can help readers to understand the importance of the target author/institution better. We can see that most of the influential institutions located in North America, Asia, Europe, and Oceania.

Fig. 11 shows the world map embedded with the top 50 most cited institutions and their citation relationships with each other. This can be seen as an overview of citation relationships between influential institutions. According to the citation ranking of papers,

Table 5. Ranking of institutions based on the average number of citations per paper from 2000-2017.

No.	Institution	Number of Researchers	Total Number of Citations	Total Number of Publications	Avg No. of Citations per Paper	Standard Deviation
1	University of California, Berkeley	82	5,636	83	67.90361446	254.5490328
2	Stanford University	69	4,744	75	63.25333333	216.1849242
3	Harvard University	112	3,822	101	37.84158416	77.58578899
4	University of Chicago	52	3,524	51	69.09803922	270.0703801
5	University of Toronto	94	3,042	91	33.42857143	56.17752474
6	University of Melbourne	68	2,823	59	47.84745763	122.0336038
7	McGill University	61	2,776	56	49.57142857	160.2311804
8	University of Wisconsin-Madison	52	2,747	46	59.7173913	199.2495546
9	Princeton University	34	2,691	38	70.81578947	259.6296588
10	Northwestern University	67	2,680	59	45.42372881	92.851235
11	University of British Columbia	85	2,574	77	33.42857143	90.10248968
12	University of California, Los Angeles	70	2,519	57	44.19298246	82.73396232
13	Lancaster University	74	2,433	88	27.64772727	73.89679659
14	University of California, San Diego	65	2,422	63	38.44444444	131.9994841
15	University of Arizona	46	2,261	39	57.97435897	136.5078535



(a) Total dataset

(b) Core dataset

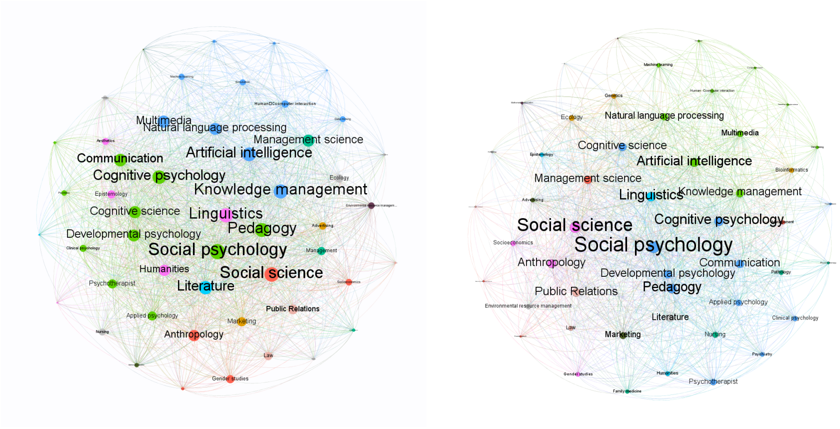
Fig. 12. Co-presence network of topics.

influential institutions are distributed in Asia, Europe, North America, and Oceania. As we can see from the figures, citation relations exist widely between North America and Europe. This shows that the dissemination of knowledge is becoming more and more global, and the way it is referenced is also very different. The size of the solid region on the map represents the relative number of agencies cited, and it can be seen that most agencies cited are located in North America. This may be because these institutions get more citations than others.

3.4. Internal structure

Metaphor is not a single topic; it also contains many themes, which are both independent and interactive. To understand metaphor research in-depth, we can divide metaphors into multiple topics by utilizing the existing fields in the dataset. At least the 19 topics with the broadest scope can be divided according to the Level 0 domain.

Fig. 12 shows the topic co-occurrence network structure as defined in Section 2. Metaphors bring together different topics. For better visualization, we select the top 50 topics with the largest number of papers in the Level-1 field. Fig. 12(a) is the topic co-occurrence network of the total data set, which is composed of 779 lines and 50 nodes. Fig. 12(b) shows the topic co-occurrence network of the core data set, which is composed of 1,238 lines and 50 nodes. The weight of the lines in the network graph is the coexistence coefficient calculated in the second section, and the degree of topic connection determines the size of the nodes so that the graph can reflect the internal topic structure of metaphor to some extent. The co-occurrence networks of the two data sets are much the same. As shown in the figures, in a paper, it is possible to include topics such as social science, social psychology, and pedagogy. This shows that metaphors contain a variety of topics, their impact, life cycle, and development are different, but they are all interactive. Cross-domain research will promote the continuous development of metaphor.



(a) Total dataset (b) Core dataset

Fig. 13. Cross-reference network of topics.

Also, we apply the methods described above. We divide metaphors into 50 different themes, which are organized by metaphors. Fig. 13 depicts these topics and their references to each other. Unlike the co-occurrence network mentioned above, the weights of lines in the cross-reference network are measured according to the number of papers on the cited topic. Nodes of the same color belong to the same Level-0 field. For example, in Fig. 14, the green nodes consist of the sub-fields of computer science in the Level-0, such as natural language processing, artificial intelligence, and multimedia, and the blue nodes comprise the sub-field of psychology such as social psychology, pedagogy, and cognitive psychology. Through the connections of different color nodes, it is easy to see that metaphor research is cross-domain rather than independent.

In addition, to reorganize the topic dynamically, as defined in the previous section, we took $\theta_k^{[t]}$ over the evolution of topic k. Fig. 14 shows the topic change over time in 19 domains at Level 0 from 2000 to 2017. These topics are ranked from bottom to top in terms of popularity. From the core data sets, it can be seen that metaphor research focuses more on topics such as psychology and sociology, and it pays less attention to environmental science, materials science, and other topics. This figure also clearly reflects the evolution of the topics, some of which have been declining over time, while others have received much attention.

To investigate the popularity of topics further, we use the r_k defined in Section 2 to evaluate these topics. Table 6 lists r_k estimates for all topics in descending order. The hottest topics are chemistry, business, and physics.

4. Conclusion

In this paper, we undertook a bibliographic analysis of metaphor research in the 21 st century. To reflect the universality of the law, we took 11,564 articles from the Web of Science as the core data set, and 44,586 papers from MAG as the whole data set. We

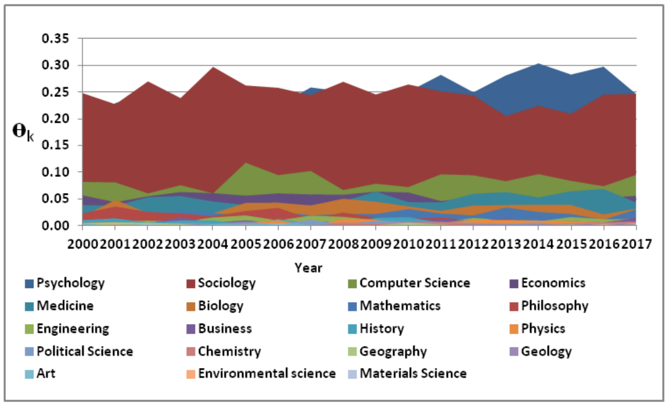


Fig. 14. The evolution of core datasets' topics over time.

Table 6. Increase index for popular topics.

Topic	rk	Topic	rk	Topic	rk
Chemistry	2.02	Medicine	0.96	Geography	0.77
Business	1.10	Mathematics	0.94	Economics	0.74
Physics	1.07	Computer Science	0.85	Sociology	0.74
Geology	0.99	Political Science	0.82	Materials Science	0.73
Art	0.30	Philosophy	0.38	History	0.58
Environmental Science	0.69				

perform the same calculations and compare the results of the two data sets. We conduct statistical analyses of the titles, authors, institutions, and reference data of each paper. We also provide a relatively comprehensive review of metaphor development over the past 18 years.

We found that the results of the two data sets are roughly the same. From the perspective of publications, authors, citations, and references, metaphor research generally shows an upward trend. From the perspective of changes in reference behavior, the development trend of metaphor is open and popular, which is reflected over time. The number of references is increasing, and cross-domain metaphor research is becoming more and more common. From the changes in the number of citations and publications, we observe that the trend of cooperation is becoming more and more obvious, and the average productivity of each researcher is declining. To quantify the development of metaphor studies better, we use the average number of citations of each paper per author/author/institution as an indicator of its importance, ranking the importance of the paper/author/institution, and screening out excellent papers, authors, and countries in the field of metaphor research. Finally, we explore the internal structure of metaphors, and we conclude that the field contains a variety of complex and changing themes, with differences and connections between them. These findings reveal the evolution of potential patterns and themes in the metaphorical world, helping researchers continue to explore the field and providing them with novel insights.

Acknowledgments. This work is partially supported by National Natural Science Foundation of China under Grants No. 62076051. We would like to thank Wei Zhang for help with experiments.

References

1. Aziz-Zadeh, L., Damasio, A.: Embodied semantics for actions: Findings from functional brain imaging. Journal of Physiology-Paris 102(1-3), 35–39 (2008)
2. Belhassen, Y., Caton, K., Vahaba, C.: Boot camps, bugs, and dreams: Metaphor analysis of internship experiences in the hospitality industry. Journal of Hospitality, Leisure, Sport & Tourism Education 27, 100228 (2020)
3. Bétrancourt, M., Tversky, B., Morrison, J.: Animation: can it facilitate. Int. J. of Human Computer Studies 57(4), 247–262 (2002)
4. Boroditsky, L.: Metaphoric structuring: Understanding time through spatial metaphors. Cognition 75(1), 1–28 (2000)
5. Brown, J.S., Duguid, P.: Knowledge and organization: A social-practice perspective. Organization science 12(2), 198–213 (2001)
6. Burt, R.S.: The network structure of social capital. Research in organizational behavior 22, 345–423 (2000)
7. Carpenter, S., Walker, B., Anderies, J.M., Abel, N.: From metaphor to measurement: resilience of what to what? Ecosystems 4(8), 765–781 (2001)
8. Correia, A., Paredes, H., Fonseca, B.: Scientometric analysis of scientific publications in cscw. Scientometrics 114(1), 31–89 (2018)
9. Davis, R., Smith, R.G.: Negotiation distributed as a metaphor for problem solving. Lecture Notes in Computer Science 2650, 51–97 (2003)
10. Delgado López-Cózar, E., Orduña-Malea, E., Martín-Martín, A.: Google Scholar as a Data Source for Research Assessment, pp. 95–127. Springer (2019)
11. Deshpande, A., Guestrin, C., Madden, S.R., Hellerstein, J.M., Hong, W.: Model-driven data acquisition in sensor networks. In: Proceedings of the thirtieth international conference on very large data bases. pp. 588–599 (2004)
12. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159 (2011)
13. Fortin, I.: Entrepreneurial gravity: An additional metaphor of tie structure. Academy of Management Proceedings 2020(1), 12468 (2020)
14. Gailliot, M.T., Baumeister, R.F., DeWall, C.N., Maner, J.K., Plant, E.A., Tice, D.M., Brewer, L.E., Schmeichel, B.J.: Self-control relies on glucose as a limited energy source: willpower is more than a metaphor. Journal of personality and social psychology 92(2), 325 (2007)
15. Garson, J.: The origin of the coding metaphor in neuroscience. Behavioral and Brain Sciences 42 (2019), e227
16. Goerigk, L., Grimme, S.: A thorough benchmark of density functional methods for general main group thermochemistry, kinetics, and noncovalent interactions. Physical Chemistry Chemical Physics 13(14), 6670–6688 (2011)
17. Gulli, R.A.: Beyond metaphors and semantics: A framework for causal inference in neuroscience. Behavioral and Brain Sciences 42 (2019), e230
18. Haggerty, K.D., Ericson, R.V.: The surveillant assemblage. The British journal of sociology 51(4), 605–622 (2000)
19. Hall Maudslay, R., Pimentel, T., Cotterell, R., Teufel, S.: Metaphor detection using context and concreteness. In: Proceedings of the Second Workshop on Figurative Language Processing. pp. 221–226. Association for Computational Linguistics, Online (Jul 2020)
20. Hatton, H., Porter, J.: The power of metaphorical language in treaty diplomacy. Tech. rep., for World report of languages 2019, Report by United Nations General Assembly 2019 Year of Indigenous Languages Awaiting confirmation of publication Open Access

21. Hayes, S.C., Barnes-Holmes, D., Roche, B.: Relational frame theory. Springer (2001)
22. Hellberg, D.: Funny in the bones: The neural interrelation of humor, irony, and metaphor as evolved mental states. Interdisciplinary Literary Studies 20(3), 237–254 (2018)
23. Hobfoll, S.E.: Social and psychological resources and adaptation. Review of general psychology 6(4), 307–324 (2002)
24. Kaushal, V., Patwardhan, M.: Emerging trends in personality identification using online social networks—a literature survey. ACM Transactions on Knowledge Discovery from Data (TKDD) 12(2), 15 (2018)
25. Kong, X., Ma, K., Hou, S., Shang, D., Xia, F.: Human interactive behavior: A bibliographic review. IEEE Access 7, 4611–4628 (2018)
26. Kong, X., Shi, Y., Yu, S., Liu, J., Xia, F.: Academic social networks: Modeling, analysis, mining and applications. Journal of Network and Computer Applications 132, 86 – 103 (2019)
27. Kopp, R.R.: Metaphor therapy: Using client generated metaphors in psychotherapy. Routledge (2013)
28. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
29. Lakoff, G., Johnson, M.: Metaphors we live by (1980)
30. Lakoff, G., Núñez, R.E.: Where mathematics comes from: How the embodied mind brings mathematics into being. Basic Books (2001)
31. Liu, J., Kong, X., Xia, F., Bai, X., Wang, L., Qing, Q., Lee, I.: Artificial intelligence in the 21st century. IEEE Access 6, 34403 – 34421 (2018)
32. Liu, J., Xia, F., Wang, L., Xu, B., Kong, X., Tong, H., King, I.: Shifu2: A network representation learning based model for advisor-advisee relationship mining. IEEE Transactions on Knowledge and Data Engineering (2019)
33. Markowitz, D.M.: The deception faucet: A metaphor to conceptualize deception and its detection. New Ideas in Psychology 59, 100816 (2020)
34. Meyer, M., Lorscheid, I., Troitzsch, K.G.: The development of social simulation as reflected in the first ten years of jasss: a citation and co-citation analysis. Journal of Artificial Societies and Social Simulation 12(4), 12 (2009)
35. Norris, F.H., Stevens, S.P., Pfefferbaum, B., Wyche, K.F., Pfefferbaum, R.L.: Community resilience as a metaphor, theory, set of capacities, and strategy for disaster readiness. American journal of community psychology 41(1-2), 127–150 (2008)
36. Parde, N., Nielsen, R.D.: Exploring the terrain of metaphor novelty: A regression-based approach for automatically scoring metaphors. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
37. Piekkari, R., Tietze, S., Koskinen, K.: Metaphorical and interlingual translation in moving organizational practices across languages. Organization Studies 41(9), 1311–1332 (2020)
38. Richards, I.A., Constable, J.: The philosophy of rhetoric. Oxford University Press (1965)
39. Rincón-Ruiz, A., Rojas-Padilla, J., Agudelo-Rico, C., Perez-Rincon, M., Vieira-Samper, S., Rubiano-Paez, J.: Ecosystem services as an inclusive social metaphor for the analysis and management of environmental conflicts in colombia. Ecosystem Services 37 (2019), 100924
40. Santos, F.C., Pacheco, J.M.: Scale-free networks provide a unifying framework for the emergence of cooperation. Physical Review Letters 95(9), 098104 (2005)
41. Semino, E.: Corpus linguistics and metaphor, pp. 463–476. Cambridge University Press (2017)
42. Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.j.P., Wang, K.: An overview of microsoft academic service (mas) and applications. In: Proceedings of the 24th international conference on world wide web. pp. 243–246 (2015)
43. Steen, G.: A method for linguistic metaphor identification: From MIP to MIPVU. John Benjamins Publishing (2010)
44. Suddendorf, T., Corballis, M.C.: The evolution of foresight: What is mental time travel, and is it unique to humans? Behavioral and brain sciences 30(3), 299–313 (2007)

45. Sun, L., Yin, Y.: Discovering themes and trends in transportation research using topic modeling. Transportation Research Part C: Emerging Technologies 77, 49–66 (2017)
46. Taylor, C.: Modern social imaginaries. Public culture 14(1), 91–124 (2002)
47. Tendahl, M., Gibbs Jr, R.W.: Complementary perspectives on metaphor: Cognitive linguistics and relevance theory. Journal of pragmatics 40(11), 1823–1864 (2008)
48. Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E., Dyer, C.: Metaphor detection with cross-lingual model transfer. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. pp. 248–258 (2014)
49. Visser, M., van Eck, N.J., Waltman, L.: Large-scale comparison of bibliographic data sources: Scopus, Web of Science, Dimensions, Crossref, and Microsoft Academic. arXiv e-prints arXiv:2005.10732 (May 2020)
50. Wan, H., Zhang, Y., Zhang, J., Tang, J.: Aminer: Search and mining of academic social networks. Data Intelligence 1(1), 58–76 (2019)
51. Wang, K., Shen, Z., Huang, C., Wu, C.H., Dong, Y., Kanakia, A.: Microsoft academic graph: When experts are not enough. Quantitative Science Studies 1(1), 396–413 (2020)
52. Wang, R., Sun, K.: Bodo winter, sensory linguistics: Language, perception and metaphor. Folia Linguistica 54(1), 269 – 275 (2020)
53. Wang, W., Liu, J., Xia, F., King, I., Tong, H.: Shifu: Deep learning based advisor-advisee relationship mining in scholarly big data. In: Proceedings of the 26th international conference on world wide web companion. pp. 303–310 (2017)
54. Woodside, J.M.: Organizational health management through metaphor: a mission-based approach. Journal of health organization and management 32(3), 374–393 (2018)
55. Wu, C., Wu, F., Chen, Y., Wu, S., Yuan, Z., Huang, Y.: Neural metaphor detecting with cnn-lstm model. In: Proceedings of the Workshop on Figurative Language Processing. pp. 110–114 (2018)
56. Xia, F., Wang, W., Bekele, T.M., Liu, H.: Big scholarly data: A survey. IEEE Transactions on Big Data 3(1), 18–35 (2017)

Dongyu Zhang received the Master degree in applied linguistics from Leicester University, UK, and the PhD degree from Dalian University of Technology, Dalian, China. She is currently a Full Professor in School of Software, Dalian University of Technology, Dalian, China. Her research interests include natural language processing, sentiment analysis, and social computing. She is a member of the Association for Computational Linguistics and the China Association of Artificial Intelligence.

Minghao Zhang received the Bachelor degree in cyber engineering from Dalian University of Technology, Dalian, China. He is pursuing a Master degree in Software Engineering at Dalian University of Technology, Dalian, China. His research interests include metaphor identification and sentiment analysis.

Ciyuan Peng is an MSc student in the Department of Computer Engineering, Chung-Ang University, Seoul, Korea. She received the BSc degree from the School of Computer and Information Science, Chongqing Normal University, Chongqing, China in 2018. Her research topics include sentiment analysis, natural language processing, data mining, deep learning, and knowledge engineering.

Jason J. Jung is a Full Professor in Chung-Ang University, Seoul, Korea, since September 2014. Before joining CAU, he was an Assistant Professor in Yeungnam University, Korea since 2007. He was a postdoctoral researcher in INRIA Rhone-Alpes, France in 2006, and a visiting scientist in Fraunhofer Institute (FIRST) in Berlin, Germany in 2004. He received the B.Eng. in Computer Science and Mechanical Engineering from Inha University in 1999. He received M.S. and Ph.D. degrees in Computer and Information Engineering from Inha University in 2002 and 2005, respectively. His research topics are knowledge engineering on social networks by using many types of AI methodologies, e.g., data mining, machine learning, and logical reasoning. Recently, he have been working on intelligent schemes to understand various social dynamics in large scale social media.

Feng Xia received the BSc and PhD degrees from Zhejiang University, Hangzhou, China. He is currently an Associate Professor and Discipline Leader in School of Engineering, IT and Physical Sciences, Federation University Australia. Dr. Xia has published 2 books and over 300 scientific papers in international journals and conferences. His research interests include data science, social computing, and systems engineering. He is a Senior Member of IEEE and ACM.

Received: November 09, 2020; Accepted: December 20, 2020.

Computer Science and Information Systems 18(1):323–350 https://doi.org/10.2298/CSIS200922057K

Incorporating privacy by design in Body Sensor Networks for Medical Applications: A Privacy and Data Protection Framework

Christos Kalloniatis[1], Costas Lambrinoudakis[1], Mathias Musahl[2], Athanasios Kanatas[1], and Stefanos Gritzalis[1]

[1]Dept. of Digital Systems, University of Piraeus,
GR 18532, Piraeus, Greece
{chkallon,clam,kanatas,sgritz}@unipi.gr
[2]German Research Center for Artificial Intelligence,
67663 Kaiserslautern, Germany
mathias.musahl@dfki.de

Abstract. Privacy and Data protection are highly complex issues within eHealth/M-Health systems. These systems should meet specific requirements deriving from the organizations and users, as well as from the variety of legal obligations deriving from GDPR that dictate protection rights of data subjects and responsibilities of data controllers. To address that, this paper proposes a Privacy and Data Protection Framework that provides the appropriate steps so as the proper technical, organizational and procedural measures to be undertaken. The framework, beyond previous literature, supports the combination of privacy by design principles with the newly introduced GDPR requirements in order to create a strong elicitation process for deriving the set of the technical security and privacy requirements that should be addressed. It also proposes a process for validating that the elicited requirements are indeed fulfilling the objectives addressed during the Data Protection Impact Assessment (DPIA), carried out according to the GDPR.

Keywords: privacy protection, data protection, GDPR, Framework.

1. Introduction

The medical developments and the rapid changes in the Europe demographics are increasing promptly the average age of European citizens. These changes pose several challenges for EU future and require urgent policy responses in order for EU to organize appropriate healthcare solutions addressing to a growing number of individuals [1]. This necessity is leading to a broader enhancement and application of Information and Communication Technology (ICT) within healthcare systems, emerging the establishment of the eHealth systems, where services and tools, based on ICT, can improve prevention, diagnosis, treatment and monitoring [2], as wells as the healthcare-related data management and exchange [3]. Plenty of research approaches in the domain of eHealth systems put special emphasis on the design of remote health-monitoring systems and equipment [1, 4-5], leading medical and public health practice to a large-scale adoption of mobile medicine/mHealth [6]. mHealth is supported by mobile devices, such as mobile phones, patient monitoring devices, personal digital assistants and other wireless devices that provide remote access to health services and users. Especially, body-worn monitoring devices and wireless medical sensor networks are on emerge [1,7]. Wireless medical sensor networks are considered of major impact on e-healthcare [8], providing plenty of benefits such as: ease of use, reduced risk of infection, reduced risk of failure, reduced patient discomfort, enhancing of mobility and low cost of care delivery, while allowing the data of a patient's vital body parameters to be collected by wearable or implantable biosensors, such as heart rate monitors, pulse oximeters, spirometers and blood pressure monitors [7]. Despite the fact that medical apps enable remote health monitoring, they are also raising security and privacy concerns due to the basic personal e-health systems' functionalities, such as: personal data storage and processing, personal data exchange with other third-party systems (personal or institutional), integration of (personalized) public data, exporting personal data for statistical use and exchange of private personal data messages [5]. These are considered to be some of the most important barriers for the fully implementation of e-healthcare solutions [9]. The "privacy paradox" significantly affects the effectiveness of existing solutions, since most of the users are concerned about the protection of their medical data and at the same time, they agree on using these devices, since they are vital for their health [10]. There are also cases of users who are not aware of the potential harm that can be caused from a deliberate or an accidental threat. In general, privacy and personal data protection constitutes a major concern for e-health consumers [11], affecting wearables adoption. Indicatively, 82% of the respondents to a PricewaterhouseCoopers (PwC) survey reported that they are worried that wearable technology will invade their privacy [12]. Therefore, users' acceptance is strongly dependent on trust. Additionally, the principles enforced by the new General Data Protection Regulation (GDPR) put special emphasis on the protection of citizens' privacy by elevating the obligations of the parties collecting, distributing and processing users' data [13]. To this respect, privacy engineering has gained much attention in Europe, and lately across the Atlantic, as a significant part of the system development process, where security and privacy software engineers along with developers should define the principles in the form of technical requirements that need to be satisfied in order for the system to ensure a minimum level of security and being trustworthy for the citizens [14]. Thus, as far as the e-health and m-health systems concern, which are critical due to the sensitivity of the collected and distributed data within them, security and privacy have been always of immense interest to research communities. However, most of previous researches [15-16-17-18] focusing on the security and privacy frameworks for m-health applications that provide users with more control regarding the use of their sensitive health data within then, does not combine the benefits of privacy engineering along with the new technical and legal aspects of GDPR legislation in medical wearables applications, while some interesting approaches, aiming to highlight the privacy requirements of the m-health applications within the context of GDPR, focus only on specific target group of patients, such as Mustafa, Pflugel & Philip's [9] study regarding patients suffering from Chronic Obstructive Pulmonary Disease, excluding larger or healthier target groups in which m-health applications are also addressed.

In this regard, BIONIC, a pioneering system, funded by the European Union's Horizon 2020 research and innovation program under Grant Agreement No 826304, going beyond previous research, introduces medical wearables to the workplace in order to form a strong paradigm of how wearable technology can respect the user's rights to privacy, maintaining the highest standards in terms of privacy and data protection and to familiarize many users with these rights and their ability to control the sharing of their data. Its mission has been manifold, aiming at the development of a) a holistic, unobtrusive, b) autonomous and c) privacy preserving platform for real-time risk alerting and continuous persuasive coaching, enabling the design of workplace interventions adapted to the needs and fitness levels of specific ageing workforce. To that aim, it provides a Privacy and Data Protection Framework, which suggests the appropriate steps so as the technical, organizational and procedural measures for the satisfaction of the security, privacy and legal requirements. BIONIC, through this Framework, covers the gaps in existing methodologies, focusing on the increase of the end users' trustworthiness to the developed software. More specifically, it provides the combination of privacy by design principles with the newly introduced GDPR requirements in order to create a strong elicitation process for deriving the set of technical requirements that should be addressed, while it proposes a process for validating that the elicited requirements are indeed fulfilling the objectives addressed during the Data Protection Impact Assessment (DPIA), carried out according to the GDPR.

The rest of the paper is organized as follows. Section 2 presents the issue of privacy within e-Health/m-Health Systems, while subsection 2.1 focuses especially on privacy preserving schemes under GDPR. In subsection 2.2 Privacy by Design approaches are presented and their relevance with GDPR. Section 3 briefly presents the BIONIC system and its main features while 4 presents the proposed privacy Framework. Finally, Section 5 concludes our work.

2. Privacy within e-Health/m-Health systems

The wide prevalence and utilization of mobile medical devices in the area of eHealth, collecting health data about individuals and performing monitoring and managing of health-related information, raises serious challenges for the eHealth systems in order to support effectively privacy protection of individuals' personal data and access control [1]. Considering that health information is a particularly sensitive subset of personal information, accompanied with ethical considerations, privacy is established as one of the main requirements of eHealth area [19]. Since within eHealth there are multiple collaborating parties coming from a diverse range of authorities under different managements, privacy concerns derive from the sensitivity of the data that the applications access, handle, store, use and from how/with who it is shared, indicating highly numbered security weaknesses and privacy threats [20]. In this regard, Omoogun et al. in [21] support that different sensors, used for monitoring within mHealth, are designed without taking into consideration security and privacy aspects and therefore are vulnerable to numerous types of attacks, or subjects to data leaks. Challenges and threats such as eavesdropping, impersonation, data integrity, data breach and collusion pose even more provocations for the privacy-aware management of the patients' personal data to control pervasive tracking and profiling [22]. Additionally, users' privacy concerns may be a great barrier to the acceptance of the eHealth technology [1,23] and in order to adopt socially acceptable health services, the security and privacy issues need to be analyzed and addressed [3]. Privacy is a highly complex issue within eHealth systems, which are designed to meet specific requirements deriving from the organizations and providers who use them, as well as from the variety of legal obligations deriving from GDPR and privacy enforcement rules that dictate protection rights of data subjects and responsibilities of data controllers [3]. Therefore, an adequate framework is needed in order to balance different levels of privacy regarding their data, considering, for providing the appropriate technical solutions, not only the individuals' requirements and needs and the health care service providers' and data controllers' purposes, but also the different sets of regulations for privacy. However, such a Framework in previous literature is lagging behind, as it will be presented in the following subsection.

2.1. Privacy approaches within eHealth systems under GDPR

GDPR provides new definitions regarding individuals' data. Specifically, as far as health/medical data concerns, in GDPR (Article 4-15) is defined as the "personal data related to the physical or mental health of a natural person, including the provision of health care services, which reveal information about his or her health status". Additionally, it is considered to be of sensitive ones according to the Article 8 paragraph 1 of the Directive 95/46/EC and as a special category of personal data according to GDPR (Article 9). Therefore, special emphasis, as well as the explicit consent of data subjects, is required when processing them. As [24] argues, the applying of the principles of privacy by design, in order to build security and privacy into the systems, could be a solution to the privacy concerns regarding e-health data. Hence, despite the importance of privacy protection of health/medical data within e-Health area, which has been recognized in a number of works [20,23], to our best knowledge not many research works approach it from the outset of the design or take into consideration the necessity of compliance with new GDPR requirements. Privacy needs to be considered during systems design and implementations [25]; otherwise as [1] support, plenty of limitations are posed on the system's deployment, leading to a not adequate data protection solution for the users. Additionally, literature [26] highlights that previous research mainly focuses on the security issues and especially on cryptographic techniques (e.g. [7]) that perform client-side encryption of data to protect against untrusted service providers, solving fragmentary some aspects within mHealth, while privacy engineering in this area is lacking. Milutinovic & De Decker in [1] p.53 have presented a list with the privacy requirements that an eHealth system should fulfill in order to be compliant with the legal and ethical requisites and gain a wider acceptance as following: a) personal and recognizable information should be protected by strict access control policies, b) non-medical professionals should not access to patients' information, excluding authorized guardians, c) data access should be logged securely, so that later auditing is enabled and d) flexible control access policies should be revoked or expanded. From a technical aspect, Sawand & Khan in [27] p.534 propose that eHealth monitoring systems should fulfill the following security requirements: a) Trusted Authority, which generates public and secret key parameters, is responsible for the issuing keys, granting as well differential access rights to the patients' based on their attributes and roles, b) cloud service provider, in order to provide secure communication mechanisms, as well secure data storage, processing and retrieval according to the access rights of the requesters, c) registered user, referring to the patients' registration to the trusted authority in order to define their data access and the attributes based on specific access policy and d) data access requester, referring to a doctor, a pharmacist, a researcher and a health care service, whose access rights and are defined by the patient who use the eHealth monitoring system.

Pirbhulal et al. in [28] pp.385-386 suggest a more analytic context regarding remote healthcare systems, including not only security, but also privacy requirements as follows: a) data confidentiality, in order to prevent any disclosure during any data transmission, b) data integrity, in order to protect original data from external attacks, c) data availability, in order for the data to be available to legitimate and authenticated nodes/users, d) data freshness, ensuring that data is updated and no one authorized or not can replay old data, e) scalability, in order to reduce latency and control computational and storing overheads and g) secure key distribution, in order to allow encryption and decryption operations for accomplishing the estimated security and confidentiality. As the previous work, the study of [29], as far as the safety of Wireless Body Area Network systems concern, supports the data confidentiality, the data integrity, data freshness and the secure management requirements, but it also maintains a) the availability of the network, in order to provide at all the times access to patients' information both to healthcare professionals and patients, e.g. in case of an emergency health issue, b) data authentication, by which the applications should be capable to verify that the information is sent from a known trust center, c) dependability, referring to the systems' reliability, since data errors may lead to health-threating issues, d) secure localization, in order to prevent attackers to transmit improper details, such as fake signals about the patient's location, e) accountability, in order for the individuals' personal information to be secured, f) flexibility, referring to the individuals' flexibility of designating the control of their medical data and g) privacy rules and compliance requirement, referring to the need to secure private health information by setting privacy measures, such as rules/polices regarding the right to access to patients' sensitive data, since several regulations are enlisted for health care services. In this regard, ambitious current approaches, such as the study of [30], which developed a framework called Privacy-Protector to preserve the privacy of patients' personal data in IoT-based healthcare applications by presenting a secret sharing scheme, which devises the patients' data and stores it in several cloud servers for optimizing the secret share size and supporting exact-share repairs, while still keeping the advantages of the previous scheme, or previous studies [15-16], focusing on the security and privacy frameworks for m-health applications that provide users with more control regarding the use of their sensitive health data within them, are not taking into consideration legal aspects of privacy that are mandatory for eHealth environments.

Especially, as far as compliance with GDPR concerns, until now few research works consider the new requirements. Authors in [25] focused their study on openEHR, a standard that embodies many principles of secure software for electronic health record and provided a list of requirements for a Hospital Information Systems compliant with GDPR, in order to examine to what extent, the openEHR may be a solution for the compliance to GDPR. Although, matches were found, the results showed that the related to the organizational processes GDPR requirements, hardly could be met by any EHR specification standard. Iwaya's et al. study in [26] p.46 on mobile health data collection systems that have been used by community health workers, provides a list with specific privacy recommendations associated with the privacy principles and challenges emerged from the systems under the GDPR. This includes: a) Transparency-enhancing tools, guidelines for purpose specification, fine-grained access control, anonymization and pseudonymization, data validation and integrity and automated data deletion measures for the principle of Data Quality within mHealth, which refers to the process of transparent data, the purpose specification, the data minimization, the data accuracy and integrity and the data retention, b) Obtaining informed consent and check validity of consent measures for the principle of Legitimate Process, which refers to a legitimate data processing of sensitive data that takes into consideration other relevant legal basis for using personal data, c) Accurate, up-to-date, easily found and understandable information about data controller, purpose, recipients measures for the principle of Information Right of Data Subject, d) TETs for individualized information (e.g., privacy dashboards) and timely response to data subject's information requests and rectifications for the principle of Access Right of Data Subject, e) Provision of interfaces for objections and timely responses measures for the principle of Data Subject's Right to Object, f) Authentication and authorization, secure communication and storage and logging measures for the principle of Security of Data, which refers to personal information confidentiality, integrity and availability and the detection and communication of personal data breaches and g) Compliance with notification requirements and logging measures for the principal of Accountability, which refers to the implementation of safeguard data protection and compliance with data protection provisions to subjects, general public and supervisory authorities. Finally, [9], under the EU project WELCOME, studied the privacy of the mHealth applications for patients suffering from Chronic Obstructive Pulmonary Disease, and proposed the following privacy requirements within the context of GDPR: a) data patients' right to access and modification or erase by the applications in any case of inaccurate measurement, b) patients' right to information for the collected data by the applications and the time period of processing, c) limitation of collected data in accordance to the functionality of the applications in combination with the respective permissions, d) patients' fully awareness of the security measures regarding data storage or transmit to other third parties, e) patients' right to information regarding the risk and benefits of an m-health application, g) the prohibition of using collected data for marketing or profiling purposes, stated on an informed medical consent, h) appropriate security mechanisms for mobile devices, providing access only to authorized users with the proper authentication, i) access controls in order for authorized users and mobile devices to be authenticate, k) integrity of the medical data provided by the applications, l) proper security mechanisms for data storage. However, this interesting approach focuses only on a specific target group of patients, excluding larger or healthier target groups in which m-health applications are also addressed.

2.2. Privacy by Design Schemes under GDPR

Privacy by Design, as it was supported by [31], has been incorporated into the GDPR and it is considered to be the most appropriate approach for meeting the privacy and data protection expectations to a large scale, since it offers realistic solutions in order for legal requirements to be combined with the technical ones [24]. In this regard, the data controllers and processors are obligated to enforce the appropriate technical and organizational measures and procedures to ensure the protection of the data subjects' rights and to be compliant with data protection principles [32]. As [33] support, data protection by design should be managed after the specification of the processing purposes and during the processing itself. The controller is obligated to ensure the security of the system, as well as to enhance Protection Impact Assessment into the architecture of the system in order to safeguard adequately by default individuals' rights regarding their data. Hence, the proposed privacy principles by [31] have been under criticism regarding its difficulties to be implemented into the system requirements [34-35]. Therefore, the stated necessity to embed the technological aspects of privacy within the regulatory field [14] and to bridge GDPR privacy regulations with technical solutions is even more emerged, in order for privacy engineering practice to confront more easily the privacy concerns and the compliance to the Regulation [36-]. To address that need, [37], aiming at translating existing GDPR requirements into technical solution templates for compliant services, defined a catalogue of three types of privacy control patterns, namely: a) general privacy control patterns, b) patterns that affect the data subjects' rights and c) patterns regarding data controllers and processors' obligations. Although authors support that the proposed patterns provide generally applicable privacy guidelines, it is important to note that their work focused only on the following specific GDPR principles, Transparency and Traceability, Purpose Limitation, Data Minimization, Accuracy and Storage Limitation, since the principles of Lawfulness, Fairness, Integrity and Confidentiality and Accountability are considered not to be fulfilled by technical measures in a manageable time limit. Authors in [36] presented an approach based on model transformations, aiming to enable a more constructive approach to privacy by design under the principles of GDPR. Although, their work consists an interesting approach to bridge privacy legal and technical field, it focuses only on limited requirements, such as purpose limitation, or accountability of the data controller and consequently it presents specific technical privacy properties. In this sense, [34] supports that the need of holistic privacy patterns is emerged in order for the systems to achieve compliance with the new GDPR regulation, while even the notable privacy approaches, such as LINDDUN, a risk-based method for modelling privacy threats in order to support software developers in identifying and addressing privacy threats early during software development, should be combined with other goal-oriented approaches so as to be effective. [38] introduced an interesting privacy ontology that models the GDPR main conceptual cores as following: data types and documents, agents and roles, processing purposes, legal bases, processing operations, and deontic operations for modelling rights and duties, focusing on the analysis of deontic operators in order to manage the checking of compliance with the GDPR obligations. However, the study has not yet achieved to integrate the different levels of semantic representation for multiple goals and the analysis was restricted only to the Right to Data Portability. Finally, the current EU project PDP4E [14] aims to integrate data protection approaches such as LINDDUN, PRIPARE and PROPAN into systems engineering methodologies and process models, specializing them to operationalize GDPR compliance. Although, authors recognize the impact of goal-driven approaches, they focus mainly on risk-based approaches as LINDDUN and PROPAN, a threat identification method, as well as on

PRIPARE methodology, derived from a previous EU project [39], which has combined articulations of risk-based methods and privacy by design principles for implementing privacy in practice. Therefore, no one pure goal- oriented methodology is considered, despite the fact that GDPR is considered as a purpose-oriented approach [34]. Thus, it is arguable to maintain that a Privacy by Design, goal-oriented privacy methodology could effectively support the implementation of the privacy technical prerequisites that the GDPR poses itself. With respect to this and taking into account that Privacy Safeguard-PriS [40], a goal-driven Privacy by Design engineering methodology, was considered as an effective one for GDPR-compliant socio-technical systems [41], lacking thus in incorporating legal aspects, we provide the ground for PriS to be implemented in our Framework, by making provision for the legal concepts that GDPR has imposed. To address that, we emphasize on the interrelation between GDPR approach and PriS methodology in a high level. PriS considers privacy requirements as organisational goals (privacy goals), which constraint the causal transformation of organisational goals into processes, and by privacy-process patterns describe the impact of privacy goals to the affected organisational processes. In particular, eight types of privacy goals are recognised, namely: Authentication, Authorisation, Identification, Anonymity, Pseudonymity, Unlinkability, Data Protection and Unobservability. At first, PriS was designed to support traditional privacy-aware information systems. Thus, cloud computing environments introduced a number of new privacy related concepts, leading to an extended version of PriS [42-43] that provides a new set of privacy requirements along with the ones already stated, namely, undetectability, isolation, provenanceability, traceability, interveanability and accountability. The next step is the modelling of the privacy-related organisational processes. These processes aim to support the selection of the system architecture that best satisfies them. Therefore, PriS provides an integrated way-of-working from high-level organisational needs to the IT systems that realize those [40]. On the other hand, GDPR aims a) to promote data collecting and processing organisations' and companies' work, by introducing specific rules and requirements as a primary goal of the organizations, as well as by providing direct instructions for the implementation of data protection, dealing thus with several complex aspects, such as company-level awareness raising, nature – scope - context and purposes of processing, adoption of both organizational and technological data protection processes and measures at the start of a project, cost of implementing the protection measures and documentation of processing operations [32-33] and b) to provide EU citizens with further control on their personal data, while minimizing the threats against their data rights and freedoms [32]. Consequently, the conceptual association with PriS methodology is more than clear, since PriS promotes a set of expressions based on which the whole processes of an organization are considered, starting from the goal level and leading to the selection of the appropriate implementation techniques.

3. The BIONIC Project

BIONIC is a pioneering system, funded by the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 826304, which provides industry with a unified methodology focused on the perspective of ageing in the workplace, by supporting its application in research.

Its mission was manifold, aiming at the development of a holistic, unobtrusive, autonomous and privacy preserving platform for real-time risk alerting and continuous persuasive coaching, enabling the design of workplace interventions adapted to the needs and fitness levels of specific ageing workforce. Since many typologies of industry jobs are physically demanding, it is necessary to support the aging workforce with appropriate tools that can help them to stay at their jobs, assessing the potential impacts of their work activities on their health, and recommending exercises to mitigate those impacts and promote healthy and active lifestyles. In this regard, the BIONIC solution constitutes a valuable tool for achieving these objectives, bringing medical wearable technology into a new paradigm accordingly to the World Health Organisation principles for e-Health and m-Health, by integrating sensor modules in multi-purpose, configurable Body Sensor Networks (BSNs) introducing key enablers of user acceptance based on value, comfort, confidence and trust.

Configurable BIONIC Body Sensor network has been originally conceived as a platform combining a wide variety of connected sensors enabling the build-up of a real-time holistic data model of the human body, by capturing static and dynamic whole-body information, such as e.g., body posture, gait, running style, etc., possibly combined with key bio-signals, such as body temperature, heart rate, and environmental signals. Dynamic monitoring of overall body posture integrated into everyday or work clothing will significantly promote the wide adoption of motion tracking wearables, by eliminating the need to attach sensing devices firmly to the body, thus affecting comfort and possibly impeding movement during work or everyday / physical activity. Existing commercial wearables with open APIs (e.g., smartwatches, sole pressure sensors) will also be integrated to enable a holistic integrated continuum of data to derive self-quantification information. Depending on the specific chronic MSD condition, body-part specific BSN modules in the form of e.g., belts or bandages (for monitoring lower back and knee chronic MSD) will be developed. Detailed monitoring of these body parts will be based on innovative localized biomechanical models, focusing on age-induced constraints and chronic impairments (age adapted body - part specific models). Additionally, BSN allows freedom in the selection and positioning of the sensors on the body, depending on the requirements of the specific application (fitness, performance, medical, ergonomics etc.), while it enables the development of customizable or even multi-functional wearables, i.e. networks with inherent redundancy of sensors, with their functionality configured by application software.

Another major key innovation relates to the concept of AI on a chip, i.e. embedding predictive Artificial Intelligence algorithms in the Body Sensor Network (BSN). Raw data pre-processing at the source prevents immense flows of data being transmitted to remote gateways. Feature extraction at the source will result in informative and non-redundant features, ready to be fed into artificial neural networks. The combination of such machine learning algorithms with biomechanical model-based estimation will allow for deducing relevant and interpretable parameters for efficient real-time, in-field and long-term personalized risk/physical strain and recovery assessment from individual sensor data.

Therefore, continuous personalized on-site assessment of the real capacity of ageing workers, using BIONIC wearables, will allow to derive valuable information both at a personal, as well as at a statistically relevant age dependent level, associating the imbalances and risks with design criteria and recommendations that facilitate the selection and adaptation of appropriate positions, while ageing workers will keep their personal data private. Feedback to the user will be provided in real-time through the BSN to actuators such as haptic, acoustic, visual systems. Communication to external Network is optional and under control of the user, who can decide case by case who will get access to the results or the raw data. To fulfil that, an integrated prototype of monitoring and data presentation software will be used, including three different applications targeting: a) workers, where self-monitoring application providing access to their movement data such as daily and archived statistics, risks identified and exercise recommendations, incorporating advanced UX and intuitive ways of human computer interaction to accommodate ageing users' requirements, ensuring optimum comprehension of the relevant warnings and advice and maximizing the preventive effect of the system, b) managers: where ergonomic risk assessment applications provide real-time feed of selected worker movement information to construction site managers to allow for injury prevention as well as periodical reports, including assessment results and recommendations for workplace interventions and c) doctors, where specially designed application provide doctors with access to workers movement and health information, efficiently structured and prioritized based on ergonomic risk, allowing doctors to support the workers in a timely manner.

In this sense, BIONIC introduces Body Sensors Networks in the everyday life to a market segment, which is not so easy for wearable electronics solutions to reach, i.e. older individuals. BIONIC's strong focus on usability and privacy aspects (e.g. HCI, gamified coaching, GDPR principles) will bring these users the confidence and trust to try more similar solutions which can improve their quality of life. It will also convince them in practice that such technology is not only for the gadget savvy youths but can provide real value related to their health and wellbeing. This integrated methodology within a broader procedure is able to facilitate aspects such as, the management of experience in companies, the transfer of knowledge or the transition to retirement, that are relevant aspects for improving companies' productivity and competitiveness. Figure 1 presents the concept diagram of BIONIC.

The BIONIC project supports two main types of services. The first is conducted during the working hours where every worker receives live feedback form the Body Sensor Network that he/she wears while the second is performed during leisure time with the support of a "lighter" Body Sensor Network consisting of less sensors than the full BSN.

More specifically, the full BSN network consists of a set of sensors attached on the workers workwear. This workwear includes a t-shirt, a helmet/cap and a trouser that the worker wears during his/her work. These sensors provide raw data to an AI chip (located in the trousers) which generates processed data related to the workers current health status. The AI chip interacts with the worker's smart watch mostly for getting additional information for the worker's statues (e.g., heart rate) and for sending notifications to the worker (e.g., alarm messages). Also, the worker possesses a mobile device for handling his/her data and interacting with BIONIC apps as well as for conducting the coaching exercises at home.

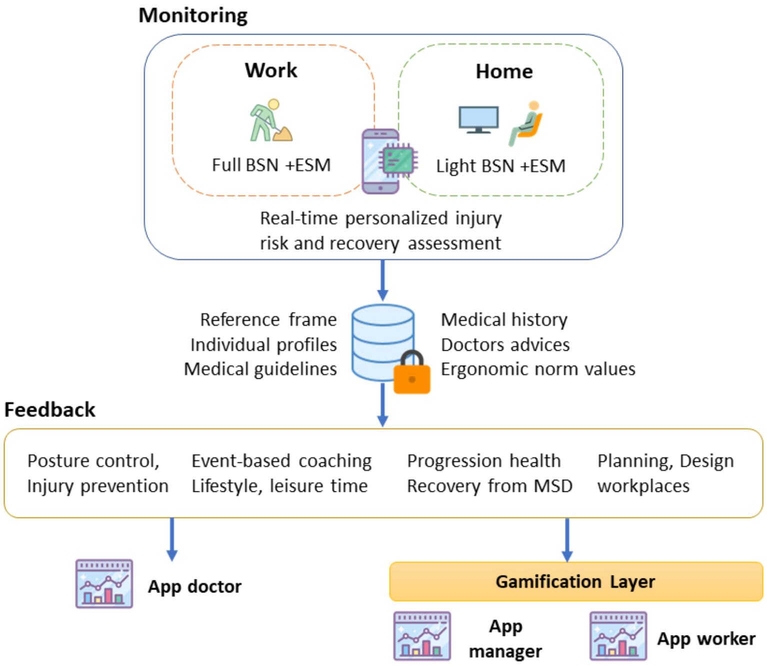


Fig. 1. BIONIC Data Flow

The "light BSN" is the body sensor network consisting of the worker's smartwatch, some Inertial Measurement Units (IMU) sensors and a mobile device and is used from the worker during his leisure time (not working hours) to exercise himself/herself based on the coaching app of the BIONIC project. Beside the use of the specific app the light BSN monitors worker's ergonomic and health data for capturing his/her habits and moves outside the working environment in order to prevent unwanted situation and/or to better advise the worker during the day.

The proposed BIONIC data flow is presented in the following figure. The full BSN and the light BSN parts are also visible in figure 2.

Based on the aforementioned figure BIONIC collects all processed data (as exported from BSN) in a secure storage in order for the developed apps to have access on and mainly provide the necessary feedback to the worker. The data produced by the light BSN are stored in the secure storage repository if the worker wishes to do so. For R&D purposes and only for the duration of the project the raw data produced by the BSN network are stored in an anonymised form in a separate database called "Research Data" in the respective figure.

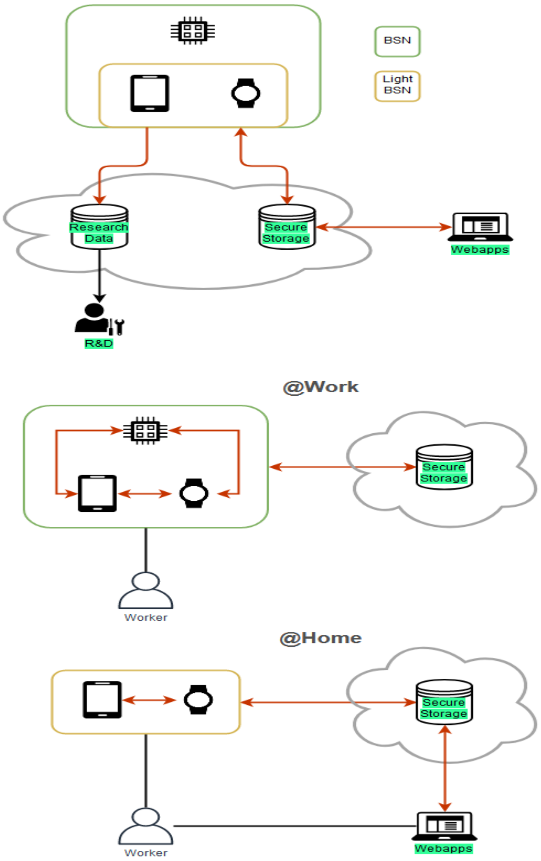


Fig. 2. BIONIC Data Flow

Regarding the types of data collected from the BSN and light BSN networks are mainly the following:

- Kinematic data, probably at IMU sample rate (e.g., joint angles)
- Kinetic data, probably at lower sample rate (e.g., vertical ground reaction force, ground contact information, external load indicators)
- Depending on chosen ergonomic tools:
  - timestamped detected events (e.g., picking something up, body positions) and repetitions
  - possibly ergonomic scores
- Physiological data (Heart rate, blood pressure, body temperature)

Following the context of BIONIC the aim of this paper is to present an efficient Data Protection Framework considering both the technical as well as the legal and organizational requirements for ensuring the safety and security of the workers. The next section presents the proposed Privacy and Data Protection Framework.

4. The Privacy and Data Protection Framework

BIONIC, considering both GDPR principles and PriS concepts, proposes a flexible and efficient GDPR Compliant Personal Data and Privacy Management methodology in order to ensure the security, safety and privacy aspects of using holistic and unobtrusive Body Sensor Networks on ageing workforce, which is going to employ the system in their daily tasks. The Framework ensures that GDPR principles, such as purpose limitation, data minimization, accuracy, accountability, the lawfulness of processing, the user consent, are fully satisfied. In this regard, the medical wearable applications will respect all the data owners' rights (ageing workers), such as their rights to object to the storage/processing of their information, their right to be forgotten, their right to restriction of processing, while the obligations of the intermediate users, such as occupational health professionals, production managers, and in general all the professional profiles that manage the system and potentially exploit all the data generated within BIONIC are specified. Finally, all necessary technical, organizational and procedural measures for the satisfaction of the eliciting security and privacy requirements are considered and thus enhancing confidence and trust among all stakeholders. It comprises of the following four stages:

Stage 1: Personal Data handling processes elicitation

The purpose of this step is to define the perimeter of the Personal Data handling processes, by capturing, reviewing and formalizing the following issues: a) The categories of the personal data processed, b) The categories of the data subjects, c) The purposes of each processing activity, d) The identification of high risk data processing activities, e) The legal basis of each processing activity (e.g. contract and/or consent and/or legitimate interest and/or statutory obligation), f) The categories of recipients to whom the personal data are disclosed, g) The envisaged time limits for erasure of the different categories of data – if they exist, h) The existing technical and organizational measures for the protection of personal data.

To this end, a Questionnaire for Accessing GDPR Compliance is designed and implemented. The objective of this questionnaire, addressed to BIONIC respective stakeholders, is to identify, through a systematic procedure, the aforementioned information for each purpose of data processing separately. The questionnaire includes seven items, requiring open responses, regarding the following issues: a) Short Description for the purpose of processing, b) Legal Basis for the purpose of processing, including the subcategories i) Law, ii)User / patient consent, iii) Contract, c) The data involved in serving the specific purpose of processing, including the subcategories i) General Data, ii) Personal Data, iii) Special categories of Personal Data, d) The necessity of the involved data for serving the specific purpose of processing, e) The sources of collected data, f) The transmission of personal data to third parties, g) The processing of automated decision-making, including profiling. Therefore, its main output concerns the listing of the Purposes of Processing and the Personal Data Categories**.**

Stage 2: Compliance Level on GDPR requirements

During the second stage, it is essential to map the organizational context following the results of stage 1. Therefore, when the above list of information has been compiled, the processing of each personal data category is being reviewed against the GDPR requirements to deduce the existing compliance level, through a GDPR gap analysis.

Topics that are examined are presented indicatively as following: a) lawfulness, fairness and transparency of the personal data processing, b) the processing purpose limitation, the data minimization, c) the consent of the data subjects, d) the personal data storage limitation, e) the measures for personal data protection, f) integrity and confidentiality, g) The readiness of the involved stakeholders to respond to the data subjects' rights' is examined, such as the 'right of access', the 'right to rectification', the 'right to be forgotten', the right to restriction of processing', the 'right to data portability', the 'right to object', h) Information to be provided where personal data have not been obtained from the data subject, i) Automated individual decision-making, including profiling, j) Data protection by design and by default, k) Joint controllers, l) Security of processing, m) Processing under the authority of the controller or processor, n) Tasks of the data protection officer, o) Transfers on the basis of an adequacy decision. Moreover, the readiness of the organization to respond to the data subjects' rights' will be examined. Indicatively the 'right of access' , the 'right to rectification', the 'right to be forgotten', the 'right to restriction of processing', the 'right to data portability', 'right to object'.

Indicative activities that will be performed during the gap analysis include: a) Review of legal basis on which the organization processes Personal Information, b) Review the necessary retention periods per category of Personal Information for various reasons such as, for compliance with a legal obligation, for inquiries of auditing authorities, for legal claims, for public interest etc, c) Review of Privacy Notices, d) Review of the legal basis for marketing services, e) Legal review of all defined internal Personal Information Protection Policies and Procedures, f) Legal Review of sample employment contracts and updating with necessary legal language to allow the processing of employees Personal Information for legitimate business purposes, g) Review of standard consent forms used to collect and record data subject consent for the processing of Personal Information, h) Legal review of standard Intra- and Third-Party contracts, Procurement contracts and Supply Contracts to identify any contractual gaps in relation to Data Protection relevant clauses. If no standard contracts are used, the review will cover key activities, which should at least include all contracts related to identified high risk processing activities, i) Understand the operational policies and procedures for the IT systems, j) Access the efficiency of the organization to protect the data (data protection measures), k) IT and Security Governance review, l) Network architecture review. The main output of this stage concerns a Set of GDPR Compliance Requirements for each identified purpose of Processing.

Stage 3: Security and Privacy requirements elicitation

Security is protection against intended incidents, i.e. incidents that happen due to a result of deliberate or planned act. Security concerns the protection of assets from threats, where these are categorised as "the potential for abuse of protected assets". Whereas, privacy concerns the protection of the assets' owner identity from users that do not have the owner's consent to view/process their data. Risk analysis or equivalently Risk Assessment is the methodology where an IT infrastructure and/or interconnection between computational devices is methodically analysed and the corresponding Security/Privacy threats are identified as long with the specific vulnerabilities and/or potential failures may cause them. Moreover, the goal of a security assessment, is to ensure that necessary security and privacy objectives are integrated into the design and implementation of an architecture. A Vulnerability is defined as a weakness, in terms of security and privacy that exists in from a resource, an actor and/or a goal [17].

Vulnerabilities are exploited by threats, as an attack or incident within a specific context. A Threat represents circumstances that have the potential to cause loss; or a problem that can put in danger the security features of the system [44]. In Stage 3, a Risk analysis, identifying threats, vulnerabilities, data, is conducted in order to deduce attack modelling and threat propagation. The need for such analysis results directly from the GDPR principle of accountability. The analysis assists in the identification and assessment of security and privacy risks and thus in the selection of the appropriate measures to reduce these risks and as such reduce the potential impact of the risks on the data subjects, the risk of non-compliance, legal actions and operational risk. At the end, the privacy by design approach Privacy Safeguard (PriS) is applied to collect all identified security and privacy requirements (Legal, Organizational, Technical), validating that they are indeed expressed in a technically sound manner and ensuring that they can be implemented in the context of the specific system. The requirements elicitation methodology supports threat, vulnerability and attack analysis, reasoning on security and privacy requirements and modelling of the system. PriS was selected for the security and privacy modelling of BIONIC since: i) It consists a privacy by design method, an approach that GDPR sets on its main principles, ii) It is one of the oldest and mostly evaluated privacy by design methodologies, iii) On a conceptual level, GDPR principles' concepts are associated with PriS privacy requirements concepts, iv) It combines stakeholders' needs and goal-driven modelling which is very important due to the purpose-oriented philosophy of the Regulation and v) It can be easily aligned with a risk-based analysis, which is also prerequisite for GDPR. The proposed steps for implementing stage 3 can be described as follows:

Substep1: Identify System Assets and Stakeholders

The purpose of this step is to define the perimeter (boundaries) of the study. A global vision of the components and communications between components will be clarified. At this step, the following data will be collected and formalized: a) Essentials assets of the BIONIC system, b) Functional description of components and relations between components, c) Security issues that need to be addressed by the study, d) Assumptions made if appropriate, e) Existing security rules (law and regulation, existing rules in other studies), f) Constraints (internal or external) from BIONIC system itself. At the end of this step, a clear vision of the components and the links between them will be formalized that are going to be used as input for the risk analysis method.

Substep2: Identify Potential Security and Privacy Threats and related System Vulnerabilities

The security/privacy threats and vulnerabilities affecting the BIONIC system will be studied as outcome from a dedicated risk analysis. The threats and vulnerabilities are going to be specific for the BIONIC's infrastructure components. The following activities will be performed: a) List the relevant attack methods (In collaboration with project partners - experts) against security and privacy, b) Characterize the threat agents for each attack method retained according to their type, c) Identify the security and privacy vulnerabilities of the entities according to attack methods, d) Estimate the vulnerability level, e) Formulate the security and privacy threats, f) Assign priority in the security and privacy threats according to the probability of their occurrence. The list of the pertinent security and privacy threats and the type of attacks will be the main outputs of this step.

Substep3: Security and Privacy Requirements Analysis

From the previous step, the identification of the respective threats and the attack methods that can be deployed to the proposed system leads to the identification of the system's vulnerabilities. At this stage, Security and Privacy vulnerabilities detection will lead to the identification of the security and privacy objectives, which are the way that vulnerabilities are reduced thus reducing the potential risk on the identified entities. PriS methodology will be used as a privacy by design approach in order to analyse from the elicited threats and vulnerabilities the security and privacy goals that will have to be fulfilled. The next step of the specific stage is the definition of the security and privacy requirements that basically describe in a specific way the realization of the identified security and privacy objectives. The following actions will be considered when identifying security and privacy requirements: a) List the security and privacy functional requirements, b) Justify the adequacy of coverage of the security and privacy objectives, c) Highlight any coverage flaws (residual risks) with justifications, d) Classify the Security and privacy requirements for each use case, e) Where appropriate, justify the coverage of dependencies of security and privacy requirements. The main output of this stage concerns a) a List of Threats and Attacks, b) the provision of Legal and Organizational Measures and c) the elicitation of the appropriate security and privacy requirements.

Stage 4: Data Protection Impact Assessment

According to the Regulation (EC) 2016/679 of the European Parliament and of the Council of 27th April 2016 for the protection of natural persons with regard to processing of personal data and on the free movement of such data, where a type of processing in particular using new technologies, and taking into account the nature, scope, context and purposes of the processing, is likely to result in a high risk to the rights and freedoms of natural persons, the controller shall, prior to the processing, carry out an assessment of the impact of the envisaged processing operations on the protection of personal data. A single assessment may address a set of similar processing operations that present similar high risks. Taking into account the systems' and threats' continuous evolution, risk management "necessitates" the identification of appropriate controls. The processing of personal data, the hierarchy and management of risks has to be examined in a way that optimises the cost and contributes to the most suitable decision-making, aiming at protecting personal data. Impact assessment contributes to the application of privacy principles, in a way that the data subjects are able to preserve control of their personal data. A data protection impact assessment, and hence, the criticality of data shall (in accordance with Regulation (EC) 2016/679 of the European Parliament and of the Council) particularly be required in the case of: a) a systematic and extensive evaluation of personal aspects relating to natural persons which is based on automated processing, including profiling, and on which decisions are based that produce legal effects concerning the natural person or similarly significantly affect the natural person (e.g., user profiling by web search activity monitoring for targeted advertising and promotion of products and services (hotels, restaurants, etc.), b) processing on a large scale of special categories of data referred to in Article 9(1), or of personal data relating to criminal convictions and offences referred to in Article 10 (e.g., processing of patients' medical records (special category of personal data) from healthcare organisations, including medical history, illnesses, and patient care, etc.) or c) a systematic monitoring of a publicly accessible area on a large scale (e.g., traffic monitoring for informing drivers of the fastest route, residence entries' monitoring, public transport entrance, etc.).

Moreover, the assessment shall contain, in accordance with Regulation (EC) 2016/679 of the European Parliament and of the Council, at least: a) a systematic description of the envisaged processing operations and the purposes of the processing, including, where applicable, the legitimate interest pursued by the controller; b) an assessment of the necessity and proportionality of the processing operations in relation to the purposes; c) an assessment of the risks to the rights and freedoms of data subjects; d) the measures envisaged to address the risks, including safeguards, security measures and mechanisms to ensure the protection of personal data and to demonstrate compliance with this Regulation taking into account the rights and legitimate interests of data subjects and other persons concerned. This privacy impact assessment is based on a robust conceptual framework related to personal data and data subject protection, to the processing of this data by Information Systems or non-automated means, as well as to an impact analysis of possible incidents of personal data for the data subjects they belong to. Impact assessment aims at the protection of personal data, according to the definition provided by the GDPR, during its processing, as well as the protection of elements that support their processing and are recognised as **Assets**. The value of such assets is equal to the **Impact** brought upon by a possible **violation** of individuals' privacy. A Feared Event is the illegitimate access to personal data, unwanted modification of personal data, as well as the data disappearance. The violation of Information Systems needs the existence of **Vulnerability** and the appearance of a relevant **Threat** coming from a **Risk source**. Summarising, we note that a Threat exploits a vulnerability of an Information System and can have as a result an incident of data protection breach, inflicting some **Impact** on data subjects.

A Risk is a hypothetical scenario that describes how Risk Sources (e.g., an employee bribed by a competitor; could exploit the vulnerabilities in personal data supporting assets (e.g., the file management system that allows the manipulation of data); in a context of threats (e.g., misuse by sending emails); and allow feared events to occur (e.g., illegitimate access to personal data); on personal data (e.g., customer file); thus, generating potential impacts on the privacy of data subjects (e.g., unwanted solicitations, feelings of invasion of privacy, etc.). The risk level is estimated in terms of **severity**, which represents the magnitude of a risk. It essentially depends on the prejudicial effect of the potential impacts, and **likelihood**, which represents the possibility for a risk to occur. It essentially depends on the level of vulnerabilities of the supporting assets facing threats and the level of capabilities of the risk sources to exploit them as shown below.

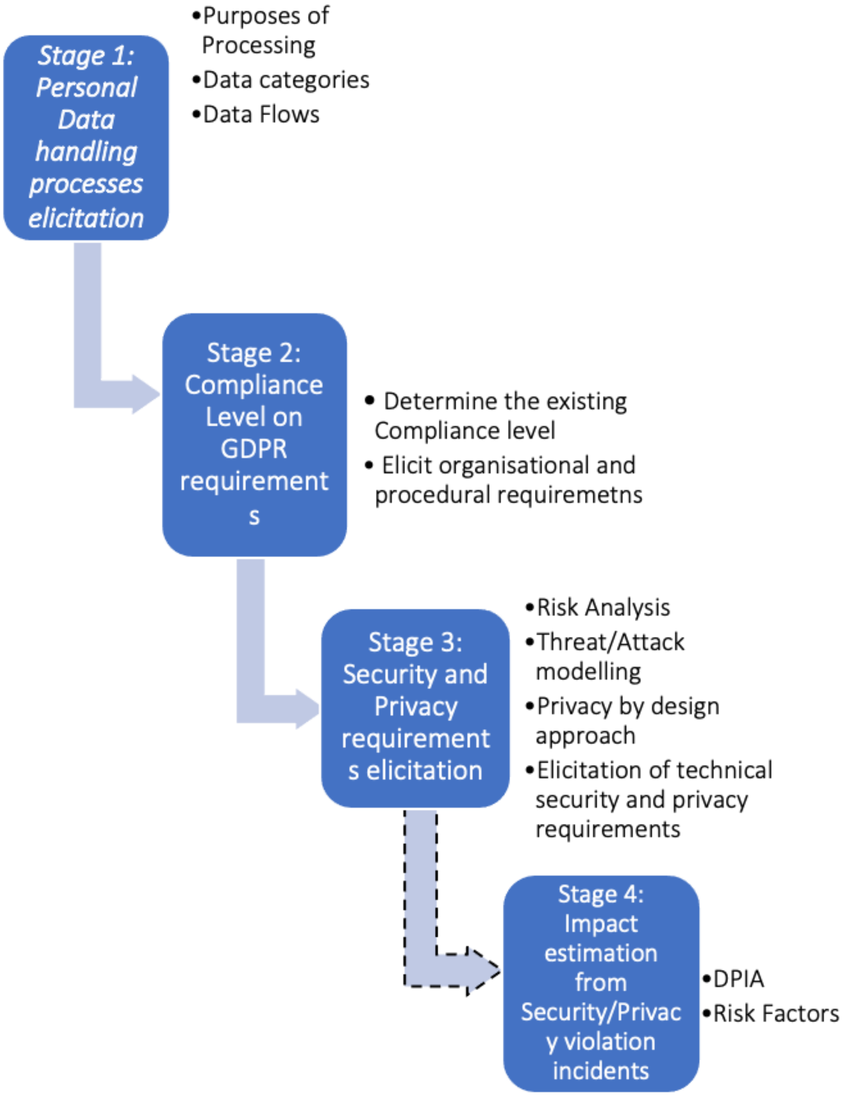

•Purposes of Processing
•Data categories
•Data Flows
Stage 1: Personal Data handling processes elicitation
Stage 2: Compliance Level on GDPR requirements
• Determine the existing Compliance level
• Elicit organisational and procedural requiremetns
Stage 3: Security and Privacy requirements elicitation
•Risk Analysis
•Threat/Attack modelling
•Privacy by design approach
•Elicitation of technical security and privacy requirements
Stage 4: Impact estimation from Security/Privacy violation incidents
•DPIA
•Risk Factors

Fig. 3. BIONIC Privacy and Data Protection Framework

In this stage a data protection impact assessment (DPIA) method will be applied in order to identify the severity and likelihood of any possible privacy violation incidents. During this stage all security and privacy requirements elicited in stage 3 will be evaluated in order to eliminate any possible conflicts prior to the selection of the security and privacy countermeasures. In parallel the DPIA will assist in the identification and assessment of privacy risks and thus in the selection of the appropriate measures to reduce these risks and as such reduce the potential impact of the risks on the data subjects, the risk of non-compliance, legal actions and operational risk. Then the appropriate technical countermeasures for the satisfaction of each requirement will be identified. This information will facilitate the developers to select and proceed with the most suitable implementation techniques for ensuring the security (confidentiality, integrity and availability) of the data processed, as well as the protection of the users' privacy (user consent on data processing and data transmission, satisfaction of user rights etc.). For conducting the DPIA it is necessary to consider both the output of stage 2 regarding the organizational and legal requirements as they are derived from the Gap analysis as well as the output of stage 3 regarding the technical security and privacy requirements derived from the risk/threat analysis and the privacy by design approach. The main output of this stage concerns a) the severity of the privacy violations incidents and b) the likelihood of the privacy violations incidents. The output of stage 4 will enable the developers to select and proceed with the most suitable implementation techniques for ensuring the security (e.g., confidentiality, integrity and availability) of the data processed, as well as the protection of the users' privacy (e.g. user consent on data processing and data transmission, satisfaction of user rights) under GDPR principles. The proposed framework is presented in figure 3.

5. Conclusions

BIONIC acknowledges that the ageing population in Europe impacts multifaceted on the EU productivity growth [45-46] and rises healthcare costs that are leading to the necessity of developing new digital forms of health self-management systems outside the health-care services [47]. Therefore, it provides the development of medical wearables applications for personalized information and treatment to ageing workers, as a form of an m-health system [5], aiming to assist them in managing their health issues and maintaining behaviours that promote health, so as to improve the quality of their work and life. In this regard, it proposes a Privacy and Data Protection Framework that complies with GDPR legislation. The use of the PriS Privacy by Design method is very critical for capturing the required information following the whole software development lifecycle. Particular attention has been given on the different categories of personal data that are processed by the BIONIC platform, such as sensitive data, including recent developments in the field of biometric and health-related data. The Framework also ensures that the applications will respect all the rights of the data owners (workers), such as their right to object to the storage/processing of their information, their right to be forgotten, their right to restriction of processing. Following the proposed privacy-by-design approach, it ensures the satisfaction of all functional requirements, but also of all non-functional (security and privacy) related requirements imposed by the users. Furthermore, during the design phase, all legal and technical requirements of the General Data Protection Regulation (GDPR) are considered, while all necessary technical, organizational and procedural measures to address the GDPR requirements related to data protection, accountability and handling of potential data breaches are taken into account. Respectively, it is ensured that principles like the data protection (purpose limitation, data minimization, accuracy, accountability), the lawfulness of processing, the user consent, are fully satisfied. Following this Framework, the applications respect all the rights of the data owners (workers) like their right to object to the storage/processing of their information, their right to be forgotten, their right to restriction of processing etc. Finally, the implementation of new concepts, such as privacy and data protection by design and by default, accountability, data minimization, lawfulness of processing and users' consent in e-health and m-health systems are enabled, since different stakeholders from several workplaces and domains with different backgrounds may understand the same terms in different ways. In parallel while the elaboration and mapping of the allocation of liability between different actors in interconnected platforms can be conducted, as well as procedures for handling potential data breaches.

Concluding, the proposed Framework, covers the gaps in existing methodologies, focusing on the increase of the end users' trustworthiness to the developed software, by providing a strong elicitation process for deriving the set of technical requirements that should be addressed, while it proposes a process for validating that the elicited requirements, fulfilling the objectives to the GDPR.

Acknowledgment. This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement #826304– BIONIC.

References

1. Milutinovic, M., De Decker, B.: Ethical aspects in eHealth–design of a privacy friendly system. Journal of Information, Communication and Ethics in Society, 14(1), 49-69. (2016)
2. WHO: "E-Health", (2015) [Online]. Available: http://www.who.int/trade/glossary/story021/en/.
3. Esposito, C., Castiglione, A., Tudorica, C. A., & Pop, F:. Security and privacy for cloud based data management in the health network service chain: a microservice approach. IEEE Communications Magazine, 55(9), 102-108. (2017)
4. Chib, A., & Lin, S. H.: Theoretical Advancements in mHealth: A Systematic Review of Mobile Apps. Journal of health communication, 23(10-11), 909-955. (2018)
5. Drosatos, G., Efraimidis, P. S., Williams, G., & Kaldoudi, E.: Towards Privacy by Design in Personal e-Health Systems. In HEALTHINF (pp. 472-477). (2016, February)
6. Marcolino, M. S., Oliveira, J. A. Q., D'Agostino, M., Ribeiro, A. L., Alkmim, M. B. M., & Novillo Ortiz, D: The impact of mHealth interventions: systematic review of systematic reviews. JMIR mHealth and uHealth, 6(1), e23. (2018)
7. Solomon, M., & Elias, E. P.: Privacy Protection for Wireless Medical Sensor Data. *International Journal of Scientific Research in Science and Technology*, 4(2), 1439-1440. (2018)
8. Almarashdeh, I., Alsmadi, M., Hanafy, T., Albahussain, A., Altuwaijri, N., Almaimoni, H.,& Al Fraihet, A.: Real-time elderly healthcare monitoring expert system using wireless sensor network. International Journal of Applied Engineering Research ISSN, 0973-4562. (2018)
9. Mustafa, U., Pflugel, E., & Philip, N.: A Novel Privacy Framework for Secure M-Health Applications: The Case of the GDPR. In 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3) (pp. 1-9). IEEE. (2019, January)

10. Lee, N., & Kwon, O.: A privacy-aware feature selection method for solving the personalization–privacy paradox in mobile wellness healthcare services. Expert systems with applications, 42(5), 2764-2771. (2015)
11. Zhang, X., Liu, S., Chen, X., Wang, L., Gao, B., & Zhu, Q.: Health information privacy concerns, antecedents, and information disclosure intention in online health communities. Information & Management, 55(4), 482-493. (2018)
12. PwC: The Wearable Future, Consumer Intelligence Series (2014). Available at: http://www.pwc.com/es_MX/mx/industrias/archivo/2014-11-pwc-the-wearable-future.pdf
13. Kurtz, C., Semmann, M. and Böhmann, T.: Privacy by Design to Comply with GDPR: A Review on Third-Party Data Processors presented at the Americas Conference on Information Systems (AMCIS), New Orleans. (2018)
14. Martin, Y. S., & Kung, A.: Methods and tools for GDPR compliance through privacy and data protection engineering. In 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (pp. 108-111). IEEE. (2018, April)
15. Alagar, V., Periyasamy K., and Wan, K.: Privacy and security for patient-centric elderly health care, 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, 2017, pp. 1-6. (2017)
16. Alibasa, M. J., Santos, M. R., Glozier, N., Harvey, S. B. and Calvo, R. A.: Designing a secure architecture for m-health applications, 2017 IEEE Life Sciences Conference (LSC), Sydney, NSW, 2017, pp. 91-94. (2017)
17. Zhou, J., Lin, X., Dong, X. and Cao, Z.: PSMPA: Patient Selfcontrollable and MultiLevel Privacy-Preserving Cooperative Authentication in Distributed-Healthcare Cloud Computing System, in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 6, pp. 1693 1703. (2015)
18. Volk, M., Sterle, J. and Sedlar, U.: Safety and Privacy Considerations for Mobile Application Design in Digital Healthcare. International Journal of Distributed Sensor Networks, 2015, pp.1-12. (2015)
19. Li, X.: Understanding eHealth literacy from a privacy perspective: eHealth literacy and digital privacy skills in American disadvantaged communities. American Behavioral Scientist, 62(10), 1431-1449. (2018)
20. Edemacu, K., Park, H. K., Jang, B., & Kim, J. W.: Privacy Provision in Collaborative Ehealth With Attribute-Based Encryption: Survey, Challenges and Future Directions. IEEE Access, 7, 89614-89636. (2019)
21. Omoogun, M., Seeam, P., Ramsurrun, V., Bellekens, X., & Seeam, A. (2017, June). When eHealth meets the internet of things: Pervasive security and privacy challenges. In 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security) (pp. 1-7). IEEE.
22. Bhuiyan, M. Z. A., Zaman, M., Wang, G., Wang, T., & Wu, J.: Privacy-protected data collection in wireless medical sensor networks. In 2017 International Conference on Networking, Architecture, and Storage (NAS) (pp. 1-2). IEEE. (2017, August)
23. Liu, L.S., Shih, P.C. and Hayes, G.R.: Barriers to the adoption and use of personal health record systems, Proceedings of the 2011 iConference, Seattle, WA, 8-11 February, ACM, pp. 363-370. (2011)
24. Romanou, A.: The necessity of the implementation of Privacy by Design in sectors where data protection concerns arise. Computer law & security review, 34(1), 99-110. (2018)
25. Sousa, M., Ferreira, D. N. G., Pereira, C. S., Bacelar, G., Frade, S., Pestana, O., & Correia, R. C.: OpenEHR Based Systems and the General Data Protection Regulation (GDPR). Building Continents of Knowledge in Oceans of Data: The Future of Co-Created EHealth. (2018)

26. Iwaya, L. H., Fischer-Hübner, S., Åhlfeldt, R. M., & Martucci, L. A.: mhealth: A privacy threat analysis for public health surveillance systems. In 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS) (pp. 42-47). IEEE. (2018, June)
27. Sawand, M. A., & Khan, N. A.: Privacy and Security Mechanisms for eHealth Monitoring Systems. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 8(4), 533-537. (2017)
28. Pirbhulal, S., Samuel, O. W., Wu, W., Sangaiah, A. K., & Li, G.: A joint resource-aware and medical data security framework for wearable healthcare systems. Future Generation Computer Systems, 95, 382-391. (2019)
29. Al-Janabi, S., Al-Shourbaji, I., Shojafar, M., & Shamshirband, S.: Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. Egyptian Informatics Journal, 18(2), 113-122. (2017)
30. Luo, E., Bhuiyan, M. Z. A., Wang, G., Rahman, M. A., Wu, J., & Atiquzzaman, M.: Privacyprotector: Privacy-protected patient data collection in IoT-based healthcare systems. IEEE Communications Magazine, 56(2), 163-168. (2018)
31. Cavoukian, A.: Privacy by design [leading edge]. IEEE Technology and Society Magazine, 31(4), 18-19. (2012)
32. Lambrinoudakis, C.: The General Data Protection Regulation (GDPR) Era: Ten Steps for Compliance of Data Processors and Data Controllers. In International Conference on Trust and Privacy in Digital Business (pp. 3-8). Springer, Cham. (2018, September)
33. Tikkinen-Piri, C., Rohunen, A., & Markkula, J.: EU General Data Protection Regulation: Changes and implications for personal data collecting companies. Computer Law & Security Review, 34(1), 134-153. (2018)
34. Huth, D.: A Pattern Catalog for GDPR Compliant Data Protection. In PoEM Doctoral Consortium (pp. 34-40). (2018)
35. Rubinstein, I.S. & Good, N.: Privacy by Design: A Counterfactual Analysis of Google and Facebook Privacy Incidents. Berkeley Technology Law Journal 28(2), 1333-1413. (2013)
36. Antignac, T., Scandariato, R., & Schneider, G.: Privacy compliance via model transformations. In 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (pp. 120-126). IEEE. (2018, April)
37. Rösch, D., Schuster, T., Waidelich, L., & Alpers, S.: Privacy Control Patterns for Compliant Application of GDPR. AMCIS 2019 Proceedings > Information Security and Privacy (SIGSEC) > 27. (2019)
38. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., & Robaldo, L.: Legal Ontology for Modelling GDPR Concepts and Norms. In JURIX (pp. 91-100). (2018, December)
39. Notario, N., Crespo, A., Martin, Y.S., Del Alamo, J.M., Metayer, D.L., Antignac, T., Kung, A., Kroener, I.,Wright, D.: PRIPARE: Integrating privacy best practices into a privacy engineering methodology. Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015 pp. 151-158. (2015)
40. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design : the PriS method. Requirements Engineering 13.3, 241-255. (2008)
41. Robol, M., Salnitri, M., & Giorgini, P.: Toward GDPR-compliant socio-technical systems: modeling language and reasoning framework. In IFIP Working Conference on The Practice of Enterprise Modeling (pp. 236-250). Springer, Cham. (2017, November)
42. Kalloniatis, C.: Incorporating Privacy in the Design of Cloud-Based Systems: A Conceptual Metamodel, Information and Computer Security Journal, Emerald. (2017) https://doi.org/10.1108/ICS-06-2016-0044
43. Kalloniatis, C.: Designing Privacy-Aware Systems in the Cloud, Proceedings of the TRUSTBUS 2015 12th International Conference on Trust Privacy and Security in

Digital Business, S. Hubner, C. Lambrinoudakis (eds), September 2015, Valencia, Spain, Springer LNCS Lecture Notes in Computer Science. (2015)

44. Mouratidis, H., Islam S., Kalloniatis C., Gritzalis S. A framework to support selection of cloud providers based on security and privacy requirements in Journal of Systems and Software Vol. 86, no 9, pp.2276-2293. (2013)
45. Carbonaro, G., Leanza, E., McCann, P., & Medda, F.: Demographic decline, population aging, and modern financial approaches to urban policy. International Regional Science Review, 41(2), 210-232. (2018)
46. Sharma, R.: The Demographics of Stagnation: Why People Matter for Economic Growth. Foreign Affairs 95 (2): 18–24. (2016)
47. Petrakaki, D., Hilberg, E., & Waring, J.: Between empowerment and self-discipline: Governing patients' conduct through technological self-care. Social Science & Medicine, 213, 146-153. (2018)

Christos Kalloniatis holds a PhD from the Department of Cultural Technology and Communication of the University of the Aegean and a master degree on Computer Science from the University of Essex, UK. Currently he is an Associate professor and head of the Department of Cultural Technology and Communication of the University of the Aegean and director of the Privacy Engineering and Social Informatics (PrivaSI) research laboratory. He is a former member of the board of the Hellenic Authority for Communication Security and Privacy. His main research interests are the elicitation, analysis and modelling of security and privacy requirements in traditional and cloud-based systems, the analysis and modelling of forensic-enabled systems and services, Privacy Enhancing Technologies and the design of Information System Security and Privacy in Cultural Informatics. He is an author of several refereed papers in international scientific journals and conferences and has served as a visiting professor in many European Institutions. Prior to his academic career he has served at various places on the Greek public sector including the North Aegean Region and Ministry of Interior, Decentraliastion and e-Governance. He has a close collaboration with the Laboratory of Information & Communication Systems Security of the University of the Aegean and the Systems Security Laboratory of the University of Piraeus. He has served as a member of various development and research projects.

Costas Lambrinoudakis holds a B.Sc. (Electrical and Electronic Engineering) from the University of Salford (1985), an M.Sc. (Control Systems) from the University of London (Imperial College -1986), and a Ph.D. (Computer Science) from the University of London (Queen Mary and Westfield College – 1991). Currently, he is a Professor at the Department of Digital Systems, University of Piraeus, Greece. From 1998 until 2009 he has held teaching position with the University of the Aegean, Department of Information and Communication Systems Engineering, Greece. For the period 2012-2015, he was a member of the board of the Hellenic Authority for Communication Security and Privacy, while from 2016 he serves on the board of the Hellenic Data Protection Authority. Finally, from 2015, he is Head of the Department of Digital Systems and Director of the Systems Security Lab. His current research interests are in the areas of Information and Communication Systems Security and of Privacy Enhancing Technologies. For many years he is working on issues related to the protection of personal data and the compliance of information systems to the National and European Legislation. He is the author of more than 100 scientific publications in refereed international journals, books and conferences, most of them on ICT security and privacy protection issues. He has served as program committee chair of 15 international scientific conferences and as a member on the program and organizing committees in more than 150 others. Also, he participates in the editorial board of two international scientific journals and he acts as a reviewer for more than 35 journals. He has been involved in many national and EU funded R&D projects in the area of Information and Communication Systems Security. He is a member of the ACM and the IEEE.

Mathias Musahl is working as Researcher in Body Sensor Network group of „Augmented Vision“ department, DFKI GmbH. He has finished his diploma in electrical engineering from TU Kaiserslautern in 2011. After that he worked for 6 years as a software engineer in the industry. There he worked on designing and implementing hardware and software for embedded devices related to distributed audio network infrastructure for the broadcasting industry.

Athanasios G. Kanatas is a Professor at the Department of Digital Systems, University of Piraeus, Greece, Director of the Telecommunication Systems Laboratory, and Director of the Postgraduate Programme in Digital Communications and Networks. He received the Diploma in Electrical Engineering from the National Technical University of Athens (1991), the M.Sc. degree in Satellite Communication Engineering from the University of Surrey, UK (1992), and the Ph.D. degree in Mobile Satellite Communications from NTUA (1997). He has published more than 200 papers in international journals and conference proceedings. He is the author of 6 books in the field of wireless and satellite communications. He has been the technical manager of several European and National R&D projects. His current research interests include the development of new waveforms and digital techniques for next generation wireless systems; wireless channel characterization and modeling; antenna design and security issues for V2V communications. He has been a Senior Member of IEEE since 2002. In 1999, he was elected Chairman of the Communications Society of the Greek Section of IEEE. From 2013 to 2017, he has served as Dean of ICT School of the University of Piraeus, Greece.

Stefanos Gritzalis is a Professor of Information and Communication Systems Security, at the Lab. of Systems Security, Dept. of Digital Systems, University of Piraeus, Greece (06.2019+). Previously, he was a Professor at the University of the Aegean, Greece, School of Engineering, Dept. of Information and Communication Systems Engineering, and member of the Info-Sec-Lab Laboratory of Information and Communication Systems Security (2002-2019). He was the Rector of the University of the Aegean, Greece (2014-2018), Head of the Dept. of Information and Communication Systems Engineering (2005-2009), Deputy Head of the Dept. of Information and Communication Systems Engineering (2012-2014), and Director of the Lab. of Information and Communication Systems Security (2005-2009). He has acted as Special Secretary for the Hellenic Ministry for Administrative Reform and Electronic Government (2009-2012). He holds a BSc in Physics, an MSc in Electronic Automation, and a PhD in Information and Communications Security from the Department of Informatics and

Telecommunications, University of Athens, Greece. His published scientific work includes more than 10 books, 33 book chapters (including the book “Digital Privacy: Theory, Technologies and Practices”, co-edited by A. Acquisti, S. Gritzalis, C. Lambrinoudakis and S. De Capitani di Vimercati, Auerbach Publications, Taylor and Francis Group). Moreover, his work has been published in 314 papers (133 in refereed journals and 181 in the proceedings of international refereed conferences and workshops). He has co-authored papers with more than 130 researchers from 25 countries during the last 28 years. The focus of his publications is on Information and Communications Security and Privacy.

Received: September 22, 2020; Accepted: December 20, 2020

Computer Science and Information Systems 18(1):349–378 https://doi.org/10.2298/CSIS201009058S

A JSSP Solution for Production Planning Optimization Combining Industrial Engineering and Evolutionary Algorithms

Sašo Sršen and Marjan Mernik

University of Maribor
Faculty of Electrical Engineering and Computer Science
Koroška cesta 46, 2000 Maribor, Slovenia
saso.srsen@student.um.si, marjan.mernik@um.si

Abstract. A Job Shop Scheduling Problem (JSSP), where *p* processes and *n* jobs should be processed on *m* machines so that the total completion time is minimal, is a well-known problem with many industrial applications. Many researchers focus on the JSSP classification and algorithms that address the different JSSP classes. In this research work, the production times, a very well-known theme covered in Industrial Engineering (IE), are integrated into an Evolutionary Algorithm (EA) to present a solution for real-world manufacturing JSSP problems solving. Since a drawback of classical IE is a manual determination of the technological times, an Internet of Things (IoT) architecture is proposed as a possible solution.

Keywords: JSSP, Genetic Algorithms, Evolutionary Algorithms, Industrial Engineering, Internet of Things

1. Introduction

The production planning/scheduling problem has been known for a very long time. Very often, we can find it under the term "Job Shop Scheduling Problem" (JSSP). The term first appeared in the 1950s, more specifically around 1954 [1]. JSSPs are generally known to belong to the group of the so-called non-deterministic problems, bound by polynomial-time hardness (NP or non-deterministic polynomial-time hardness). In practice this means that the time to calculate the optimal solution increases exponentially with the problem's size. JSSP is still considered to be one of the most challenging problems in terms of computation complexity today.

In the early research, several analytical techniques, like a branch and bound and heuristic approaches, have been proposed to solve the problem and deliver an optimal or near-optimal solution. With problem size growth (number of machines, jobs, processes), those approaches were not able to deliver the expected results anymore. Hence, more recently, the studies turned to other techniques, like simulation, Artificial Intelligence (AI) [38], and Evolutionary Algorithms (EAs) [26]. EAs are population-based search algorithms, which mimic concepts from biological evolution, such as survival of the fittest, crossover, and mutation. EAs are known to have a remarkable balance between exploration and exploitation [22] [51], which is needed to search an enormous space of all possible solutions efficiently [29], [21] [32]. Early examples of EAs are Genetic Algorithms (GAs), Evolutionary Strategies (ES), and Genetic Programming (GP) [26]. While recently, state-of-the-art metaheuristics are variants of Differential Evolution (DE) (e.g., jDE [20], SHADE [39]), and CMA-ES (IPOP-CMA-ES [7]). Algorithms that mimic problem-solving from nature are nowadays flourishing, and belong to a wider group of Swarm Intelligence (e.g., ABC [31] [34] [25]), or Computational Intelligence (e.g., TLBO [36], [23]). They are suitable for solving complex real-world problems [28], [37], [30], where the search space is simply too big to check all possible solutions.

Since 1985, when Davis [2] proposed the first GA based solution for the JSSP problem, a lot of research has been done to address the scheduling problem. The studies that followed developed different constraints, representations, and algorithms to classify, differentiate, and solve Shop Scheduling Problems (SSPs). In the recent review [9] fourteen classes of JSSP have been identified, based on their main characteristics: Job arrival process, inventory policy, duration time processing, and job attributes, as shown in Figure 1. We will explain the main characteristics of those 14 types in Section 2.2.

Although all of them are trying to solve the JSSP by arranging an optimal schedule with different goals (e.g., minimizing the makespan [4], completion time, the lateness of the due date, tardiness, throughput time), they still rely on a universal unit of measurement: Time. If we want to apply any JSSP solution in the real world, we need to clarify what kind of times should be used where and how to measure them successfully. Here, we rely on the relatively old knowledge from Industrial Engineering (IE) to define, measure and classify the production times precisely, as shown in Figure 1. That way, a "communication channel" between the real world and JSSP domain can be made. According to [8,10] three different approaches for production time determination exist: the actual data approach, the plan data approach, and the hybrid approach. In order to schedule production, any approach can be used, usually resulting in the implementation of a global standard methodology for time determination [8,10,11] (e.g. MTM, REFA, MOST, Work factor). If a manufacturing company product diversity is taken into account, the existence of a production time data database is a necessity and is usually covered by an "Enterprise Resource Planning" (ERP) System. Production management can access the data to schedule the production, but the traditional scheduling approaches (especially ad-hoc) very often don't achieve the desired results, possibly causing significant production efficiency decrease.

In this paper, we want to take advantage of the principles and methods found in IE [8,10,11] and combine them with an Evolutionary Algorithm (EA) JSSP approach in order to optimize and simplify the manufacturing scheduling process for production management by introducing a solution in the form of a tool.

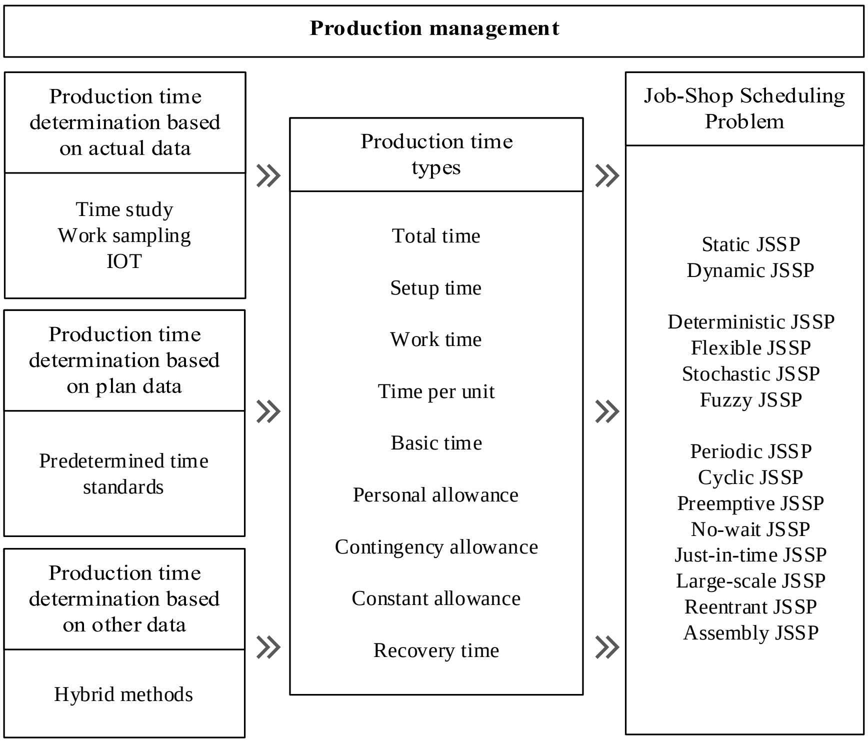


Fig. 1. Production time types and determination approaches, different JSSP classes

The main contributions of this research work are:

- Extending the JSSP with the time parameters found in manufacturing as defined by IE,
- Introducing an EA solution for a static flexible deterministic JSSP in the form of a tool,
- Proposing an IoT architecture to mitigate manual determination of technological times,
- Providing a use case with real-world data.

The rest of the paper is organized as follows. Section 2 covers the problem explanation and the time determination possibilities as per IE. In Section 3, we explain the proposed approach. Section 4 displays the solution use case example, problems, and results. Section 5 concludes with a summary of this work, adding some possible future research possibilities.

2. The proposed approach

2.1. JSSP description

How can we explain what the basic JSSP is? Informally, the problem could be described as follows: We have a set of jobs and a set of machines. Each job consists of a sequence of continuously performed processes for a specific time on a particular machine. Each machine can complete only one process at a time. The "schedule" represents the occupancy of machines with processes at specific time intervals. The key problem for this situation is creating a schedule where the finish time of the final process in the schedule is minimal. In general, the problem could be described formally as follows. Let the finite set M represent the set of all machines, and the finite set J represent the set of all jobs:

$$M = \{M_1, M_2, M_3, \dots M_m\} \tag{1}$$

and

$$J = \{J_1, J_2, J_3, \dots J_n\}. \tag{2}$$

If each job needs to be processed on all machines, but only once on each machine, the set representing the job sequence per machine could be written as a matrix x of size $m\ x\ n$. For example:

$$x = \begin{pmatrix} 4 & 2 & 3 & 1 \\ 2 & 1 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}. \tag{3}$$

Each row in the matrix represents a job order for the machine $M_1, M_2, M_3, \dots M_m$. The above matrix can therefore be read as: Machine processes on the machine M_1 will be performed in the sequence: J_4,J_2,J_3,J_1 processes, on the machine M_2 will be performed in the sequence: J_2,J_1,J_4,J_3, and processes on the machine M_3 will be performed in the sequence: J_1,J_2,J_3,J_4 . We can quickly conclude that the matrix x is only one element of a broader set (let's call it, for example, the set X) of all possible combinatorial variants, i.e., $x \in$ X.

If we want to search for an optimal schedule, we need a general estimation function f that can calculate the exact "value" for each matrix $x \in$ X:

$$f: \mathrm{X} \rightarrow [0, +\infty] \tag{4}$$

or, if we look more precisely, for each element of the matrix:

$$f_{ij}: J \times M \rightarrow [0, +\infty]\ . \tag{5}$$

Throughout the paper, we will use the index i for jobs and index j for machines. Because JSSP algorithms are used to optimize the time required, a common output of the estimation function represents the total execution time (also called timespan or makespan). Other possible function outputs include, and are not limited to, flow time (total weighted completion times) and lateness or tardiness (with a due date) [9]. The f_{ij} function, therefore, calculates the total execution time of the job J_i on the machine M_j. The JSSP solution is, therefore, a matrix $x \in$ X, where the makespan $f(x\ )$ for completing all the tasks (or jobs) is minimal, or that there is no known $y \in$ X where $f(x\ ) > f(y\ )$. In other words, the solution of JSSP is to find a schedule where:

- simultaneous processing of multiple jobs on the same machine is not possible,
- the same job cannot be processed simultaneously on multiple machines,

- each operation for an individual job occupies each machine for a specific time T, and
- the makespan is minimal.

2.2. Related work, JSSP classes and types

Although we have limited ourselves to the JSSP (Figure 1), we should first explain that, in general, three basic Shop Scheduling Problems (SSP) types exist [3]:

- Flow Shop Scheduling Problem (FSSP),
- Job Shop Scheduling Problem (JSSP), and
- Open Shop Scheduling Problem (OSSP).

The problem that needs to be solved for all the above-mentioned Shop Scheduling Problems is the same; the only difference is their limitations. In the case of FSSP, each job has exactly the same number of machine processes, and the sequence of machine processes for each job is predefined and the same, for example:

$$\begin{aligned} J_1 &: M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \\ J_2 &: M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \\ J_3 &: M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4. \end{aligned} \tag{6}$$

The possible sequences are usually limited by the technological process, and could be written for every job as: J_i: $M_1 \rightarrow M_2 \rightarrow \ldots \rightarrow M_j$, where $1 \leq j \leq m$ and $1 \leq i \leq n$. As already explained, the solution to the problem lies in finding a machine sequence where the estimation function (makespan) for completing all jobs is minimal.

In contrast to FSSP, the JSSP machine sequences are also limited by technology and known in advance, but they can vary, for instance:

$$\begin{aligned} J_1 &: M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \\ J_2 &: M_2 \rightarrow M_1 \rightarrow M_4 \rightarrow M_3 \\ J_3 &: M_1 \rightarrow M_2 \rightarrow M_4 \rightarrow M_3. \end{aligned} \tag{7}$$

Thus, for each job, we can write the technological machine sequence for a specific job J_i as $M_{i1} \rightarrow M_{i2} \rightarrow M_{i3} \ldots \rightarrow M_{i4}$, where $1 \leq j \leq m$ and $1 \leq i \leq n$. It should be emphasized that, in the case of simple JSSPs, we assume that the number of processes for each job is equal to m or the number of machines (i.e., each job "travels" through all machines exactly once). In real life, however, often there is a situation where this number is less than m, meaning that each job does not need to be processed by all machines. Another case is where the number of processes exceeds m, resulting in repeating the machine process on an operation multiple times.

In the case of OSSP, the sequences of machine processes aren't predefined. It is often assumed that the number of machine processes for a job is equal to m, meaning that all machine processes must be completed for each job. We have to emphasize that OSSP occurrence is extremely rare in the real world.

If we look at JSSP, we can see many different types in Figure 1. We can classify them further by different criteria: job arrival criteria, time parameter criteria, and other criteria.

Using job arrival criteria two types of JSSP can be defined:

- Static JSSP, and
- Dynamic JSSP.

For static JSSPs, a finite number of jobs are ready for processing on a finite number of machines at the time zero [40]. An unexpected event occurrence is not possible.

Dynamic JSSPs are similar, except the job occurrence is random [3]. In both cases, the order of precedence of operations and processing times are predefined.

Using time parameter criteria, several types of JSSP can be defined:

- Deterministic JSSP,
- Flexible JSSP,
- Stochastic JSSP, and
- Fuzzy JSSP.

If the processing time for every operation of job j on every machine m is known in advance and the operation sequence order is predefined, we can classify it as a deterministic JSSP (also called a crisp JSSP) [42]. The flexible JSSP extends the deterministic JSSP by allowing a machine operation to be processed by one machine out of a set of machines, thereby adding the problem of assigning each operation to a specific machine (routing) [41]. Stochastic JSSPs introduce parameters dealing with probability conditions, for instance, machine breakdown or processing time [16]. Since real-world JSSP times often don't have deterministic value, fuzzy values (processing times, due date, ranking) have been incorporated into JSSP, hence the name fuzzy JSSP [43].

Using other criteria, further types of JSSP can be defined:

- Periodic JSSP,
- Cyclic JSSP,
- Preemptive JSSP,
- No-wait JSSP,
- Just-in-time JSSP,
- Large-scale JSSP,
- Reentrant JSSP, and
- Assembly JSSP.

The periodic JSSP is an iterative version of the JSSP where a batch of size n of each job is processed iteratively [44]. The cyclic JSSP deals with a set of process operations that cycle an indefinite number of times by minimizing the period length [45]. In case the algorithm allows the interruption of an operation during processing on a specific machine and to continue at a later time, we're talking about pre-emptive JSSP [46]. The no-wait JSSP introduces the no-wait constraint between two sequential operations by delaying the job starting time at the first machine operation [47]. The just-in-time JSSP is solving the earliness-tardiness problem of jobs by penalizing both options [48]. The large-scale JSSP approach can be used when huge numbers of machines and jobs are required [49]. The Reentrant JSSP extends a deterministic JSSP, where a job operation may be repeated multiple times [50]. The assembly JSSP extends the JSSP by appending an assembly stage and introducing lot streaming (LS) thereby splitting the job into smaller batches and taking away job independence.

Many other subtypes exist, many of them extending the basic types with different constraints and Objective Functions, for instance, machine blocking constraints [17]. Recent research even covers the so-called "low-carbon" JSSPs by pursuing the goal to minimize the sum of completion time cost and energy consumption cost [18].

2.3. Time in Industrial Engineering (IE)

Since we'll be using production time as a parameter for the JSSP solution, we must take a look at how IE is determining and structuring production time. According to REFA [8], similar to Seifermann [10], when we need to determine a time for specific work or work part on an operational level, different approaches exist, as shown in Figure 2:

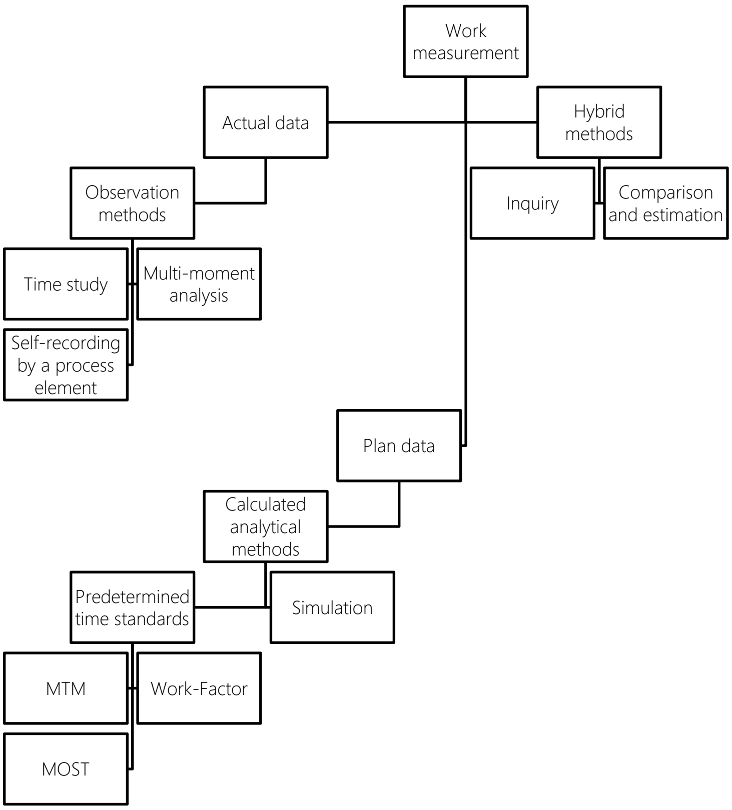


Fig. 2. Overview of different IE methods for time determination

The actual data approach requires the presence of an analyst in the workplace for work observation and measurement. In contrast, the plan data approach just requires a detailed work process analysis for work time determination. The hybrid approach combines techniques from the mentioned ones.

We should emphasize that the times that need to be determined for JSSP use are usually called "target times" ("Sollzeit") [8] or norm times. They often represent the foundation for different manufacturing divisions, like production planning and management, costing, controlling, and remuneration. Some of the mentioned divisions require another type of time called "actual time" ("Istzeit") [8] for their work, that represents the actual spent amount of time that has been used to complete a specific job.

According to REFA [8][14], the following applies:

$$T = t_r + t_a \,. \tag{8}$$

The total target time T is divided into setup time (t_r, "Rüstzeit") and work time (t_a, "Arbeitszeit"). The setup (also called changeover) times are quantity independent, and can be defined as:

- Fixed, for a specific job/machine, and
- Variable or sequence-dependent, for a specific job/machine.

In serial production, work time t_a can be written as:

$$T = t_r + \mathrm{m} * t_e \ . \tag{9}$$

Whereas quantity m ("Menge") stands for the total quantity of products required for a specific operation and/or job, depending on the level being used. The variable called t_e ("Einzelzeit") stands for time per unit, and defines the target time required to manufacture/process one unit of the product (liter, kg, meter, piece). The setup time is not quantity dependent, as displayed in Figure 3:

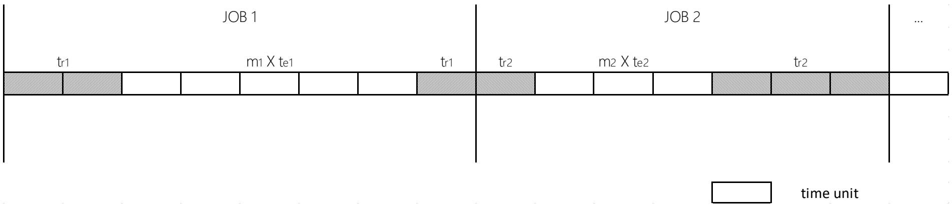


Fig. 3. Production timeline example

Each time we swap the product (or change the job) on a machine, the setup time usually occurs at the beginning and/or at the end of the process. The processes that require setup times can start before the previous process step in the job has finished. As shown in the example in Figure 4, the setup time t_r is one unit long. The total time T for job 3 on machine 3 is actually two units, but since machine 3 is IDLE before job 3 on machine 4 is finished, the setup on machine 3 can start.

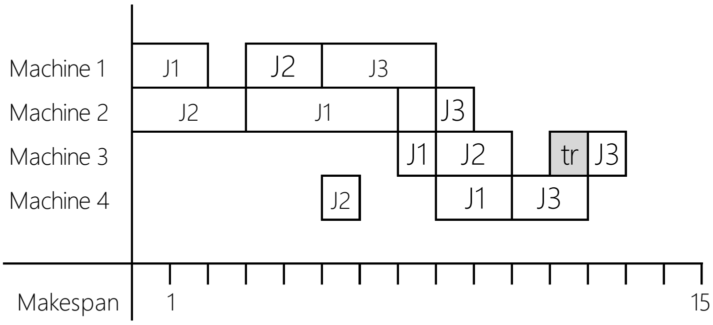


Fig. 4. Setup time

Time per unit t_e gets divided further into three parts, namely:

$$T = t_r + \mathrm{m} * (t_g + t_v + t_{er}) \ . \tag{10}$$

The first variable in the equation, basic time t_g ("Grundzeit") represents the time of a bare machine or manual work or a combination of them. The second variable, t_v ("Verteilzeit") or allowances represent a percentage of smaller interruptions/disturbances in the work process that occur randomly that gets added to the basic time [15]. Those disturbances can be divided further into contingency allowances t_{vsv}, personal allowances t_{vp} and special/constant allowances t_{vsk}. Contingency allowances cover short stochastic process delays like machine breakdown or raw material shortage, for example. Personal allowances cover personal needs like toilets or drinking fluids. Special or constant allowances are usually given per work period, and cover the time needed for work that isn't bound directly to any work order, like cleaning at the end of the shift or tooling at the beginning of the work. Allowances can be defined per product/process/machine, but, usually, they are defined on a higher level, for example, a group of workplaces or a sector, or even for the entire production plant. The third variable, t_{er} ("Erholungszeit") or relaxation allowances, occur only in harsh work conditions (heat, radioactive environment, etc.) and, similar to the allowances, raise the basic time by a certain percent to compensate for the delegated breaks.

A specific approach for determination can be used for every time component mentioned. To determine basic times, usually an observation methodology is used, like time study, or a predetermined time system (MTM, Work factor or MOST).

Time study

A time study approach requires the use of a chronometer, and is completed by an analyst, who is present while the work is being executed. The process can be divided into four phases:

- Work (place) analysis and phases definition,
- On-site work measurement (using a chronometer), performance rating,
- Time study analysis, and
- Reporting and data updating.

The measurement usually requires multiple cycles (or samples) to get reliable data.

Predetermined time systems

In contrast, predetermined time systems instead use motion study as the foundation [11]. There is no need for on-site presence if you are able to break down the work into single standardized motion elements, the building blocks of a predetermined time system.

Only basic time is determined by summing times for all identified motion elements using matrices on cards, as defined by the Standard (MTM, MOST or Work factor). Table 1 displays the standard MTM-1 motions. By combining basic motion elements (MTM-1, for example), higher-level motion elements can be defined (MTM-UAS or MTM-MEK, for example).

Table 1. MTM-1 standard motion elements [13]

Hand/arm motion elements	**Eye motion elements**	**Body, leg and foot motions**
Reach (R)	Eye travel (ET)	Foot motion (FM)
Grasp (G)	Eye focus (EF)	Leg motion (LM)
Release (RL)		Sidestep (SS)
Move (M)		Turn body (TB)
Position (P)		Walk (W)
Apply pressure (AP)		Bend (B), Arise from bend (AB)
Disengage (D)		Stoop (S), Arise from stoop (AS)
Turn (T)		Kneel on one knee (KOK), Arise from kneeling on one knee (AKOK)
		Kneel on both knees (KBK), Arise from kneeling on both knees (AKBK)
		Sit (SIT), Stand (STD)

Allowances

Work sampling (Multimoment analysis) or long-term time study (LTTS) can be used for determining allowances. Since we're determining stochastic events` frequency and duration, a much larger sample size is required compared to the classical time study.

If we want to extend the technology matrix (e.g. from Tables 3 or 4) with the time definitions, as explained earlier in this Section, we should define the time for each cell in the technology matrix as:

$$T_{ik} = t_{r_{ik}} + \mathrm{m} * (t_{g_{ik}} + t_{v_{ik}} + t_{er_{ik}}) \ . \tag{11}$$

where $1 \leq i \leq m$ and $1 \leq k \leq p$.

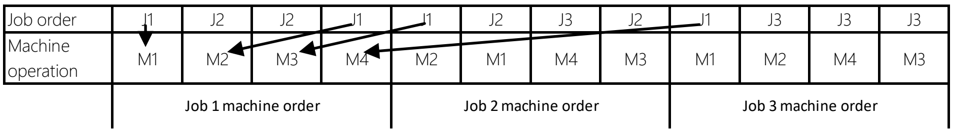
3. Solution implementation

According to Section 2.2, the presented approach could be classified as a static flexible JSSP solution, meaning that we have all the jobs ready at time zero. All the processing times are predefined and for each operation, and the solution can choose a machine out of a set of machines. The number of processes p is not necessarily equal to m machines, with the meaning:

- If $p < m$: The job doesn't have to visit every machine, and
- If $\mathrm{p} > m$: The job re-enters a specific machine as defined by the technology matrix.

Chromosome representation

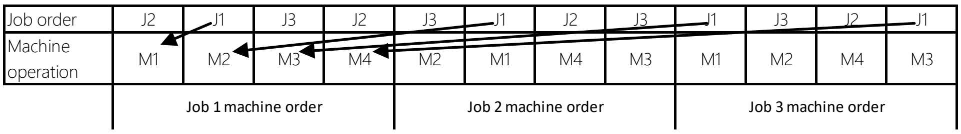
When implementing an EA solution, the first problem is the representation of an individual in the form of a chromosome [26]. We chose to use the unpartitioned permutation with m-repetitions representation [5][12]. Every job J (out of n jobs) consists of p processes that we have to schedule on m machines, thus giving us a two-row chromosome with the size of $n * p$ elements N, as shown in Figure 5:



Job order	J1	J2	J2	J1	J1	J2	J3	J2	J1	J3	J3	J3
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Fig. 5. Chromosome representation

In this formulation, each job J_j appears exactly p times (number of defined processes) in the first row of the chromosome, while the second row specifies the process sequence for a specific job, consisting of p elements (machines). When scanning the job order from left to right, each job iteration increases the machine operation index for that job by 1 (Figure 5). Permutation, in this case, only means a change in the order in which the job processes will be performed, as shown in Figure 6:



Job order	J2	J1	J3	J2	J3	J1	J2	J3	J1	J3	J2	J1
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Fig. 6. Schedule permutation example

Table 2. Predefined machine order with times` example

Operation **Job**	**1.**	**2.**	**3.**	**4.**
J1	M1, 2	M2, 4	M3, 1	M4, 2
J2	M2, 3	M1, 2	M4, 1	M3, 2
J3	M1, 3	M2, 1	M4, 2	M3, 1

Usually, gene J1 with occurrence index 1 in the chromosome means that the processing (if possible) must begin on machine 1, the next occurrence of this index implies that the processing must begin on machine 2, the next occurrence on machine 3, etc. as defined by the Job 1 machine order.

A job/machine is usually defined as a fixed Table value (Table 2), meaning that the processing for job 1 on machine 1 lasts two units, on machine 2 four units, etc. Since we classified the solution as flexible, rather than using Table 2, we defined another Table, named the technology matrix, for each job J consisting of p processes on m machines. Table 3 shows an example of a technology matrix filled with times for job J1 from Table 2. The Table itself introduces the flexibility by letting the solution choose between different machines for the same process, in case we have multiple machines available, as shown in Table 4. Process 1 for Job 1 can be completed on machine 1 lasting two units, or on machine 3 lasting three units, and process 2 can be completed on machine 2 or 3 (4 and 5 time units). In table 3, we also demonstrated the option where the number of processes exceeds the number of machines ($p > m$). If $p > m$ for any job, the chromosome size raises to $n *$ max (p). In case the number of processes for a specific job is lower than the number of machines ($p < m$), the unused processes simply get a value of 0 for any machine.

Table 3. Technology matrix for job J1 with fixed machines and times as per the example

	M1	M2	M3	M4
P1	2	X	X	X
P2	X	4	X	X
P3	X	X	1	X
P4	X	X	X	2

Table 4: Technology matrix for job J1 with machine options and $p > m$:

	M1	M2	M3	M4
P1	2	X	3	X
P2	X	4	5	X
P3	X	X	1	X
P4	X	X	X	2
P5	X	3	X	X

Because the 2nd row of the chromosome representation from Figure 5 still can't change (static JSSP), we can calculate the search space for the chromosome upper part for the JSSP representation as:

$$\frac{(p\ x\ n)!}{(p!)^n}\,. \tag{12}$$

Formula (12) is based on permutations with repetition and because the number of processes is the same for all jobs, the denominator has the exponent n.

If we use the formula in our example from Table 1, the search space is limited by 34,650 different chromosomes. If we extend the static JSSP to flexible JSSP, where we define a matrix with different machine options (routing) for each job process, we can extend formula (12) to:

$$\frac{(p\ x\ n)!}{(p!)^n}\ x \prod\nolimits_{i=1}^{n} \prod\nolimits_{k=1}^{p} count(possible\ machines)_{(ik)} \tag{13}$$

meaning that the number of feasible chromosomes will grow because now even the 2nd row of the chromosome (machine sequence) can change accordingly. By using formula (13), our example search space (let's presume the technology matrix from Table 4 is the same for all 3 jobs) grows to 2,217,600 different chromosomes. If we presume that all technology matrices are full (worst-case scenario, there are no infeasible solutions, every process can be completed on every machine), we can define the search space as:

$$\frac{(p\ x\ n)!}{(p!)^n}\ x\ m^{p\ x\ n}\,. \tag{14}$$

This gives us a search space of 581,330,534,400 different chromosomes for our example.

The chromosome phenotype [26] can be represented by using a Gantt chart (Figure 7):

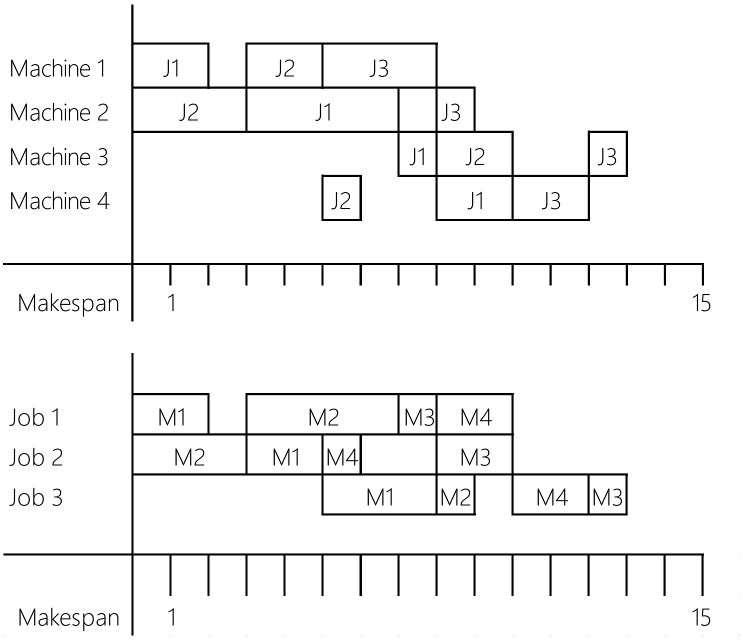


Fig. 7. Phenotype representations of Table 2 using a Gantt diagram *(machine/job)* and *(job/machine)* representation

Or, if we want to use the matrix representation (3), we could write:

$$x = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}. \tag{15}$$

If a job needs fewer processes than p to finish, the excessive processes get machine processing times with a duration of 0. If a process can not be completed on a specific machine as per the technology matrix (that chromosome represents an infeasible solution), that individual gets a "bad" fitness value which may, possibly, drive it out of the population.

Initial population

In the current solution a random schedule population is generated with defined parameters: Number of jobs, number of machines, number of processes. A technology matrix with the size of $p\ x\ m$ is required for each job J_i.

Selection operator(s)

Two selection operators have been implemented: Tournament selection and roulette wheel selection [26]. The tournament selection picks a group of a specific size randomly out of the population and orders the group according to the chromosomes' fitness values. The best individual is selected as one of the parents. The roulette wheel uses the individual fitness value (makespan) and normalizes it to 1 by dividing it by the total fitness of all individuals in the group, thus defining the probability of selection.

Crossover operators

Four different crossover operators have been implemented: Single point, two point, uniform, and ordered crossover operators [26]. For single point crossover, a random index k is selected between 1/4 and 3/4 of the chromosome size N. Then the first k genes are copied from parent 1 and the rest from parent 2, starting at gene $k + 1$, considering every job can only occur p times in every chromosome, skipping the gene otherwise. When the copy index reaches N, we continue at the beginning of the 2nd parent chromosome (Figure 8).

The two-point genetic operator defines two indexes, the starting index k and ending index l. They both get selected randomly, then the genes between index k and l get copied from parent 2. The next step is to copy the genes from parent 1, where the occurrence count of a specific job in the gene of the chromosome doesn't exceed the number of processes p. Finally, the empty spaces are filled with copies of genes from the second parent, but in an order in which they appear in the second parent after the ending index l (Figure 9).

The uniform operator works very simply; it's flipping a coin for every gene, to decide whether the offspring will contain the gene from parent 1 or 2, starting at the beginning of the chromosome and counting the occurrence of each job in the gene. When the job occurrence count for the chosen job reaches p, we try to insert the gene from the other parent, and if the occurrence count of that gene hasn't reached p, we copy it to the offspring. Otherwise, we skip the gene. After the index reaches the chromosome size N, we copy the remaining missing genes from parent 2, starting with the first gene.

Parent 1

Job order	J1	J2	J2	J1	J1	J2	J3	J2	J1	J3	J3	J3
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Parent 2

Job order	J2	J1	J3	J2	J3	J1	J2	J3	J1	J3	J2	J1
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Offspring

Job order	J1	J2	J2	J1	J3	J1	J2	J3	J1	J3	J2	J3
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Fig. 8. Single point crossover operator (Random position k = 4, chromosome size N = 12)

Parent 1

Job order	J1	J2	J2	J1	J1	J2	J3	J2	J1	J3	J3	J3
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Parent 2

Job order	J2	J1	J3	J2	J3	J1	J2	J3	J1	J3	J2	J1
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Offspring

Job order	J1	J2	J2	J1	J3	J1	J2	J3	J1	J3	J3	J2
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M4
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Fig. 9. Two-point crossover operator (Random *position k = 4, l = 9, chromosome size N = 12*)

The ordered crossover operator (OX, Figure 10) is very similar to the two-point genetic operator, except that, after copying the genes between the starting index k and ending index l, the rest is copied from the first parent, starting at the index l, and skipping values where the occurrence count of a specific gene exceeds the number of processes p.

Parent 1

Job order	J1	J2	J2	J1	J1	J2	J3	J2	J1	J3	J3	J3
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Parent 2

Job order	J2	J1	J3	J2	J3	J1	J2	J3	J1	J3	J2	J1
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Offspring

Job order	J2	J2	J1	J2	J3	J1	J2	J3	J1	J3	J3	J1
Machine operation	M2	M3	M4	M1	M2	M1	M4	M3	M1	M2	M4	M1
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Fig. 10. Ordered crossover operator (OX)

Mutation operators

Two widespread mutation operators have been implemented: Exchange values and change values (Figure 11). The change value operator first selects a gene in the chromosome randomly, then changes the machine in the gene to a random machine n, where $n \in M$:

Original

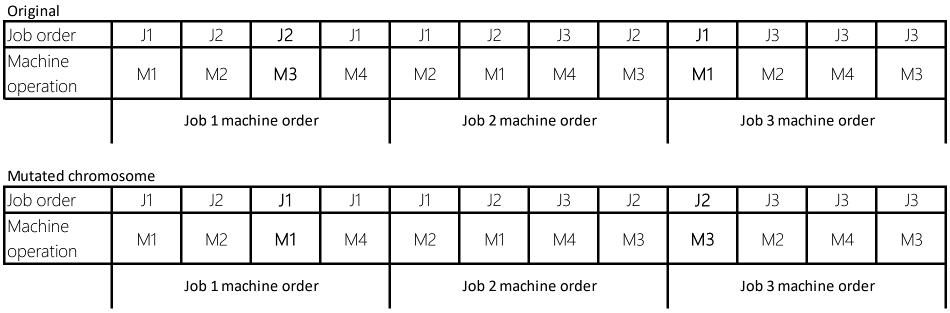
Job order	J1	J2	J2	J1	J1	J2	J3	J2	J1	J3	J3	J3
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Mutated chromosome

Job order	J1	J2	J2	J1	J1	J2	J3	J2	J1	J3	J3	J3
Machine operation	M1	M2	M1	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Fig. 11. Change value mutation operator

The change value operator cannot affect the job order because of the occurrence limitation. The exchange value operator selects two random genes and switches the job order and machine values (Figure 12).



Original

Job order	J1	J2	J2	J1	J1	J2	J3	J2	J1	J3	J3	J3
Machine operation	M1	M2	M3	M4	M2	M1	M4	M3	M1	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Mutated chromosome

Job order	J1	J2	J1	J1	J1	J2	J3	J2	J2	J3	J3	J3
Machine operation	M1	M2	M1	M4	M2	M1	M4	M3	M3	M2	M4	M3
	Job 1 machine order				Job 2 machine order				Job 3 machine order			

Fig. 12. Exchange values mutation operator

Fitness function

While building the solution, we focused on a static deterministic production, meaning that the process times are known, and the jobs are ready at time zero. As already mentioned, we chose makespan as the fitness function for the implementation. Makespan can be written as f_{max} and represents the time when the last operation process is completed [6]:

$$f_{max} = \max(f_1, f_2, \dots, f_n) \tag{16}$$

where:

$$f_j = \sum_{k=1}^{p} \left(W_{jk} + T_{mj(k)}\right). \tag{17}$$

Variable f_j stands for job j completion time, W_{jk} stands for waiting (or IDLE) time of job j at sequence k, and $T_{mj(k)}$ stands for the processing time for job j on machine m at sequence k.

Algorithm

The pseudocode of the proposed GA is shown in Figure 13. At first, a random population of a predefined size is generated and makespans are calculated, the best one noted. Then, in a predefined loop of size MaxIterations, we use the chosen selection to select a group of chromosomes and perform a crossover with the two best parents in the group. Afterward, we apply the mutation operator to the two children. The two worst individuals in the group get substituted by the two children, the best makespan gets checked. We apply another mutation on a random chromosome in the population and recheck if the best makespan changed. If 1/3 of the population went through the loop and the best makespan hasn't changed, we inject a % of fresh random chromosomes into the population.

The algorithm can be improved further using Long Term Memory Assistance (LTMA) [24], where duplicate solutions are identified. As such, time-consuming fitness evaluation is spared.

All the algorithm parameters can be changed by the user directly in the tool, like population size, selection group size, mutation probability pm, terminal condition MaxIteration, selection type, crossover-type, mutation type, and % of random chromosome injection if the fitness function didn't evaluate any better solution for 1/3 of the MaxIteration. The crossover probability pc is set to 1.

```
createInitialPopulation(population size)
calculateMakeSpans()
while (I < MaxIteration)
{
     doSelection(group size)
     doCrossover(best parent1, best parent2)
     doMutation(best child1, best child2)
     replace(worst parent1, worst parent2) with (best child1, best child2)
     checkBestMakeSpanChanged(best child1, best child2)
     doMutation(random chromosome)
     checkBestMakeSpanChanged(mutated chromosome)
     if (bestMakeSpan has not changed after 1/3 of MaxIteration)
     {
        injectNewRandomChromosomes(a % of population size)
        reset MakeSpanChanged counter
     }
}
```

Fig. 13. Genetic algorithm pseudocode

3.1. IoT

A potential approach in determining production times could also be by implementing a solution based on the Internet of Things (IoT) technology [35][19]. The goal of using IoT is to minimize or completely eliminate the need for human intervention in actual and target time data gathering, e.g., using IR scanners or terminals.

A sample IoT architecture for such purpose can be seen in Figure 14. Because the use case in the next Section was completed in a shoe factory, we will explain the working principle for the latter. The workers are using trolleys to transport upper shoes from one machine to another. Currently, each trolley receives a unique job (work order) barcode, so the worker can scan this barcode and a barcode on the machine to signal the beginning or end of a work process. This way, the ERP system can track the job completion status at the process level. However, tracking time and status in the explained way requires a high amount of discipline among the workers, meaning that any delay or mistake (e.g., forgetting to register at the beginning, or at the end of the process) in the registration can potentially produce unexplainable errors in the data interpretation. IoT use minimizes the registration mistake possibility.

Figure 14 explains a different approach proposal. The trolley must "know" which job it is carrying, so the Production Manager must somehow provide the ERP system with that information before the trolley is launched (e.g. by using a barcode or Radio-Frequency Identification (RFID) technology). The trolley must be equipped with a small computer, System on Chip (SoC), ESP8266 used in the example, and different sensors. In our case, the trolley is also equipped with two load cells, an RFID scanner, an accelerometer, and a gyroscope. When the worker drives the trolley around the shop, the accelerometer and gyroscope sensors detect movement and send the movement time data across WiFi to the Message Queueing Telemetry Transport (MQTT) broker using the MQTT protocol. The time resolution and data amount require the use of a No-SQL

Database (e.g., MongoDB) to store the streamed data. When the worker stops in front of a specific machine to start working on a job, the trolley has to stand still in a specific place for a certain amount of time. RFID tags on the reserved trolley position should be used to bind the machine ID to the current job ID. Again, the time data should be sent to the broker as soon as the sensor recognizes the RFID tag, and also when it leaves the reserved position. While the worker is completing the job on the machine, the two load cells stream the trolley weight data to the MQTT broker. Because only streaming the time data to a database still wouldn't provide input to the JSSP solution, an intelligent service is required to match and identify production events described above and allocate the time data to a specific process and define the type of time.

Using IoT as a means of determining production times opens a new perspective, not only for the JSSP solutions:

- Automatic target times` actualization based on chronological time data: Basic times, setup times and allowances in the ERP and/or other services like a JSSP tool,
- Delegation of short-term delayable allowance events (e.g., filling containers with very fine material, non-urgent cutting tool change, personal needs, etc.),
- Scheduling progress monitoring & dynamic re-launch of the JSSP search in case of unforeseen schedule deviations (e.g. longer machine breakdowns, unexplainable long delays, etc.).

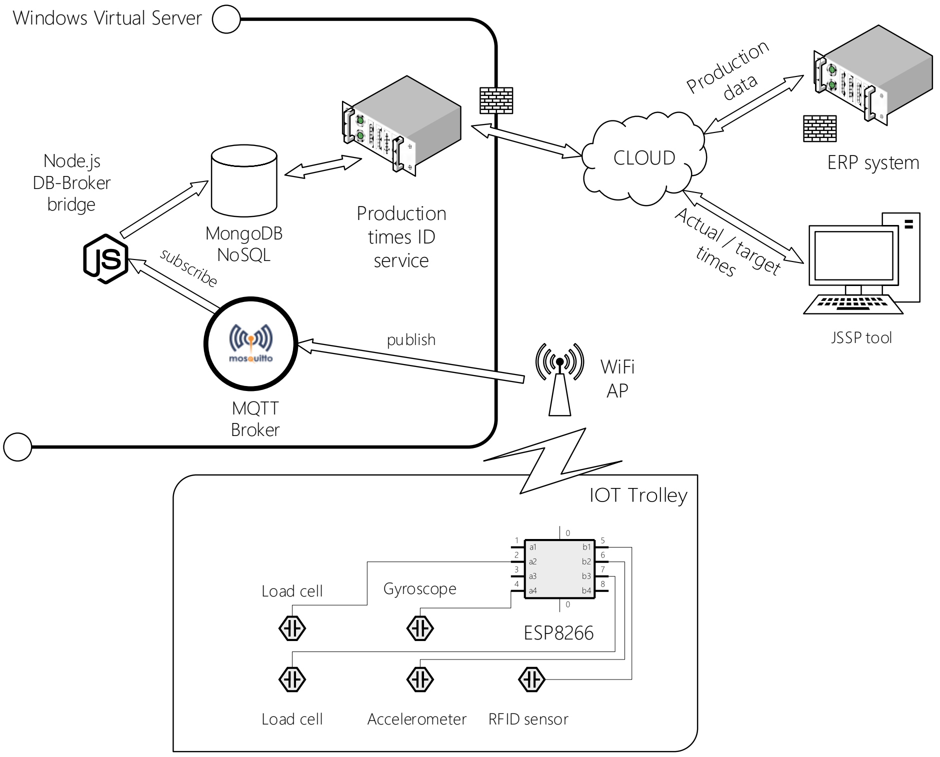


Fig. 14. An IoT production time determination architecture proposal

4. Use case

The implementation was made with a specific goal in mind, to optimize daily production scheduling/job launch for a shoe company. The shoe company has many production sectors, but for the use case, we focused on only a specific one, the upper shoe production sector. The problem that occurred is that the company was not able to define a daily job schedule with a predictable outcome. Additionally, the information about jobs and quantities for the upcoming day are usually defined at the end of the shift. Usually, the Production Managers were the ones delegating the work according to their past experience.

Papers and journals often use the term "machines" in JSSP, but, in practice, we can generalize the meaning of "resource." This small alteration is beneficial, since, in general, the term is a useful definition for manual workplaces as well as machines or machine types or groups; the principles described in Section 3.3 are valid for all. If we look at the parameters in Table 5 for a specific production request at the beginning of the week we got from the company:

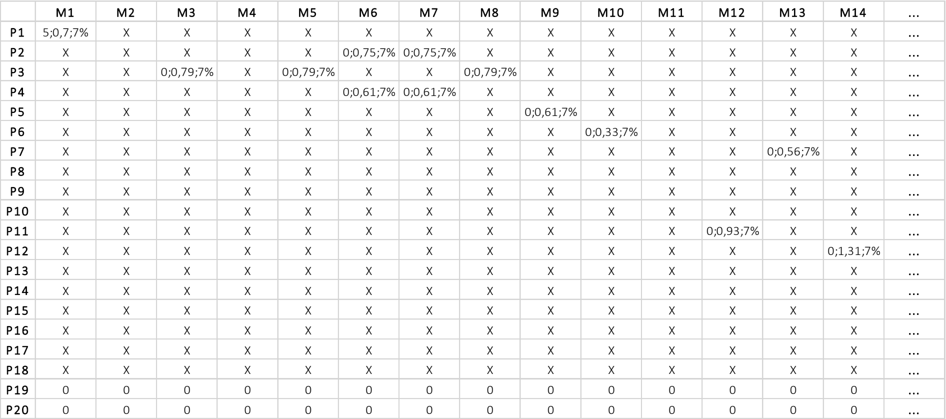
Table 5. JSSP / time parameters

No. of different resources m … **38**
Max. no. of processes p_{max} … **20**
No. of jobs/shift… **7**
Basic times t_g … defined per resource / process / job

Setup times t_r … fixed per resource, only M1 and M38 needing 5 minutes
Allowances t_v … **7% (**fixed for all resources / jobs)
Relaxation allowances t_{er} … 0% (normal working conditions)

Job	m (pcs required)
1	20
2	280
3	140
4	140
5	150
6	50
7	100
sum	880

Immediately, we can see the dimensions of the real-world problem. The total number of pieces (pairs) of shoes was 880, the task was split into seven jobs. Each job technology matrix contained 760 different matrix cells (possible process times) per job, meaning that each job contained a maximum of twenty processes that could be performed on 38 resources, as shown in Figure 15:



	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	...
P1	5;0,7;7%	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P2	X	X	X	X	X	0;0,75;7%	0;0,75;7%	X	X	X	X	X	X	X	...
P3	X	X	0;0,79;7%	X	0;0,79;7%	X	X	0;0,79;7%	X	X	X	X	X	X	...
P4	X	X	X	X	X	0;0,61;7%	0;0,61;7%	X	X	X	X	X	X	X	...
P5	X	X	X	X	X	X	X	X	0;0,61;7%	X	X	X	X	X	...
P6	X	X	X	X	X	X	X	X	X	0;0,33;7%	X	X	X	X	...
P7	X	X	X	X	X	X	X	X	X	X	X	X	0;0,56;7%	X	...
P8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P11	X	X	X	X	X	X	X	X	X	X	X	0;0,93;7%	X	X	...
P12	X	X	X	X	X	X	X	X	X	X	X	X	X	0;1,31;7%	...
P13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P14	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P15	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P17	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P18	X	X	X	X	X	X	X	X	X	X	X	X	X	X	...
P19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
P20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Fig. 15. Job 1 technology matrix for the first 20 machines as defined by the product technology

For each row (or process), we calculated the time $\mathrm{m} * t_e$ that is needed for a specific resource, in order to complete the job quantity m. Figure 15 displays the technology matrix containing the required time data: t_r, t_g and t_v, where the times are delimited with a semi-column. If multiple cells are filled in the same row, the process can be completed with either of the resources, as explained in Section 2. For manual work, the times were usually the same, or very similar, for every possible resource, because the resource location is not very time relevant. The letter X in a cell means that it is impossible for the process to be completed on this resource. Since Job 1 in Figure 15 only needs 18 processes to finish, processes 19 and 20 get a value 0 for every machine. The company used LTTS for allowances` determination, and used a fixed value for all resources. Figures 16 and 17 display the actual tool window where all the parameters can be set, and the obtained solution is visualized [33].

To calculate the search space, we used the formula (12), so the upper chromosome part can be evaluated as:

$$\frac{(p\ x\ n)!}{(p!)^n} = \frac{(20\ x\ 7)!}{(20!)^7} \cong 2{,}67\ x 10^{112}$$

and the lower part to (without infeasible solutions):

$$\prod_{i=1}^{n} \prod_{k=1}^{p} \cong\ 7{,}54\ x\ 10^{30}$$

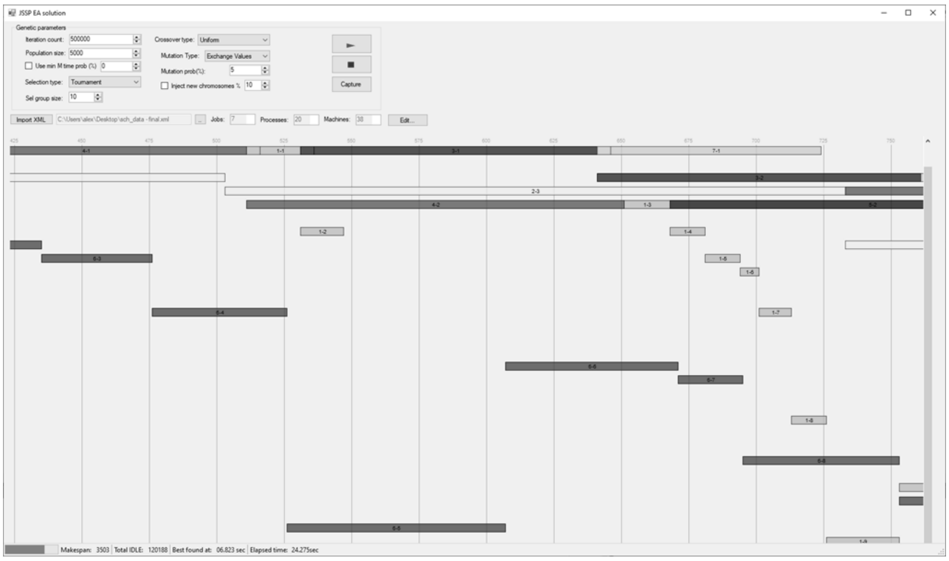
meaning a search space size without any infeasible solutions is roughly around $2{,}01\ x\ 10^{143}$ different chromosomes, or around $1{,}01\ x\ 10^{253}$ including the infeasible solutions.

The settings we used for GA were chosen experimentally in order to find the optimal solution as fast as possible:

Table 6. Used genetic parameters and their settings

Iteration count	500,000
Population size	5,000
Crossover type	Uniform (pc = 0.5)
Selection type	Tournament
Tournament size	10
Mutation type	Exchange values
Mutation probability	5%

The algorithm has been run 20 times in sequence using the parameters from Table 6. 95% (19/20) of the time, the algorithm was able to find the optimal makespan (3,503 minutes) in around 30-35 seconds on the I7-7800X 6 core computer (12 logical processors) with 16GB of RAM.



JSSP EA solution

Genetic parameters

Iteration count: 500000

Population size: 5000

□ Use min M time prob (%) 0

Selection type: Tournament

Sel group size: 10

Crossover type: Uniform

Mutation Type: Exchange Values

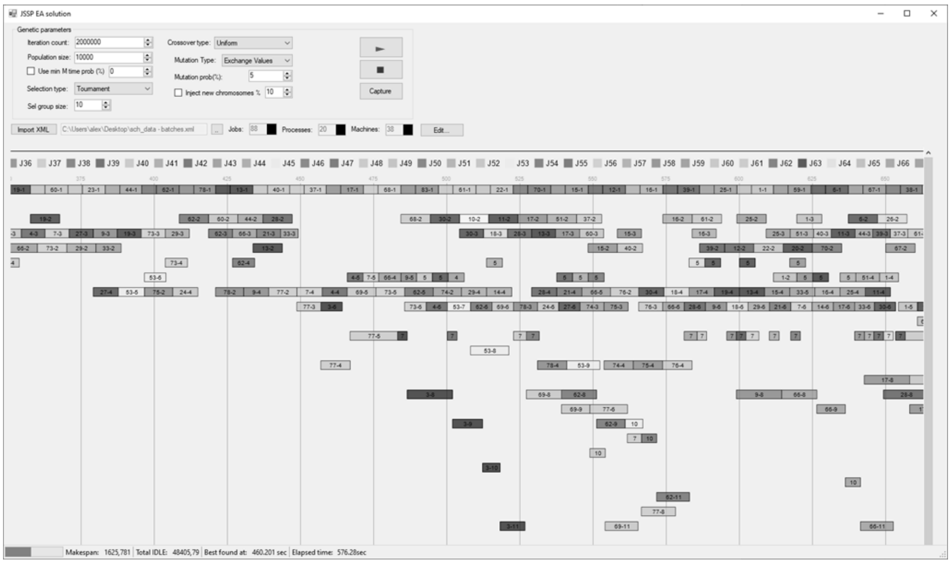
Mutation prob(%): 5

□ Inject new chromosomes % 10

Import XML C:\Users\alex\Desktop\sch_data - final.xml Jobs: 7 Processes: 20 Machines: 38

Makespan: 3503 | Total IDLE: 120188 | Best found at: 06.823 sec | Elapsed time: 24.275sec

Fig. 16. App window example without lot streaming



JSSP EA solution

Genetic parameters

Iteration count: 2000000

Population size: 10000

□ Use min M time prob (%) 0

Selection type: Tournament

Sel group size: 10

Crossover type: Uniform

Mutation Type: Exchange Values

Mutation prob(%): 5

□ Inject new chromosomes % 10

Import XML C:\Users\alex\Desktop\sch_data - batches.xml Jobs: 88 Processes: 20 Machines: 38

Makespan: 1625,781 | Total IDLE: 48405,79 | Best found at: 460.201 sec | Elapsed time: 576.28sec

Capture
Edit...
J36 J37 J38 J39 J40 J41 J42 J43 J44 J45 J46 J47 J48 J49 J50 J51 J52 J53 J54 J55 J56 J57 J58 J59 J60 J61 J62 J63 J64 J65 J66
375
400
425
450
475
500

Fig. 17. App window example using lot streaming

Table 7 shows the data gathered across the 20 runs. Figure 18 shows a 3d histogram displaying the discrete IDLE time occurrence per machine.

Table 7. Results of running the algorithm 20 times

	Best found at [s]	**Elapsed time [s]**	**Total IDLE time [min]**	**Makespan [min]**
Run 1	7,367	30,707	120369	3503
Run 2	4,532	30,979	120384	3503
Run 3	6,311	30,694	120379	3503
Run 4	6,91	31,228	120305	3503
Run 5	4,425	30,874	120305	3503
Run 6	25,999	30,84	120379	3506
Run 7	5,987	35,819	120196	3503
Run 8	6,469	35,394	120308	3503
Run 9	7,946	35,564	120389	3503
Run 10	8,427	35,429	120379	3503
Run 11	9,1	34,821	120359	3503
Run 12	7,852	34,854	120374	3503
Run 13	4,72	34,581	120318	3503
Run 14	10,76	34,057	120342	3503
Run 15	8,272	34,643	120364	3503
Run 16	5,286	34,831	120285	3503
Run 17	10,051	34,49	120152	3503
Run 18	7,093	34,797	120374	3503
Run 19	4,017	34,763	120374	3503
Run 20	8,819	35,921	120384	3503

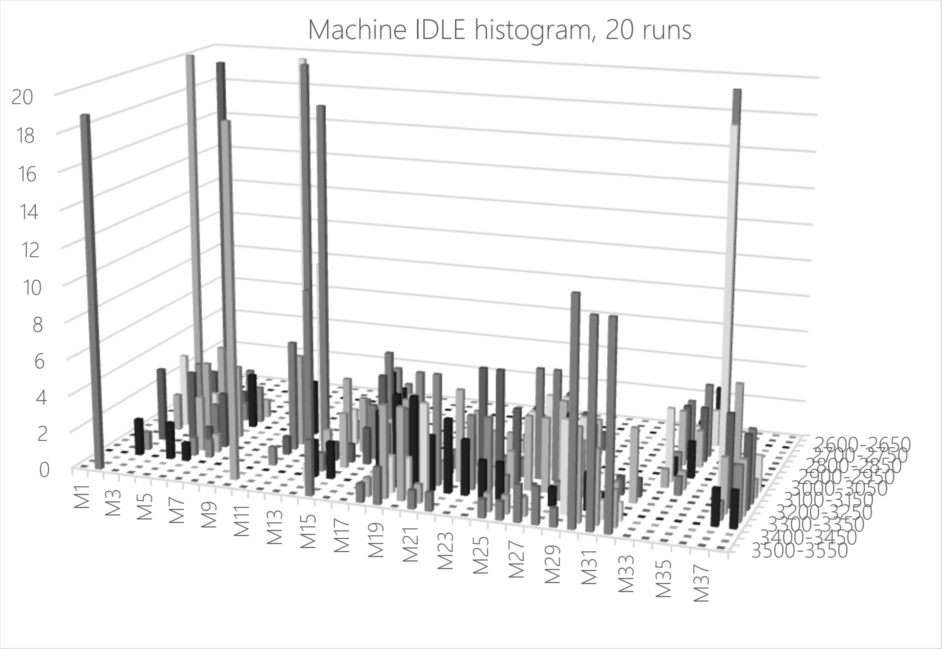


Fig. 18. IDLE time histogram per resource

The IDLE interval for all resources lay between [2583,3586] so the histogram IDLE intervals are defined between [2600,3550], with a step of 50. The graph shows that the machine IDLE times` occurrences are mainly concentrated in the upper bound of the IDLE interval, closer to the makespan (3000 < IDLE < makespan), except for maybe M33 and M34, meaning all the other resources are badly utilized. Ideally, the IDLE time occurrence count should be as close to 0 as possible.

Because the shoe company production was lot-organized, they split the entire job (order) quantity into smaller lots. The lot size was standardized and consisted of 10 pairs. This kind of approach is called lot streaming [4]. By splitting each job into sub-jobs of 10, the EA job parameter stretches from 7 to 88, remarkably increasing the search space (formula 12):

$$\frac{(p\ x\ n)!}{(p!)^n} = \frac{(20\ x\ 88)!}{(20!)^{88}} \cong 6{,}12\ x 10^{4949}$$

$$\prod_{i=1}^{n} \prod_{k=1}^{p} count(possible\ machines)_{(ik)} \cong\ 3{,}47\ x\ 10^{357}$$

meaning we have around $2{,}12\ x\ 10^{5307}$ different chromosomes without infeasible solutions, or $2{,}32\ x\ 10^{6711}$ including all the infeasible solutions. Because of the increase the algorithm wasn't able to find the optimal solution with the EA parameters described in the use case. Even by doubling the initial population and quadrupling the iteration count, the algorithm success rate dropped significantly. The best-found makespan was 1,504, reducing the result found with the non-lot approach by 57%.

Table 8. Comparison of IDLE time/utilization between lot streaming and conventional scheduling

	Utilization		**IDLE time**	
	8 jobs	**88 jobs**	**8 jobs**	**88 jobs**
M1	20,75%	75,04%	79,25%	24,96%
M2	0,00%	0,00%	100,00%	100,00%
M3	10,76%	26,86%	89,24%	73,14%
M4	18,01%	41,94%	81,99%	58,06%
M5	12,15%	26,78%	87,85%	73,22%
M6	9,89%	22,72%	90,11%	77,28%
M7	10,78%	25,40%	89,23%	74,60%
M8	22,36%	51,78%	77,64%	48,22%
M9	18,84%	43,87%	81,16%	56,13%
M10	0,20%	0,47%	99,80%	99,53%
M11	10,31%	21,74%	89,70%	78,26%
M12	10,90%	25,92%	89,10%	74,08%
M13	9,11%	21,20%	90,90%	78,80%
M14	11,37%	27,39%	88,63%	72,61%
M15	9,17%	21,71%	90,84%	78,29%
M16	12,18%	29,73%	87,82%	70,27%
M17	10,45%	22,98%	89,55%	77,02%
M18	5,40%	8,28%	94,60%	91,72%
M19	3,61%	7,72%	96,39%	92,28%
M20	3,49%	8,79%	96,51%	91,21%
M21	4,11%	12,08%	95,90%	87,92%
M22	7,71%	20,39%	92,29%	79,61%
M23	7,00%	17,25%	93,00%	82,75%
M24	8,60%	17,88%	91,40%	82,12%
M25	7,99%	15,16%	92,01%	84,84%
M26	9,73%	24,28%	90,27%	75,72%
M27	8,48%	18,13%	91,52%	81,87%
M28	9,61%	21,09%	90,40%	78,91%
M29	4,65%	12,56%	95,35%	87,44%
M30	0,61%	4,63%	99,40%	95,37%
M31	1,15%	8,20%	98,86%	91,80%
M32	1,94%	4,34%	98,06%	95,66%
M33	17,31%	48,08%	82,69%	51,92%
M34	19,03%	36,54%	80,98%	63,46%
M35	19,01%	44,27%	80,99%	55,73%
M36	12,25%	28,52%	87,76%	71,48%
M37	8,26%	23,84%	91,74%	76,16%
M38	7,75%	22,76%	92,26%	77,24%
AVG	**9,60%**	**23,43%**	**90,40%**	**76,57%**

Table 8 displays the utilization data comparison between using the lot streaming (best found) and not using the lot streaming (conventional scheduling). The average machine utilization rate rose from 9,60% to 23,43%.

We must emphasize the importance of setup times t_r when using lot streaming. They play a significant role when searching for the optimal schedule. If they occur a maximum of p times per job j in conventional planning, the number of occurrences multiplies by $j\ /\ lot\ size$ when using lot streaming. This can cause the utilization rate to rise, but can also cause the makespan to increase.

Since the production was using lot streaming, we must compare the computed makespan of 1504 minutes with real-world data. The 88 carts were finished in around 3,5 shifts, resulting in 1680 minutes, meaning an efficiency increase of around 10,5% could be achieved by scheduling alone, confirming that using JSSP solutions in real-work can prove beneficial. Currently, daily scheduling is still left to the product engineers. Because of the size and complexity of the scheduling problem, no traditional tool can be used, so they launch the products daily simply by experience alone.

5. Conclusion

Implementing an EA solution for a specific Job Shop Scheduling Problem and combining it with Industrial Engineering knowledge proved to deliver good results. Many issues found in other types of JSSP were addressed, like shorter stochastic events and setup times, for example. Using the typical time components found in manufacturing brought the user domain and the science domain closer together. However, any schedule provided by a schedule optimization system like the one proposed can only be used as a guideline. The tool introduced in the article can be used at the end of the day for next-day scheduling by product engineers to search for the best option on how and when to launch the planned products but still must be extended by experience parameters that aren't included in the algorithm like specific worker skills or machine experience, for instance. Any longer stochastic event that can occur (and is not covered by allowances) would render the provided schedule unusable. The user can still rerun the optimization with the current situation parameters, but that could prove time-consuming. The proposed schedule could require rerouting the resources in a completely different way. Another drawback of classical Industrial Engineering is that all the time components necessary still demand a lot of work from a time analyst before all the technological times are determined. On the plus side of Industrial Engineering, we can find many manufacturing companies and even ERP systems using the time definitions mentioned. Another plus side is that any company target times should be as accurate as possible, otherwise causing inaccurate planning and cost calculations, providing an excellent fundament for JSSP. Future work should extend the solution with the proposed IoT architecture to measure and classify production times in real-time, and deliver the required data for the JSSP solution that would be run dynamically automatically.

References

1. Johnson, S. M: Optimal Two and Three-Stage Production Schedules with Setup Times Included. Naval Research Logistic Quarterly, Vol. 1, No. 1. (1954)

2. Davis, L.: Job Shop Scheduling with Genetic Algorithms., Proc. of 1st Int. Conf. on Genetic Algorithms, Lawrence Erlbaus Associates, p. 136-140. (1985)
3. Werner, F.: Genetic algorithms for shop scheduling problems: a survey. Preprint Series. 11. 1-66. (2011)
4. Yigit, T., Birogul, S., Elmas, C.: Lot streaming based job-shop scheduling problem using hybrid genetic algorithm. Scientific research and essays. 6. 2873-2887. 10.5897/SRE10.152. (2011)
5. Bierwirth C.: A generalized permutation approach to job shop scheduling with genetic algorithms. Operations-Research-Spektrum, June 1995, Volume 17, Issue 2-3, pp 87-92. (1995)
6. Omar, M., Baharum, A., Hasan, A. Y.: A job-shop scheduling problem (JSSP) using genetic algorithm (GA). Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications Universiti Sains Malaysia, Pennang, June 13-15. (2006)
7. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. IEEE Congress on Evolutionary Computation, 1769-1776. (2005)
8. REFA Verband für Arbeitsgestaltung: REFA Methodenlehre der Betriebsorganisation: Datenermittlung. München: Carl Hanser Verlag. (1997)
9. Abdolrazzagh-Nezhad, M., Abdullah, S.: Job Shop Scheduling: Classification, Constraints and Objective Functions. World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:11, No:4, p. 423-428. (2017)
10. Seifermann S., Böllhoff J., Metternich J., Bellaghnach A.: Evaluation of Work Measurement Concepts for a Cellular Manufacturing Reference Line to enable Low Cost Automation for Lean Machining, Procedia CIRP 17 p. 588 – 593. (2014)
11. Maynard H, Stegemerten G, Schwab J.: Methods-Time Measurement. New York: McGraw-Hill. (1948)
12. Bierwirth C., Mattfeld D.C., Kopfer H.: On permutation representations for scheduling problems. In: Voigt HM., Ebeling W., Rechenberg I., Schwefel HP. (eds) Parallel Problem Solving from Nature — PPSN IV. PPSN 1996. Lecture Notes in Computer Science, vol 1141. Springer, Berlin, Heidelberg. (1996)
13. Bokranz R, Landau K. Produktivitätsmanagement von Arbeitssystemen. MTM-Handbuch (Productivity Management of Work Systems. MTMHandbook). Stuttgart: Schäffer-Poeschel. (2006)
14. Bundesministerium des Innern/Bundesverwaltungsamt (Hrsg.): Handbuch für Organisationsuntersuchungen und Personalbedarfsermittlung. (2018)
15. Poeschel, F.: Verteilzeit. In: Landau, Kurt (Hrsg.): Lexikon Arbeitsgestaltung : Best Practise im Arbeitsprozess. Stuttgart: Genter. (2007)
16. Sotskov, Y.N.; Matsveichuk, N.M.; Hatsura,V.D.: Schedule Execution for Two-Machine Job-Shop to Minimize Makespan with Uncertain Processing Times. Mathematics 2020, 8, 1314. (2020)
17. Sauvey, C.; Trabelsi, W.; Sauer, N. Mathematical Model and Evaluation Function for Conflict-Free Warranted Makespan Minimization of Mixed Blocking Constraint Job-Shop Problems. Mathematics 2020, 8, 121. (2020)
18. Luan, F.; Cai, Z.; Wu, S.; Liu, S.Q.; He, Y. Optimizing the Low-Carbon Flexible Job Shop Scheduling Problem with Discrete Whale Optimization Algorithm. Mathematics 2019, 7, 688. (2019)
19. Bak, N., Chang, B-M., N., Choi, K.: Smart Block: A visual block language and its programming environment for IoT. Journal of Computer Languages 60, 100999. (2020)
20. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation 10(6), 646-657. (2006)
21. Črepinšek, M., Mernik, M., Žumer, V.: Extracting grammar from programs: brute force approach. ACM SIGPLAN Notices 40(4), 29-38. (2005)

22. Črepinšek, M., Liu, S.-H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: A survey. ACM Computing Surveys 45(3): 35:1-35:33. (2013)
23. Črepinšek, M., Liu, S.-H., Mernik, L., Mernik, M.: Is a comparison of results meaningful from the inexact replications of computational experiments? Soft Computing 20, 223-235. (2016)
24. Črepinšek, M., Liu, S.-H., Mernik, M., Ravber, M.: Long term memory assistance for evolutionary algorithms, Mathematics 7 (11). (2019)
25. Du, Z., Chen, K.: Enhanced Artificial Bee Colony with Novel Search Strategy and Dynamic Parameter. Computer Science and Information Systems 16(3), 939–957. (2019)
26. Eiben, A,.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg. (2003)
28. Hrnčič, D., Mernik, M., Bryant, B. R., Javed, F.: A memetic grammar inference algorithm for language learning. Applied Soft Computing 12 (3), 1006-1020. (2012)
29. Javed, F., Bryant, B. R., Črepinšek, M., Mernik, M., Sprague, A.: Context-free grammar induction using genetic programming, in: Proceedings of the 42nd Annual Southeast Regional Conference, ACM-SE 42, 404-405. (2004)
30. Jesenik, M., Mernik, M., Trlep, M.: Determination of a hysteresis model parameters with the use of different evolutionary methods for an innovative hysteresis model. Mathematics 8 (2). (2020)
31. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8(1), 687-697. (2008)
32. Kovačević, Ž., Mernik, M., Ravber, M., Črepinšek, M.: From grammar inference to semantic inference-an evolutionary approach. Mathematics 8 (5). (2020)
33. Mei, H., Ma, Y., Wei, Y., Chen, W.: The design space of construction tools for information visualization: A survey. Journal of Visual Languages & Computing 44, 120 – 132. (2018)
34. Mernik, M., Liu, S.-H., Karaboga, D., Črepinšek, M.: On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation. Information Sciences 291, 115-127. (2015)
35. Grammatikis, P.I.R, Sarigiannidis, P.G., Moscholios, I.D.: Securing the Internet of Things: Challenges, threats and solutions. Internet of Things 5, 41-70. (2019)
36. Rao, V. R., Savsani, V., Vakharia, P. D.: Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems. Information Sciences 183, 1-15. (2012)
37. Rathee, A., Chhabra, J. K.: A multi-objective search based approach to identify reusable software components. Journal of Computer Languages 52, 26-43. (2019)
38. Russell, S., Norvig, P.: Artificial intelligence: a modern approach. Prentice Hall. (2002)
39. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: IEEE Congress on Evolutionary Computation, pp. 71–78. (2013)
40. Qiu, X., & Lau, H. Y.. An AIS-based hybrid algorithm for static job shop scheduling problem. Journal of Intelligent Manufacturing, 25(3), 489-503. (2014)
41. Zhang, G., Gao, L., & Shi, Y.: An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications, 38(4), 3563-3573. (2011)
42. Pinedo, M. L. (Ed.).: Scheduling, Theory, Algorithm and Systems. New York: Springer. (2012)
43. Kuroda, M., & Wang, Z.: Fuzzy job shop scheduling. International Journal of Production Economics, 44(1), 45-51. (1996)
44. Ahmad, F., & Khan, S. A.:Module-based architecture for a periodic job-shop scheduling problem. Computers & Mathematics with Applications, 64(1), 1-10. (2012)
45. Brucker, P., Burke, E. K., & Groenemeyer, S.:A mixed integer programming model for the cyclic job-shop problem with transportation. Discrete applied mathematics, 160(13-14), 1924-1935. (2012)
46. Ebadi, A., & Moslehi, G.: Mathematical models for preemptive shop scheduling problems. Computers & Operations Research, 39(7), 1605-1614. (2012)

47. Schuster, C. J., & Framinan, J. M.: Approximative procedures for no-wait job shop scheduling. Operations Research Letters, 31(4), 308-318. (2003)
48. Baptiste, P., Flamini, M., & Sourd, F.: Lagrangian bounds for just-in-time job-shop scheduling. Computers & Operations Research, 35(3), 906-915. (2008)
49. Zhang, R., & Wu, C.: A hybrid approach to large-scale job shop scheduling. Applied intelligence, 32(1), 47-59. (2010)
50. Topaloglu, S., & Kilincli, G.: A modified shifting bottleneck heuristic for the reentrant job shop scheduling problem with makespan minimization. The International Journal of Advanced Manufacturing Technology, 44(7), 781-794. (2009)
51. Jerebic, J., Mernik, M., Liu, S-H., Ravber M., Baketarić, M., Mernik, L., Črepinšek, M.: A novel direct measure of exploration and exploitation based on attraction basins. Expert Systems with Applications, Volume 167, 114353. (2021)

Sašo Sršen has graduated from the Faculty of Electrical Engineering and Computer Science, University of Maribor, in 2014. Currently, he is a Ph.D. student and co-owner of the company PISK, a licensed REFA/MTM company. He is working as a consultant/teacher for production companies. His research interests include domain-specific languages, evolutionary algorithms, Industry 4.0, predictive analytics, IOT, and simulation.

Marjan Mernik received the M.Sc. and Ph.D. degrees in computer science from the University of Maribor in 1994 and 1998, respectively. He is currently a professor at the University of Maribor, Faculty of Electrical Engineering and Computer Science. He was a visiting professor at the University of Alabama at Birmingham, Department of Computer and Information Sciences. His research interests include programming languages, compilers, domain-specific (modeling) languages, grammar-based systems, grammatical inference, and evolutionary computations. He is a member of the IEEE, ACM, and EAPLS. Dr. Mernik is the Editor-In-Chief of the Journal of Computer Languages, as well as Associate Editors of the Applied Soft Computing journal, Information Sciences journal, and Swarm and Evolutionary Computation journal. He is being named a Highly Cited Researcher for years 2017 and 2018. More information about his work is available at https://lpm.feri.um.si/en/members/mernik/

Received: October 09, 2020; Accepted: December 30, 2020

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

004

COMPUTER Science and Information
Systems : the International journal /
Editor-in-Chief Mirjana Ivanović. – Vol. 18,
No 1 (2021) - . – Novi Sad (Trg D. Obradovića 3):
ComSIS Consortium, 2021 - (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