

# Advances in Architectures, Big Data, and Machine Learning Techniques for Complex Internet of Things Systems

Lead Guest Editor: David Gil

Guest Editors: Magnus Johnsson, Higinio Mora, and Julian Szymanski





---

# **Advances in Architectures, Big Data, and Machine Learning Techniques for Complex Internet of Things Systems**

Complexity

---

# **Advances in Architectures, Big Data, and Machine Learning Techniques for Complex Internet of Things Systems**

Lead Guest Editor: David Gil

Guest Editors: Magnus Johnsson, Higinio Mora,  
and Julian Szymanski



Copyright © 2019 Hindawi. All rights reserved.

This is a special issue published in “Complexity.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



## Editorial Board

José A. Acosta, Spain  
Carlos F. Aguilar-Ibáñez, Mexico  
Mojtaba Ahmadi Khanezar, UK  
Tarek Ahmed-Ali, France  
Alex Alexandridis, Greece  
Basil M. Al-Hadithi, Spain  
Juan A. Almendral, Spain  
Diego R. Amancio, Brazil  
David Arroyo, Spain  
Mohamed Boutayeb, France  
Átila Bueno, Brazil  
Arturo Buscarino, Italy  
Guido Caldarelli, Italy  
Eric Campos-Canton, Mexico  
Mohammed Chadli, France  
Émile J. L. Chappin, Netherlands  
Diyi Chen, China  
Yu-Wang Chen, UK  
Giulio Cimini, Italy  
Danilo Comminiello, Italy  
Sara Dadras, USA  
Sergey Dashkovskiy, Germany  
Manlio De Domenico, Italy  
Pietro De Lellis, Italy  
Albert Diaz-Guilera, Spain  
Thach Ngoc Dinh, France  
Jordi Duch, Spain  
Marcio Eisencraft, Brazil  
Joshua Epstein, USA  
Mondher Farza, France  
Thierry Floquet, France  
Mattia Frasca, Italy  
José Manuel Galán, Spain  
Lucia Valentina Gambuzza, Italy  
Bernhard C. Geiger, Austria

Carlos Gershenson, Mexico  
Peter Giesl, UK  
Sergio Gómez, Spain  
Lingzhong Guo, UK  
Xianggui Guo, China  
Sigurdur F. Hafstein, Iceland  
Chittaranjan Hens, India  
Giacomo Innocenti, Italy  
Sarangapani Jagannathan, USA  
Mahdi Jalili, Australia  
Jeffrey H. Johnson, UK  
M. Hassan Khooban, Denmark  
Abbas Khosravi, Australia  
Toshikazu Kuniya, Japan  
Vincent Labatut, France  
Lucas Lacasa, UK  
Guang Li, UK  
Qingdu Li, China  
Chongyang Liu, China  
Xiaoping Liu, Canada  
Xinzhi Liu, Canada  
Rosa M. Lopez Gutierrez, Mexico  
Vittorio Loreto, Italy  
Noureddine Manamanni, France  
Didier Maquin, France  
Eulalia Martínez, Spain  
Marcelo Messias, Brazil  
Ana Meštrović, Croatia  
Ludovico Minati, Japan  
Ch. P. Monterola, Philippines  
Marcin Mrugalski, Poland  
Roberto Natella, Italy  
Sing Kiong Ngung, New Zealand  
Nam-Phong Nguyen, USA  
B. M. Ombuki-Berman, Canada

Irene Otero-Muras, Spain  
Yongping Pan, Singapore  
Daniela Paolotti, Italy  
Cornelio Posadas-Castillo, Mexico  
Mahardhika Pratama, Singapore  
Luis M. Rocha, USA  
Miguel Romance, Spain  
Avimanyu Sahoo, USA  
Matilde Santos, Spain  
Josep Sardanyés Cayuela, Spain  
Ramaswamy Savitha, Singapore  
Hiroki Sayama, USA  
Michele Scarpiniti, Italy  
Enzo Pasquale Scilingo, Italy  
Dan Selişteanu, Romania  
Dehua Shen, China  
Dimitrios Stamovlasis, Greece  
Samuel Stanton, USA  
Roberto Tonelli, Italy  
Shahadat Uddin, Australia  
Gaetano Valenza, Italy  
Dimitri Volchenkov, USA  
Christos Volos, Greece  
Zidong Wang, UK  
Yan-Ling Wei, Singapore  
Honglei Xu, Australia  
Yong Xu, China  
Xinggang Yan, UK  
Baris Yuce, UK  
Massimiliano Zanin, Spain  
Hassan Zargarzadeh, USA  
Rongqing Zhang, USA  
Xianming Zhang, Australia  
Xiaopeng Zhao, USA  
Quanmin Zhu, UK

# Contents

## **Advances in Architectures, Big Data, and Machine Learning Techniques for Complex Internet of Things Systems**

David Gil , Magnus Johnsson , Higinio Mora , and Julian Szymanski 

Editorial (3 pages), Article ID 4184708, Volume 2019 (2019)

## **Hybrid Genetic Grey Wolf Algorithm for Large-Scale Global Optimization**

Qinghua Gu , Xuexian Li , and Song Jiang 

Research Article (18 pages), Article ID 2653512, Volume 2019 (2019)

## **Review of the Complexity of Managing Big Data of the Internet of Things**

David Gil , Magnus Johnsson , Higinio Mora , and Julian Szymański 





Review Article (12 pages), Article ID 4592902, Volume 2019 (2019)

## **Recent Progress of Anomaly Detection**

Xiaodan Xu, Huawen Liu, and Minghai Yao 

Review Article (11 pages), Article ID 2686378, Volume 2019 (2019)

## **Secure UAV-Based System to Detect Small Boats Using Neural Networks**

Moisés Lodeiro-Santiago , Pino Caballero-Gil , Ricardo Aguasca-Colomo ,  
and Cándido Caballero-Gil 


Research Article (11 pages), Article ID 7206096, Volume 2019 (2019)

## **Fuzzy Linguistic Protoforms to Summarize Heart Rate Streams of Patients with Ischemic Heart Disease**

María Dolores Peláez-Aguilera, Macarena Espinilla , María Rosa Fernández Olmo, and Javier Medina



Research Article (11 pages), Article ID 2694126, Volume 2019 (2019)

## **Fully Flexible Parallel Merge Sort for Multicore Architectures**

Zbigniew Marszałek, Marcin Woźniak , and Dawid Połap


Research Article (19 pages), Article ID 8679579, Volume 2018 (2019)

## **Application of the Variable Precision Rough Sets Model to Estimate the Outlier Probability of Each Element**

Francisco Maciá Pérez , Jose Vicente Berna Martienz , Alberto Fernández Oliva,  
and Miguel Abreu Ortega

Research Article (14 pages), Article ID 4867607, Volume 2018 (2019)

## **Case-Based Reasoning: The Search for Similar Solutions and Identification of Outliers**

P. S. Szczepaniak and A. Duraj 

Research Article (12 pages), Article ID 9280787, Volume 2018 (2019)

## Editorial

# Advances in Architectures, Big Data, and Machine Learning Techniques for Complex Internet of Things Systems

David Gil <sup>1</sup>, Magnus Johnsson <sup>2,3,4</sup>, Higinio Mora <sup>1</sup>, and Julian Szymanski <sup>5</sup>

<sup>1</sup>University of Alicante, Alicante, Spain

<sup>2</sup>Malmö University, Malmö, Sweden

<sup>3</sup>Department of Intelligent Cybernetic Systems, NRNU MEPhI, Moscow, Russia

<sup>4</sup>AI Research AB, Höör, Sweden

<sup>5</sup>Gdansk University of Technology, Gdansk, Poland

Correspondence should be addressed to David Gil; david.gil@ua.es

Received 27 February 2019; Accepted 27 February 2019; Published 24 March 2019

Copyright © 2019 David Gil et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The field of Big Data is rapidly developing with a lot of ongoing research, which will likely continue to expand in the future. A crucial part of this is Knowledge Discovery from Data (KDD), also known as the Knowledge Discovery Process (KDP). This process is a very complex procedure, and for that reason it is essential to divide it into several steps (Figure 1). Some authors use five steps to describe this procedure, whereas others use only four. We use the following four-step description:

- (1) Generation of Data: Data is generated from a multitude of various data sources, such as sensors, social media, the web, a multitude of devices, software applications, people, and various kinds of sensors [1].
- (2) Collection of Data: This step involves the storage of data into various types of databases, such as MongoDB, elastic, InfluxDB, MySQL, and NoSQL suitable for Big Data technologies [2–7]. Often the raw sensor data are collected, but it is common to also link them to contextual information [8–10]. Other tasks such as cleaning, integration, and transformation of data are essential for the optimal storage in databases [11–13]. Several tools and technologies suitable to semi-automatize tasks such as Extract, Transform, and Load (ETL) [14] exist, e.g., Apache NIFI [15, 16] and Pentaho [17, 18]. These are very useful since there are many tasks involved in this step of the KDD procedure.
- (3) Machine Learning and Data Mining: Diverse machine learning and data mining methods are applied

and benchmarked, and the results are compared. It should be kept in mind that though there are many machine learning methods, not all of them are suitable to use with Big Data [19, 20].

- (4) Classification, Prediction, and Visualization: This step focuses on, in particular, the obtaining of visualizations that present all the classification and prediction results in a useful way. Tools such as Grafana [21] could help to interpret the data visually, but also to simplify the identification of Key Performance Indicators (KPI) [22].

This special issue received in total 19 submitted papers, and after a meticulous reviewing process the editors decided to accept eight of these for publication, which implies an acceptance rate of about 42%.

Anomaly analysis is a crucial issue since it is a significant part of many areas, such as medical health, credit card fraud, and intrusion detection (X. Xu et al.). The authors of this paper provide a complete state-of-the-art presentation of anomaly detection. High dimensionalities and mixed types of data are the focus of this study as the identification of anomalous patterns is far from trivial. The authors introduce the reader to current advances on anomaly detection, while debating the pros and cons of various detection methods.

There are areas that are well-known, though barely referenced in the literature. For example, the one presented by M. Lodeiro-Santiago et al., where the goal is to detect small boats (pateras) to help address the problem of dangerous immigration. In this paper, the authors use deep



FIGURE 1: Procedure for the KDD.

convolutional neural networks to improve detection methods based on image processing through the application of filters. Their novel approach is able to recognise the boats through patterns regardless of where they are located. The proposed approach, which works in real-time, allows the detection of boats and people for search and rescue teams in order to plan for rescue operations before an emergency happens. The proposed method includes the use of essential cryptographic protocols for the protection of the highly sensitive information managed.

A method for the evaluation of heart rate streams in patients with ischemic heart disease is presented by M. D. Peláez-Aguilera et al. The authors present an innovative linguistic approach to manage relevant linguistic descriptions (protoforms). This provides a foundation for the cardiac rehabilitation team to identify sessions with significance indicators through linguistic summaries. As it is faced in the manuscript, cardiac rehabilitation programs are crucial to significantly decrease mortality rates in high-risk patients with ischemic heart disease.

In the work presented by Z. Marszałek et al., a fully flexible sorting method designed for parallel processing is presented. The authors describe a method based on modified merge sort designed for multicore architectures. The flexibility of the method, which is implemented for a number of processors, increases the efficiency of sorting by distributing the tasks between logical cores in a flexible way. Since powerful computer resources are often not very well exploited, their main goal is to use efficient algorithms to support the proficient use of all available resources.

F. M. Pérez et al. present a theoretical framework based on a generalisation of rough sets theory. This allows the establishment of a stochastic approach to solving the problem of outliers within a specific universe of data. An algorithm based on this theoretical framework is developed to make it suitable for large data volume applications. The experiments carried out validate the proposed algorithm in comparison to various algorithms analysed in the literature.

The work proposed by P. S. Szczepaniak and A. Duraj concerns the problem of outlier detection through the application of case-based reasoning. The authors argue that while this method has been successfully applied in an extensive variety of other domains, it has never been used for outlier detection.

In the manuscript presented by Q. Gu et al., the authors propose a Hybrid Genetic Grey Wolf Algorithm in order to improve the disadvantage of Grey Wolf Optimizer when solving Large-Scale Global Optimization problems.

Finally, D. Gil et al. provide a review highlighting the fact that the complexity of managing Big Data is one of the

main challenges in the developing field of the Internet of Things (IoT). The review divides the discovery of knowledge into the four general steps sketched above and evaluates the most novel technologies involved. These include IoT data gathering, data cleaning and integration, data mining and machine learning, and classification, prediction, and visualization.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors acknowledge the support of the Internet of Things and People (IOTAP) Research Center at Malmö University in Sweden. This work was also supported by the Spanish Research Agency (AEI) and the European Regional Development Fund (ERDF) under the project CloudDriver4Industry TIN2017-89266-R. This work has also been funded by the Spanish Ministry of Economy and Competitiveness (MINECO/FEDER) under the granted Project SEQUOIA-UA (management requirements and methodology for Big Data analytics) TIN2015-63502-C3-3-R.

David Gil  
Magnus Johnsson  
Higinio Mora  
Julian Szymanski

## References

- [1] C. A. Zaslavsky and D. G. Perera, "Sensing as a service and big data," <https://arxiv.org/abs/1301.0159>, 2013.
- [2] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database," in *Proceedings of the 6th International Conference on Pervasive Computing and Applications (ICPCA '11)*, pp. 363–366, Port Elizabeth, South Africa, October 2011.
- [3] R. Cattell, "Scalable SQL and NoSQL data stores," *ACM SIGMOD Record*, vol. 39, no. 4, pp. 12–27, 2010.
- [4] Ishwarappa and J. Anuradha, "A brief introduction on big data 5Vs characteristics and hadoop technology," *Procedia Computer Science*, vol. 48, no. C, pp. 319–324, 2015.
- [5] S. Patni, *Pro RESTful APIs: Design, Build and Integrate with REST, JSON, XML and JAX-RS*, Apress, Berkeley, CA, USA, 2017.
- [6] M. Chen, S. Mao, and Y. Liu, "Big data: a survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [7] C. L. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: a survey on big data," *Information Sciences*, vol. 275, pp. 314–347, 2014.

- [8] S. Satpathy, B. Sahoo, and A. K. Turuk, "Sensing and actuation as a service delivery model in cloud edge centric internet of things," *Future Generation Computer Systems*, vol. 86, pp. 281–296, 2018.
- [9] J.-P. Calbimonte, H. Jeung, O. Corcho, and K. Aberer, "Semantic sensor data search in a large scale federated sensor network," in *Proceedings of the 4th International Workshop on Semantic Sensor Networks 2011, SSN 2011 - A 10th International Semantic Web Conference, ISWC 2011*, pp. 23–38, Germany, October 2011.
- [10] S. Li, L. D. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, 2013.
- [11] Z. Doan, A. Halevy, and A. Ives, *Principles of Data Integration*, Elsevier, 2012.
- [12] H. Gonzalez, A. Halevy, C. S. Jensen et al., "Google fusion tables: web-centered data management and collaboration," in *Proceedings of the 1st ACM symposium*, p. 175, June 2010.
- [13] A. Y. Halevy, "Answering queries using views: a survey," *The VLDB Journal*, vol. 10, no. 4, pp. 270–294, 2001.
- [14] P. Vassiliadis, "A survey of extract-transform-load technology," *International Journal of Data Warehousing and Mining*, vol. 5, no. 3, pp. 1–27, 2009.
- [15] Apache NiFi.
- [16] J. N. Hughes, M. D. Zimmerman, C. N. Eichelberger, and A. D. Fox, "A survey of techniques and open-source tools for processing streams of spatio-temporal events," in *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS 2016*, USA.
- [17] R. Bouman and J. Van Dongen, *Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL*, 2009.
- [18] M. Casters, R. Bouman, and J. Van Dongen, *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration*, John Wiley & Sons, 2010.
- [19] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.
- [20] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the internet of things: literature review and challenges," *International Journal of Distributed Sensor Networks*, vol. 2015, no. i, 2015.
- [21] Grafana, "The open platform for analytics and monitoring".
- [22] E. Betke and J. Kunkel, "Real-time I/O-monitoring of HPC applications with SIOX, elasticsearch, Grafana and FUSE," in *Proceedings of the ISC High Performance 2017: High Performance Computing*, pp. 174–186, Springer, Cham, Switzerland, 2017.

## Research Article

# Hybrid Genetic Grey Wolf Algorithm for Large-Scale Global Optimization

Qinghua Gu , Xuexian Li , and Song Jiang 

*School of Management, Xi'an University of Architecture and Technology, Shaanxi 710055, China*

Correspondence should be addressed to Song Jiang; [jiangsong925@163.com](mailto:jiangsong925@163.com)

Received 18 October 2018; Revised 26 December 2018; Accepted 22 January 2019; Published 12 February 2019

Guest Editor: Higinio Mora

Copyright © 2019 Qinghua Gu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most real-world optimization problems tackle a large number of decision variables, known as Large-Scale Global Optimization (LSGO) problems. In general, the metaheuristic algorithms for solving such problems often suffer from the “curse of dimensionality.” In order to improve the disadvantage of Grey Wolf Optimizer when solving the LSGO problems, three genetic operators are embedded into the standard GWO and a Hybrid Genetic Grey Wolf Algorithm (HGGWA) is proposed. Firstly, the whole population using Opposition-Based Learning strategy is initialized. Secondly, the selection operation is performed by combining elite reservation strategy. Then, the whole population is divided into several subpopulations for cross-operation based on dimensionality reduction and population partition in order to increase the diversity of the population. Finally, the elite individuals in the population are mutated to prevent the algorithm from falling into local optimum. The performance of HGGWA is verified by ten benchmark functions, and the optimization results are compared with WOA, SSA, and ALO. On CEC'2008 LSGO problems, the performance of HGGWA is compared against several state-of-the-art algorithms, CCPSO2, DEwSAcc, MLCC, and EPUS-PSO. Simulation results show that the HGGWA has been greatly improved in convergence accuracy, which proves the effectiveness of HGGWA in solving LSGO problems.

## 1. Introduction

Large-Scale Global Optimization (LSGO) is widely applied in practical engineering problems, such as large-scale job shop scheduling problem [1], large-scale vehicle routing problems [2], and reactive power optimization of large-scale power system [3]. It is a very important and challenging task in the optimization domain and usually involves thousands of decision variables. As the number of dimensions increases, the complexity of the problem increases exponentially, so it is very difficult to solve. Metaheuristic algorithms are a class of intelligent optimization algorithms inspired by biological activities or physical principles. In recent years, various metaheuristic algorithms such as genetic algorithms, particle swarm optimization algorithms, artificial bee colony algorithms [4], and differential evolution algorithms [5] have been applied to a variety of large-scale global optimization problems. However, due to the existence of “curse of dimensionality”, it is difficult for general metaheuristic algorithms to find the optimal solution of LSGO problems. In order to

solve LSGO problems better, many valuable attempts have been made in recent years by using metaheuristic algorithms.

In general, solving large-scale problems can be considered in two ways. First, from the perspective of the problem itself, the solution process can be made more efficient by simplifying the problem. Secondly, from the perspective of the method, the optimal solution to the problem can be solved with greater possibility by improving the performance of the solution algorithm. First of all, in terms of the problem itself, a large number of decision variables are the main cause of the complexity of the problem [6]. In the famous book *A Discourse on Method* [7], Descartes pointed out that it is necessary to study complex problems and decompose them into a number of relatively simple small problems and then solve them one by one. He called it a “divide and conquer” strategy, which is to solve the whole problem by decomposing the original large-scale problem into a set of smaller and simpler subproblems which are more manageable and easier to solve and then solve each subproblem independently. This method based on “divide and conquer” strategy is



also called Cooperative Coevolution (CC), and its effectiveness for solving large-scale optimization problems has been demonstrated in many classical optimization methods. But a major difficulty in applying CC is the choice of a good decomposition strategy, because different decomposition strategies may lead to different optimization effects. Due to the diversity of problem-solving, there may be very large differences between different problems. Therefore, it is necessary to study the structure inherent in the problem and analyze the relationship between the variables in order to find a suitable solution.

In addition, from an algorithmic perspective, two or more distinct methods when combined together in a synergistic manner can enhance the problem-solving capability of the derived hybrid [8]. EAs hybridized with local search algorithms have been successful within function optimization domain. These kinds of EAs are often named as memetic algorithms (MAs) [9]. The optimization performance of the algorithm for large-scale optimization problems is improved by different optimization strategies such as designing new mutation operators, dynamic neighborhood search strategies, multiple classifier [10], and Opposition-Based Learning [11]. Similar to evolutionary computation, the swarm intelligence (SI) is a kind of optimization algorithm inspired by the biological foraging or hunting behavior in nature, which simulates the intelligent behavior of insects, bird groups, ant colonies, or fish schools. Inspired by the grey wolf population hierarchy and hunting behavior, Mirjalili proposed another swarm intelligence optimization algorithm, grey wolf optimization algorithm (GWO) in 2014 [12]. The GWO algorithm has the advantages of less control parameters and fast convergence and it has been applied to various optimization fields, such as medical image fusion [13], multiple input multiple output power systems [14], and job shop scheduling problem [15]. In addition, the proposal of different prediction techniques [16] also provides reference for the study of large-scale optimization problems.

To direct at the large-scale global optimization problems, this paper improves the HGGWA algorithm proposed in literature [17] by adding a parameter nonlinear adjustment strategy, which further improves the global convergence and solution accuracy. The crossover operation of the HGGWA algorithm divides the whole population into several subpopulations by referring to the idea of cooperative coevolution, and then independently evolves each subpopulation separately. More specifically, this paper has the following research objectives:

- (1) To review the current literature on solving large-scale global optimization problems and to analyze the existing problems in the research
- (2) To improve the Hybrid Genetic Grey Wolf Algorithm (HGGWA) by adding the nonlinear adjustment strategy of parameter, which further improves the global convergence and solution accuracy for solving large-scale global optimization problems
- (3) To analyze the global convergence and computational complexity of the improved HGGWA algorithm and to verify the effectiveness of HGGWA for solving large-scale

global optimization problems by several different numerical experiments

The remainder of this paper is organized as follows: in Section 2, a review of Large-Scale Global Optimization (LSGO) and various solving strategies is given. Then, Section 3 briefly describes the Grey Wolf Optimizer (GWO). In Section 4, the principles and steps of the improved algorithm (HGGWA) are introduced. Section 5 illustrates and analyzes the experimental results. Finally, the conclusions are drawn in Section 6.

## 2. Literature Review

**2.1. Decomposition-Based CC Strategy.** Large-scale global optimization problems are mainly solved in the following two ways. One is Cooperative Coevolution (CC) with problem decomposition strategy. The idea of decomposition was first proposed by Potter and De Jong [18, 19] in 1994. They designed two Cooperative Coevolutionary Genetic Algorithms (CCGA-1, CCGA-2) to improve the global convergence of GA. The CC strategy is to reduce the scale of the high-dimensional problem by dividing the whole solution space into multiple subspaces and then use multiple subpopulations to achieve the cooperative coevolution. Table 1 summarizes proposed major variants of the CC method. It can be divided into the static grouping-based CC methods and the dynamic grouping-based CC methods.

The static grouping-based CC methods divide the population into multiple fixed-scale subpopulations. The Cooperative Coevolutionary Genetic Algorithm (CCGA) proposed by Potter and De Jong is the first attempt to combine the idea of Cooperative Coevolution with metaheuristic algorithm. They decomposed an  $n$ -dimensional problem into  $n$  1-dimensional subproblems each of which is optimized by a separated GA. Van den Bergh et al. [20] firstly proposed two improved PSO by integrating cooperative coevolution strategy into PSO, called CPSO-SK and CPSO-HK, which divides an  $n$ -dimensional problem into  $k$   $s$ -dimensional subproblems. Similarly, Mohammed EI-Abd [21] presented two cooperative coevolutionary ABC algorithms, namely, CABC-S and CABC-H. Shi et al. [22] applied the cooperative coevolution strategy to the differential evolution algorithm, called the cooperative coevolutionary differential evolution (CCDE). The algorithm partitions high-dimensional solution space into two equal-sized subcomponents but it does not perform well as the dimensionality increases. For a fully separable problem with no interrelationship between variables, each variable can be optimized independently to obtain the optimal solution for the entire problem. However, for the nonseparable problem of the relationship between variables, the optimization effect of the fixed-scale grouping strategy will be worse, and even the optimal solution will not be obtained. Therefore, many scholars began to study the decomposition strategy that can solve the nonseparable problems.

The dynamic grouping-based CC methods dynamically adjust the size of the subpopulation to accommodate different types of large-scale optimization problems. They are efficient to handle nonseparable problems. Yang et al. [23] proposed

TABLE 1: A summary of major variants of the CC method.

Year	Author	Algorithm	Year	Author	Algorithm
1994	Potter and De Jong [18]	CCGA-1, CCGA-2	2011	Omidvar et al. [29]	CBCC
1996	R. Storn [30]	CCDE	2012	Li and Yao [31]	CCPSO2
1999	Weicker, K et al. [32]	CCLVI	2012	Sayed et al. [33]	HDIMA
2001	Liu et al. [34]	FEPCC	2012	Sayed et al. [35]	DIMA
2004	Van den Bergh et al. [20]	CPSO-SK, CPSO-HK	2013	Ren and Wu [36]	CCOABC
2005	Shi et al. [22]	CCDE	2013	Liu and Tang [37]	CC-CMA-ES
2008	Yang et al. [23]	DECC-G	2014	Omidvar et al. [28]	DECC-DG
2008	Yang et al. [38]	MLCC	2014	Omidvar et al. [39]	MLSoft
2009	Li et al. [24]	CCPSO	2014	Mahdavi et al. [40]	DM-HDMR
2009	Ray et al. [26]	CCEA-AVP	2015	Kazimipour et al. [41]	CBCC
2010	Omidvar et al. [25]	DECC-ML	2015	Cao et al. [42]	CC-GLS
2010	Chen et al. [27]	CCVIL	2016	Peng et al. [43]	mCCEA
2010	Singh et al. [44]	CCEA-AVP	2017	Peng et al. [45]	MMO-CC
2010	Mohammed El-Abd [21]	CABC-S, CABC-H	2017	Yang et al. [46]	CCFR
2011	Omidvar et al. [47]	DECC-D	2018	Peng et al. [48]	CC-SMP



a new cooperative coevolution framework that is capable of optimizing large-scale nonseparable problems. It uses a random grouping strategy to divide the problem solution space into multiple subspaces of different sizes. Li et al. [24] embedded a random grouping strategy and an adaptive weighting mechanism into the particle swarm optimization algorithm and proposed a cooperative coevolutionary particle swarm optimization (CCPSO) algorithm. The results of the random grouping method show the effective performance to solve the scalable nonseparable benchmark function (up to 1000D). However, as the number of interaction variables increases, its performance becomes invalid [25]. Therefore, some scholars consider adding a priori knowledge of the problem in the process of algorithm evolution and propose many learning-based dynamic grouping strategies. Ray et al. [26] discussed the effects of problem size, the number of subpopulations, and the number of iterations on some separable and nonseparable problems in cooperative coevolutionary algorithms. Then, a Cooperative Coevolutionary Algorithm with Correlation based Adaptive Variable Partitioning (CCEA-AVP) is proposed to deal with scalable nonseparable problems. Chen et al. [27] proposed a CC method with Variable Interaction Learning (CCVIL) which is able to adaptively change group sizes. Considering that there is no prior knowledge that cannot decompose the problem, Omidvar et al. [28] proposed an automatic decomposition strategy, which can keep the correlation between decision variables at the minimum.

**2.2. Nondecomposition Strategy.** The other is nondecomposition methods. Different from the decomposition-based strategy, the nondecomposition methods mainly study the algorithm itself and improve the corresponding operator in order to improve the performance of the algorithm for solving large-scale global optimization problems. Nondecomposition-based methods mainly include swarm intelligence, evolutionary computation, and local search-based approaches. Hsieh et al. [49] added variable particles and information sharing mechanism to the basic particle swarm optimization algorithm and proposed a variation called the Efficient Population Utilization Strategy for Particle Swarm Optimizer (EPUS-PSO). A modified PSO algorithm with a new velocity updating method, as the rotated particle swarm, was proposed in literature [50]. It transforms coordinates and uses information from other dimensions to maintain the diversity of each dimension. Fan et al. [51] proposed a novel particle swarm optimization approach with dynamic neighborhood that is based on kernel fuzzy clustering and variable trust region methods (called FT-DNPSO) for large-scale optimization. In terms of evolutionary computation, Hedar et al. [52] modified genetic algorithm with new strategies of population partitioning and space reduction for high-dimensional problems. In order to improve the performance of differential evolution algorithm, Zhang et al. [53] proposed an adaptive differential evolution algorithm (called JADE) using a new mutation and adaptive control parameter strategy. Local search-based approaches have been recognized as an effective algorithm framework for solving optimization problems. Hvattum et al. [54] introduced a new

direct search method, based on Scatter Search, designed to remedy the lack of a good derivative-free method for solving problems of high dimensions. Liu et al. [55] improved the local search depth parameter in the memetic algorithm and proposed an adaptive local search depth (ALSD) strategy, which can arrange the local search computing resources according to its performance dynamically.

**2.3. Related Work of GWO.** Grey Wolf Optimizer (GWO) is a recently proposed swarm intelligence algorithm inspired by the social hierarchy and hunting behavior of wolves. It has the advantages of less control parameters, high solution accuracy, and fast convergence speed. Compared with other classical metaheuristic algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), and gravitational search algorithm (GSA), the GWO shows powerful exploration capability and better convergence characteristics. Owing to its simplicity and ease of implementation, GWO has gained significant attention and has been applied in solving many practical optimization problems since its invention.

Recently, many scholars have applied the GWO to different optimization problems and also proposed many varieties in order to improve the performance of GWO. Saremi et al. [56] introduced a dynamic population updating mechanism in the GWO, removing individuals with poor fitness in the population and regenerating new individuals to replace them in order to enhance the exploration ability of GWO. However, the diversity of grey population was reduced by this strategy, which leads to the algorithm being local optimum. Zhu et al. [57] integrated the idea of differential evolution into the GWO, which prevented the early stagnation of the GWO algorithm and accelerated the convergence speed. However, it fails to converge in high-dimensional problems. Kumar et al. [58] designed a novel variant of GWO for numerical optimization and engineering design problems. They utilized the prey weight and astrophysics-based learning concept to enhance the exploration ability. However, this method also does not show good performance in solving high-dimensional function problems. In the application of the GWO algorithm, Guha et al. [59] applied the grey wolf algorithm to the load control problem in large-scale power systems. In literature [60], a multiobjective discrete grey wolf algorithm is proposed for the uniqueness of welding job scheduling problem. The results show that the algorithm can solve the scheduling problem in practical engineering. Emary et al. [61] proposed a new binary grey wolf optimization algorithm for the selection of the best feature subset.

The swarm intelligence optimization algorithm has natural parallelism, so it has advantages in solving large-scale optimization problems. In order to study the ability of grey wolf optimization algorithm to solve high-dimensional functions, Long et al. [62] proposed a nonlinear adjustment strategy for convergence factors, which balanced well the exploration and exploitation of the algorithm. But this strategy also does not solve the high-dimensional function optimization problem well. In 2017, Gupta et al. [63] studied the performance of the GWO algorithm for solving 100 to 1000-dimensional function optimization problems. The results indicate that

GWO is a powerful nature-inspired optimization algorithm for large-scale problems, except Rosenbrock function.

In summary, the decomposition-based CC strategy has a good effect in solving the nonseparable problem, but it cannot solve the imbalance problem well, and it still needs to be improved in the accuracy of the solution. In addition, in the method based on nondecomposition strategy, although the performance of many algorithms has been greatly improved, but the research on hybrid algorithms is still rare. Therefore, this paper combines three genetic operators of selection, crossover, and mutation with GWO algorithm and proposes a Hybrid Genetic Grey Wolf Algorithm, (HGGWA). The main improvement strategies of the algorithm are as follows: (1) initialize the population based on the Opposition-Based Learning strategy; (2) the nonlinear adjustment of parameters effectively balances exploration and exploitation capabilities while accelerating the convergence speed of the algorithm; (3) select the population through a combination of elite retention strategy and gambling roulette; (4) based on the method of dimensionality reduction and population division, the whole population is divided into multiple subpopulations for cross-operation to increase the diversity of the population; (5) perform mutation operations on elite individuals in the population to prevent the algorithm from falling into local optimum.

### 3. Grey Wolf Optimization Algorithm

**3.1. Mathematical Models.** In the GWO algorithm [12], the position of  $\alpha$  wolf,  $\beta$  wolf, and  $\delta$  wolf is the fittest solution, the second-best solution, and the third-best solution, respectively. And the positions of the  $\omega$  wolves are the remaining candidate solutions. In the whole process of searching for prey in the space, the  $\omega$  wolves gradually update their positions according to the optimal position of the  $\alpha$ ,  $\beta$ , and  $\delta$  wolves. So they can gradually approach and capture the prey, that is, to complete the process of searching for the optimal solution. The position updating imaginary diagram in GWO is shown in Figure 1.

The grey wolves position update formula in this process is as follows:

$$\vec{D} = \left| \vec{C} * \vec{X}_p(t) - \vec{X}(t) \right| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} * \vec{D} \quad (2)$$

where  $\vec{D}$  indicates the distance between the grey wolf individual and the prey,  $t$  indicates the current iteration,  $\vec{A}$  and  $\vec{C}$  are the coefficient vectors,  $\vec{A}$  is a convergence coefficient used to balance the exploration and the exploitation,  $\vec{C}$  is used to simulate the effects of nature,  $\vec{X}_p$  is the position vector of the prey, and  $\vec{X}$  indicates the position vector of a grey wolf.

The coefficient vectors  $\vec{A}$  and  $\vec{C}$  are calculated as follows:

$$\vec{A} = 2\vec{a} * \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 * \vec{r}_2 \quad (4)$$

where components of  $\vec{a}$  are linearly decreased from 2 to 0 over the course of iterations and  $r_1$  and  $r_2$  are random vectors in  $[0, 1]$ . At the same time, the fluctuation range of the coefficient vector  $\vec{A}$  gradually decreases as  $\vec{a}$  decreases.

Thus, according to (1) and (2), the grey wolf individual randomly moves within the search space to gradually approach the optimal solution. The specific location update formula is as follows:

$$\vec{D}\alpha = \left| \vec{C}1 * \vec{X}\alpha - \vec{X} \right|, \quad (5)$$

$$\vec{D}\beta = \left| \vec{C}2 * \vec{X}\beta - \vec{X} \right|, \quad (6)$$

$$\vec{D}\delta = \left| \vec{C}3 * \vec{X}\delta - \vec{X} \right|, \quad (7)$$

$$\vec{X}1 = \vec{X}\alpha - \vec{A}1 * \vec{D}\alpha, \quad (8)$$

$$\vec{X}2 = \vec{X}\beta - \vec{A}2 * \vec{D}\beta, \quad (9)$$

$$\vec{X}3 = \vec{X}\delta - \vec{A}3 * \vec{D}\delta, \quad (10)$$

$$\vec{X}(t+1) = \frac{\vec{X}1 + \vec{X}2 + \vec{X}3}{3} \quad (11)$$

where  $\vec{D}\alpha$ ,  $\vec{D}\beta$ ,  $\vec{D}\delta$  indicate the distance between the  $\alpha$ ,  $\beta$ ,  $\delta$  wolves and the prey, respectively.  $\vec{X}1$ ,  $\vec{X}2$ ,  $\vec{X}3$  represent the positional components of the remaining grey wolves based on the positions of  $\alpha$ ,  $\beta$ , and  $\delta$  wolves, respectively.  $\vec{X}(t+1)$  represents the position vector after the grey wolf is updated.

#### 3.2. Basic Process of GWO.

**Step 1.** Randomly generate an initial population, and initialize parameters  $\vec{a}$ ,  $\vec{A}$ , and  $\vec{C}$ .

**Step 2.** Calculate the fitness of the grey wolf individual, and save the positions of the first three individuals with the highest fitness as  $\vec{X}_\alpha$ ,  $\vec{X}_\beta$ , and  $\vec{X}_\delta$ .

**Step 3.** Update the position information of each grey wolf according to (5) to (11), thereby obtaining the next generation population, and then update the values of parameters  $\vec{a}$ ,  $\vec{A}$ , and  $\vec{C}$ .

**Step 4.** Calculate the fitness of individuals in the new population, and update  $\vec{X}_\alpha$ ,  $\vec{X}_\beta$ , and  $\vec{X}_\delta$ .

**Step 5.** Repeat Steps 2–4 until the optimal solution is obtained or the maximum number of iterations is reached.

### 4. Hybrid Genetic Grey Wolf Algorithm

This section describes the details of HGGWA, the hybrid algorithm proposed in this paper. Firstly, the initial population strategy based on Opposition-Based Learning is

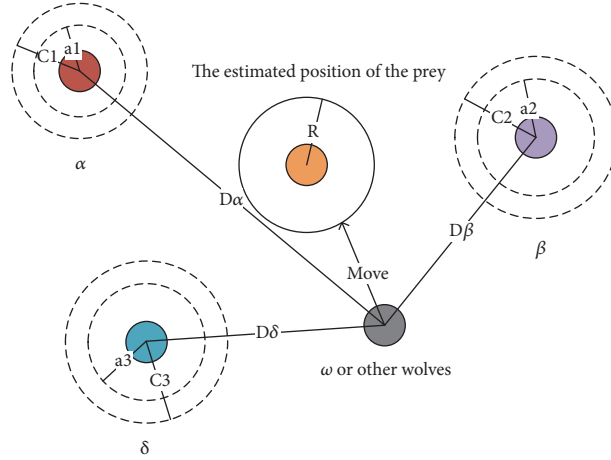


FIGURE 1: Position updating in GWO.

described in detail in Section 4.1. Secondly, Section 4.2 introduces the nonlinear adjustment strategy for parameter  $a$ . Then three operations of selection, crossover, and mutation are carried out in Sections 4.3–4.5, respectively. Finally, the time complexity of HGGWA is analyzed in Section 4.6.

**4.1. Initial Population Strategy Based on Opposition-Based Learning.** For the swarm intelligence optimization algorithm based on population iteration, the diversity of initial populations lays the foundation for the efficiency of the algorithm. The better diversity of the population will reduce the computation time and improve the global convergence of the algorithm [64]. However, like other algorithms, the GWO algorithm also uses random initialization when generating populations, which will have a certain impact on the search efficiency of the algorithm. Opposition-Based Learning is a strategy proposed by Tizhoosh [65] to improve the efficiency of algorithm search. It uses the opposite points of known individual positions to generate new individual positions, thereby increasing the diversity of search populations. Currently, the Opposition-Based Learning strategy has been successfully applied to multiple swarm optimization algorithms [66, 67]. The opposite point is defined as follows.

Assume that there is an individual  $X (x_1, x_2, \dots, x_D)$  in the population  $P$ , then the opposite point of the element  $x_i$  ( $i = 1, 2, \dots, D$ ) on each dimension is  $x'_i = l_i + u_i - x_i$ , where  $l_i$  and  $u_i$  are the lower bound and the upper of the  $i$ th dimension, respectively. According to the above definition, the Opposition-Based Learning strategy initializes the population as follows:

(a) Randomly initialize the population  $P$ , calculate the opposite point  $x'_i$  of each individual position  $x_i$ , and all the opposite points constitute the opposition population  $P'$ .

(b) Calculate the fitness of each individual in the initial population  $P$  and the opposition population  $P'$ , and arrange them in descending order.

(c) Select the top  $N$  grey wolf individuals with the highest fitness as the final initial population.

**4.2. Parameters Nonlinear Adjustment Strategy.** It is significantly important how to balance the exploration and exploitation for swarm intelligence algorithms. In the early iteration of the algorithm, the powerful exploration capability is beneficial for the algorithm to expand the search range, so that the algorithm can search for the optimal solution with greater probability. In the later stage of the algorithm's iteration, the effective exploitation ability can speed up the optimization process of the algorithm and improve the convergence accuracy of the final solution. Therefore, only when the swarm optimization algorithm can better coordinate its global exploration and local exploitation capabilities can it have strong robustness and faster convergence speed.

It can be seen from (2) and (3) that the dynamic change of the parameter  $a$  plays a crucial role in the algorithm. In the basic GWO, the linear decrement strategy of parameter  $a$  does not reflect the actual convergence process of the algorithm. Therefore, the algorithm does not balance the exploration and exploitation capabilities well, especially when solving large-scale multimodal function problems. In order to dynamically adjust the global exploration and local exploitation process of the algorithm, this paper proposes a nonlinear adjustment strategy for the parameter  $a$ , which is beneficial for HGGWA to search for the optimal solution. The improved parameter nonlinear adjustment equation is as follows:

$$a = a_{max} + (a_{min} - a_{max}) \cdot \left( \frac{1}{1 + e^{t/Max_{iter}}} \right)^k \quad (12)$$

where  $a_{max}$  and  $a_{min}$  are the initial value and the end value of the parameter  $a$ ,  $t$  is the current iteration index,  $Max_{iter}$  is the maximum number of iterations, and  $k$  is nonlinear adjustment coefficient.

Thus, the variation range of the convergence factor  $A$  is controlled by the nonlinear adjustment of the parameter  $a$ . When  $|A| > 1$ , the grey wolf population expands the search range to find better prey, which corresponds to the global exploration of the algorithm; when  $|A| < 1$ ,

the whole population shrinks the search range; thus, an encirclement is formed around the prey to complete the final attacking against the prey, which corresponds to the local exploitation process of the algorithm. The whole process has a positive effect on balancing global search and local search, which is beneficial to improve the accuracy of the solution and further accelerate the convergence speed of the algorithm.

**4.3. Selection Operation Based on Optimal Retention.** For the intelligent optimization algorithm based on population evolution, the evolution of each generation of population directly affects the optimization effect of the algorithm. In order to inherit the superior individuals in the paternal population to the next generation without being destroyed, it is necessary to preserve the good individuals directly. The optimal retention selection strategy is an effective way to preserve good individuals in genetic algorithms. This paper integrates them into the GWO to improve the efficiency of the algorithm. Assuming that the current population is  $P(X_1, X_2, \dots, X_D)$ , the fitness of the individual  $X_i$  is  $F_i$ . The specific operation is as follows:

(a) Calculate the fitness  $F_i$  ( $i = 1, 2, \dots, D$ ) of each individual and arrange them in descending order.

(b) Selecting the individuals with the highest fitness to copy directly into the next generation of population.

(c) Calculate the total fitness  $S$  of the remaining individuals and the probability  $p_i$  that each individual is selected.

$$S = \sum_{i=1}^{D-1} F_i \quad (i = 1, 2, \dots, D-1) \quad (13)$$

$$p_i = \frac{F_i}{\sum_{i=1}^{D-1} F_i} \quad (i = 1, 2, \dots, D-1) \quad (14)$$

(d) Calculate the cumulative fitness value  $s_i$  for each individual, and then the selection operation is performed in the manner of the bet roulette until the number of individuals in the children population is consistent with the parent population.

$$s_i = \frac{\sum_{i=1}^i F_i}{S} \quad (i = 1, 2, \dots, D-1) \quad (15)$$

**4.4. Crossover Operation Based on Population Partitioning Mechanism.** Due to the decrease of population diversity in the late evolution, the GWO algorithm is easy to fall into local optimum for solving the large-scale high-dimensional optimization problem, in order to overcome the problems caused by large-scale and complexity and to ensure that the algorithm searches for all solution spaces.

In the HGGWA algorithm, the whole population  $P$  is divided into  $v \times \eta$  subpopulations  $P_{i,j}$  ( $i = 1, \dots, v; j = 1, \dots, \eta$ ). The optimal size of each subpopulation  $P_{i,j}$  was tested to be  $5 \times 5$ . Individuals of each subpopulation were

cross-operated to increase the diversity of the population. The specific division method is shown below.

The initial population:  $P$

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p,1} & x_{p,2} & \cdots & x_{p,d} \end{bmatrix}$$

Subpopulations:  $P_{i,j}$

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,5} \\ \vdots & \ddots & \vdots \\ x_{5,1} & \cdots & x_{5,5} \end{bmatrix} \cdots \begin{bmatrix} x_{1,(d-4)} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{5,(d-4)} & \cdots & x_{5,d} \end{bmatrix} \quad (16)$$

$P_{1,1} \quad \vdots \quad \cdots \quad P_{1,\eta}$

$$\begin{bmatrix} x_{(p-4),1} & \cdots & x_{(p-4),5} \\ \vdots & \ddots & \vdots \\ x_{p,1} & \cdots & x_{p,5} \end{bmatrix} \cdots \begin{bmatrix} x_{(p-4),(d-4)} & \cdots & x_{(p-4),d} \\ \vdots & \ddots & \vdots \\ x_{p,(d-4)} & \cdots & x_{p,d} \end{bmatrix}$$

$P_{v,1} \quad \cdots \quad P_{v,\eta}$

In genetic algorithms, cross-operation has a very important role and is the main way to generate new individuals. In the improved algorithm of this paper, each individual in the subpopulation is cross-operated in a linear crossover manner. Generating a corresponding random number  $r_i \in (0,1)$  for each individual  $x_i$  in the subpopulation. When the random number  $r_i$  is less than the crossover probability  $P_c$ , the corresponding individual  $x_i$  is paired for cross-operation.

An example is as follows: Crossover ( $p_1, p_2$ )

(a) Generate a random number  $\lambda \in (0, 1)$ .

(b) Two children  $c^1(c_1^1, \dots, c_D^1)$ ,  $c^2(c_1^2, \dots, c_D^2)$  are generated by two parents  $p^1(p_1^1, \dots, p_D^1)$  and  $p^2(p_1^2, \dots, p_D^2)$ :

$$c_i^1 = \lambda p_i^1 + (1 - \lambda) p_i^2, \quad i = 1, 2, \dots, D \quad (17)$$

$$c_i^2 = \lambda p_i^2 + (1 - \lambda) p_i^1, \quad i = 1, 2, \dots, D \quad (18)$$

**4.5. Mutation Operation for Elite Individuals.** Due to the existence of the selection operation based on the optimal preservation strategy, the grey wolf individuals in the whole population are concentrated in a small optimal region in the later stage of the iterative process, which easily leads to the loss of population diversity. If the current optimal individual is a locally optimal solution, then the algorithm is easy to fall into local optimum, especially when solving high-dimensional multimodal functions. To this end, this paper introduces the mutation operator in the HGGWA algorithm to perform mutation operations on elite individuals in the population. The specific operations are as follows.

Assume that the optimal individual is  $x_i(x_1, x_2, \dots, x_d)$  and the mutation operation is performed on  $x_i$  with the mutation probability  $P_m$ . That is, select a gene  $x_k$  from the optimal individual with probability  $P_m$ , instead of the gene  $x_k$  with a random number between upper and lower bounds to



```

(1) Initialize parameters  $a, A, C$ , population size  $N$ ,  $P_c$  and  $P_m$ 
(2) Initialize population  $P$  using OBL the strategy
(3)  $t=0$ 
(4) While  $t < Max_{iter}$ 
(5)   Calculate the fitness of all search agents
(6)    $X_\alpha$  = the best search agent
(7)    $X_\beta$  = the second best search agent
(8)    $X_\delta$  = the third best search agent
(9)   for  $i = 1:N$ 
(10)    Update the position of all search agents by equation (11)
(11)   end for
(12)    $newP1 \leftarrow P$  except the  $X_\alpha$ 
(13)   for  $i = 1:N-1$ 
(14)    Generate  $newP2$  by the Roulette Wheel Selection on  $newP1$ 
(15)   end for
(16)    $P \leftarrow newP2$  and  $X_\alpha$ 
(17)   Generate  $subPs$  by the population partitioning mechanism on  $P$ 
(18)   Select the individuals by crossover probability  $P_c$ 
(19)   for  $i = 1:N * P_c$ 
(20)    Crossover each search agent by equation (17) and (18)
(21)   end for
(22)   Generate the  $X'_\alpha$ ,  $X'_\beta$  and  $X'_\delta$  by equation (19) on mutation probability  $P_m$ 
(23)    $t = t+1$ 
(24) end while
(25) Return  $X_\alpha$ 

```

ALGORITHM 1: Pseudo code of the HGGWA.

generate a new individual  $x'_i = (x'_1, x'_2, \dots, x'_d)$ . The specific operation is as follows:

$$x'_i = \begin{cases} l + \lambda * (u - l) & i = k \\ x_i & i \neq k \end{cases} \quad (19)$$

where  $\lambda$  is a random number in  $[0, 1]$  and  $l$  and  $u$  are the lower and upper bounds of the individual  $x_i$ , respectively.

**4.6. Pseudo Code of the HGGWA and Time Complexity Analysis.** This section describes how to calculate an upper bound for the total number of fitness evaluations (FE) required by HGGWA. As shown in Algorithm 1, the computational complexity of HGGWA in one generation is mainly dominated by the position updating of all search agents in line (10). The position updating requires a time complexity of  $O(N * dim * Max_{iter})$  ( $N$  is the population size,  $dim$  is the dimensions of each benchmark function, and  $Max_{iter}$  is the maximum number of iterations of HGGWA) to obtain the needed parameters and to finish the whole position updating progress of all search agents. In addition, the process of crossover operations is another time-consuming step in line (20). It needs a time complexity of  $O(2 * v * \eta * Max_{iter})$  ( $v * \eta$  is the total number of subpopulations,  $v = N/5$ ,  $\eta = dim/5$ ) to complete the crossing of individuals in each subpopulations according to the crossover probability  $P_c$ . It should be noted that this is only the worst case computational complexity. If there are only two individuals in each subpopulation that need to be crossed, then the minimum amount of computation is  $O(v * \eta * Max_{iter})$ .

Therefore, the maximum computational complexity caused by the two dominant processes in the algorithm is  $\max\{O(N * dim * Max_{iter}), O(2 * v * \eta * Max_{iter})\}$  ( $v = N/5$ ,  $\eta = dim/5$ ) in one generation, i.e.,  $O(N * dim * Max_{iter})$ .

## 5. Numerical Experiments and Analysis

In this section, the proposed HGGWA algorithm will be evaluated on both classical benchmark functions [68] and the suite of test functions provided by CEC2008 Special Session on large-scale global optimization. The algorithms used for comparison include not only conventional EAs but also other CC optimization algorithms. Experimental results are provided to analyze the performance of HGGWA in the context of large-scale optimization problems. In addition, the sensitivity analysis of nonlinear adjustment coefficient and the global convergence of HGGWA are discussed in Sections 5.4 and 5.5, respectively.

**5.1. Benchmark Functions and Performance Measures.** In order to verify the ability of HGGWA algorithm to solve high-dimensional complex functions, 10 general high-dimensional benchmark functions (100D, 500D, 1000D) were selected for optimization test. Among them,  $f_1 \sim f_6$  are unimodal functions, which are used to test the local searchability of the algorithm;  $f_7 \sim f_{10}$  are multimodal functions, which are used to test the global searchability of the algorithm. These test functions have multiple local optima points with uneven distribution, nonconvexity, and strong oscillation, which are very difficult to converge to the global optimal solution, especially in the case of being high-dimensional. The specific

TABLE 2: 10 high-dimensional benchmark functions.

Functions	Function formula	Range	$f_{min}$	Accuracy
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	$[-100, 100]$	0	$1 \times 10^{-8}$
Schwefel's 2.22	$f_2(x) = \sum_{i=1}^d  x_i  + \prod_{i=1}^d  x_i $	$[-10, 10]$	0	$1 \times 10^{-8}$
Schwefel's 2.21	$f_3(x) = \max_i \{ x_i , 1 \leq i \leq d\}$	$[-100, 100]$	0	$1 \times 10^{-2}$
Rosenbrock	$f_4(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0	$1 \times 10^0$
Schwefel's 1.2	$f_5(x) = \sum_{i=1}^d \left( \sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0	$1 \times 10^{-3}$
Quartic	$f_6(x) = \sum_{i=1}^d ix_i^4 + \text{random}[0, 1]$	$[-1.28, 1.28]$	0	$1 \times 10^{-4}$
Rastrigin	$f_7(x) = \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0	$1 \times 10^{-8}$
Ackley	$f_8 = -20 \exp \left( -0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]$	0	$1 \times 10^{-5}$
Griewank	$f_9 = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]$	0	$1 \times 10^{-5}$
	$f_{10} = \frac{\pi}{d} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] + (y_d - 1)^2 \right\}$ $+ \sum_{i=1}^d u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$			
Penalized 1	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0 & \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]$	0	$1 \times 10^{-2}$

characteristics of the 10 benchmark functions are shown in Table 2.

In order to show the optimization effect of the algorithm more clearly, this paper selects two indicators to evaluate the performance of the algorithm [69]. The first is the solution accuracy (AC), which reflects the difference between the optimal results of the algorithm and the theoretical optimal value. In an experiment, if the final convergence result of the algorithm is less than the AC, then the optimization is considered successful. Assuming that the optimal value of a certain optimization is  $X_{opt}$  and the theoretical optimal value is  $X_{min}$ , then the AC is calculated as follows:

$$AC = |X_{opt} - X_{min}| \quad (20)$$

The second is the successful ratio (SR), which reflects the proportion of the number of successful optimizations to the total number of optimizations under the condition that the algorithm has a certain accuracy. Assume that in the  $T$  iterations of the algorithm, the number of successful optimizations is  $t$ , then the SR is calculated as follows:

$$SR = \frac{t}{T} \times 100\% \quad (21)$$

**5.2. Results on Classical Benchmark Functions.** In this section, a set of simulation tests were performed in order to verify the effectiveness of HGGWA. Firstly, the 10 high-dimensional benchmark functions in Table 2 are optimized by HGGWA algorithm proposed in this paper, and the results are compared with WOA, SSA, and ALO. In addition, the running time of several algorithms is compared under the given convergence accuracy.

**5.2.1. Parameters Settings for the Compared Algorithms.** In all experiments, the values of the common parameters, such as the maximum number of iterations ( $Max_{iter}$ ), the dimension of the functions (D), and the population sizes (N) were chosen the same. For all test problems, we focus on investigating the optimization performance of the proposed method on problems with D = 100, 500, and 1000. The maximum number of iterations is set to 1000, and the population size is set to 50. The parameter setting of WOA, SSA, and ALO is derived from the original literature [70–72]. The optimal parameter settings of the HGGWA algorithm proposed in this paper are shown in Table 3.

For each experiment of an algorithm on a benchmark function, 30 independent runs are performed to obtain a

TABLE 3: Parameter settings of the HGGWA.

Parameters	Definition	Value
N	Population size	50
$P_c$	Cross probability	0.8
$P_m$	Mutation probability	0.01
$Max_{iter}$	Maximum number of iterations	1000
$k$	Nonlinear adjustment coefficient	0.5

fair comparison among the different algorithms. The optimal value, the worst value, the average value, and the standard deviation of each algorithm optimization are recorded, and then the optimization successful ratio (SR) is calculated. All programs were coded in Matlab 2017b (Win64) and executed on a Lenovo computer with Intel (R) Core I5-6300HQ, 8G ROM, 2.30GHz under Windows 10 operating system.

**5.2.2. Comparison with State-of-the-Art Metaheuristic Algorithms.** In order to verify the efficiency of the HGGWA algorithm, several state-of-the-art metaheuristic algorithms recently proposed are compared with their optimization results. These algorithms are Whale Optimization Algorithm (WOA), Salp Swarm Algorithm (SSA), and Ant Lion Optimization Algorithm (ALO). The test was performed using the same 10 high-dimensional functions in Table 2, and the results of the comparison are shown in Table 4.

From Table 4, it can be known that the convergence accuracy and optimization successful ratio of the HGGWA algorithm are significantly higher than the other three algorithms for most of the test functions. In the case of 100 dimensions, HGGWA algorithm can converge to the global optimal solution for the other nine test functions at one time in addition to function 3. And the algorithm has a successful ratio of 100% for the nine functions under the condition of certain convergence accuracy. The convergence result of the WOA algorithm is better than the other three algorithms for functions  $f_1$  and  $f_2$ , and its robustness is excellent. With the increase of function dimension, the convergence precision of several algorithms decreases slightly, but the optimization results of HGGWA algorithm are better than those of WOA, SSA, and ALO. In general, the HGGWA algorithm exhibits better optimization effects than the WOA, SSA, and ALO algorithms in the 100-, 500-, and 1000-dimensional test functions, which proves the effectiveness of the HGGWA algorithm for solving high-dimensional complex functions.

Generally speaking, the HGGWA algorithm performs better than other algorithms in most test functions. In order to compare the convergence performance of the four algorithms more clearly, the convergence curves of the four algorithms on Sphere, Schwefel's 2.22, Rastrigin, Griewank, Quartic, and Ackley functions with  $D=100$  are shown in Figure 2. Sphere and Schwefel's 2.22 represent the two most basic unimodal functions. Rastrigin and Griewank represent the two most basic multimodal functions. It is seen that HGGWA has higher precision and faster convergence speed than other algorithms.

**5.2.3. Comparative Analysis of Running Time.** To evaluate the actual runtime of the several compared algorithms, including SSA, WOA, GWO, and HGGWA, their average running times (in seconds: s) from 30 runs on function  $f_1, f_2, f_6, f_7, f_8, f_9$  and  $f_{10}$  are plotted in Figure 3. The convergence accuracy of each test function has been described in Table 2.

Figure 3 shows the convergence behavior of different algorithms. Each point on the plot was calculated by taking the average of 30 independent runs. As can be seen from Figure 3, several algorithms show different effects under the same calculation accuracy. For the functions  $f_1, f_2, f_7, f_8, f_9$ , the GWO algorithm has the fastest convergence rate as a result of the advantages of the GWO algorithm itself. However, for the test functions  $f_6$  and  $f_{10}$ , the running time of the HGGWA algorithm is the shortest. Through further analysis, it is known that HGGWA algorithm adds a little computing time compared to GWO algorithm because of several improved strategies such as selection, crossover, and mutation. However, the convergence speed of HGGWA algorithm is still better than SSA and WOA algorithms. Overall, the convergence speed of the HGGWA algorithm performs better in several algorithms compared. Moreover, the running time of different test functions is very small, and it is kept between 1s and 3s, which shows that the HGGWA algorithm has excellent stability.

**5.3. Results on CEC'2008 Benchmark Functions.** This section provides an analysis of the effectiveness of HGGWA in terms of the CEC'2008 large-scale benchmark functions and a comparison with four powerful algorithms which have been proven to be effective in solving large-scale optimization problems. Experimental results are provided to analyze the performance of HGGWA for large-scale optimization problems.

We tested our proposed algorithms with CEC'2008 functions for 100, 500, and 1000 dimensions and the means of the best fitness value over 50 runs were recorded. While F1 (Shifted Sphere), F4 (Shifted Rastrigin), and F6 (Shifted Ackley) are separable functions, F2 (Schwefel Problem), F3 (Shifted Rosenbrock), F5 (Shifted Griewank), and F7 (Fast Fractal) are nonseparable, presenting a greater challenge to any algorithm that is prone to variable interactions. The performance of HGGWA is compared with other algorithms CCPSO2 [31], DEwSAcc [5], MLCC [38], and EPUS-PSO [49] for 100, 500, and 1000 dimensions, respectively. The maximum number of fitness evaluations (FEs) was calculated by the following formula,  $FEs = 5000 \times D$ , where  $D$  is the number of dimensions. In order to reflect the fairness of the comparison results, the optimization results of the other four algorithms are directly derived from the original literature. The specific comparison results are shown in Table 5.

Table 5 shows the experimental results and the entries shown in bold are significantly better than other algorithms. A general trend that can be seen is that HGGWA algorithm has a promising optimization effect for most of the 7 functions.

Generally speaking, HGGWA outperforms CCPSO2 on 6 out of 7 functions with 100 and 500 dimensions, respectively. Among them, the result of HGGWA is slightly worse than

TABLE 4: Comparison results of 10 high-dimensional benchmark functions by WOA, SSA, ALO, and HGGWA.

Function	Dim	WOA			SSA			ALO			HGGWA		
		Mean	Std	SR	Mean	Std	SR	Mean	Std	SR	Mean	Std	SR
$f_1$	100	<b>5.77E-96</b>	<b>4.69E-96</b>	100%	1.01E-02	1.24E-02	0	3.04E-01	2.31E-01	0	4.85E-53	2.41E-52	100%
	500	<b>7.24E-93</b>	<b>2.98E-94</b>	100%	3.24E+04	2.18E+04	0	7.34E+04	1.32E+04	0	8.35E-25	8.86E-25	100%
	1000	<b>2.05E-89</b>	<b>3.16E-90</b>	100%	1.19E+05	1.04E+05	0	2.48E+05	2.12E+05	0	1.75E-17	2.73E-17	100%
$f_2$	100	<b>1.89E-99</b>	<b>2.33E-99</b>	100%	8.85E+00	2.37E+00	0	4.29E+02	3.25E+02	0	4.01E-33	4.34E-33	100%
	500	<b>1.38E-99</b>	<b>2.16E-99</b>	100%	3.37E+02	2.17E+02	0	2.31E+03	2.03E+03	0	9.58E-18	2.36E-18	100%
	1000	<b>1.80E-99</b>	<b>1.37E-98</b>	100%	8.45E+02	3.43E+02	0	3.25E+04	4.13E+04	0	2.50E-11	1.66E-11	100%
$f_3$	100	3.84E+01	2.17E+00	0	<b>2.22E+01</b>	<b>1.05E+01</b>	0	2.47E+01	1.34E+01	0	4.55E+01	3.69E+01	0
	500	9.32E+01	5.32E+01	0	<b>3.27E+01</b>	<b>2.33E+01</b>	0	3.92E+01	2.34E+01	0	7.22E+01	5.01E+00	0
	1000	9.86E+01	3.24E+01	0	<b>3.31E+01</b>	<b>1.28E+01</b>	0	4.93E+01	3.99E+01	0	8.31E+01	4.36E+00	0
$f_4$	100	9.67E+01	3.98E+01	0	9.00E+02	5.16E+02	0	1.40E+03	1.03E+03	0	<b>6.82E+01</b>	<b>1.25E-01</b>	100%
	500	4.95E+02	3.14E+01	0	7.18E+06	1.46E+06	0	2.09E+07	1.37E+07	0	<b>3.67E+02</b>	<b>8.23E-01</b>	0
	1000	9.91E+02	2.48E+02	0	2.98E+07	1.35E+07	0	1.52E+08	1.00E+07	0	<b>9.64E+02</b>	<b>3.74E+01</b>	0
$f_5$	100	7.83E+05	2.36E+05	0	2.17E+04	1.03E+04	0	3.60E+04	2.16E+04	0	<b>3.45E-05</b>	<b>2.77E-05</b>	100%
	500	1.98E+07	1.32E+07	0	4.29E+05	3.16E+05	0	1.18E+06	1.03E+06	0	<b>2.67E-03</b>	<b>4.38E-03</b>	80%
	1000	6.68E+07	3.44E+07	0	2.48E+06	1.23E+06	0	3.72E+06	3.49E+06	0	<b>8.33E+00</b>	<b>5.12E+00</b>	0
$f_6$	100	2.65E-04	1.43E-04	100%	1.12E+00	1.03E+00	0	8.28E-01	2.68E-01	0	<b>2.34E-09</b>	<b>3.17E-10</b>	100%
	500	3.94E-04	3.22E-04	60%	4.97E+01	3.66E+01	0	1.13E+02	1.01E+02	0	<b>5.73E-05</b>	<b>4.33E-06</b>	100%
	1000	2.16E-03	1.36E-03	0	3.93E+02	2.78E+02	0	1.77E+03	2.33E+03	0	<b>1.08E-04</b>	<b>1.62E-03</b>	80%
$f_7$	100	<b>0.00E+00</b>	<b>0.00E+00</b>	100%	6.27E+01	4.36E+01	0	3.54E+02	3.32E+02	0	<b>0.00E+00</b>	<b>0.00E+00</b>	100%
	500	<b>0.00E+00</b>	<b>0.00E+00</b>	100%	2.04E+03	1.97E+03	0	3.34E+03	4.56E+03	0	<b>0.00E+00</b>	<b>0.00E+00</b>	100%
	1000	<b>0.00E+00</b>	<b>0.00E+00</b>	100%	5.69E+03	1.66E+03	0	7.19E+03	2.14E+03	0	<b>5.57E-11</b>	<b>3.15E-12</b>	100%
$f_8$	100	4.44E-15	3.12E-15	100%	5.15E+00	4.13E+00	0	4.93E+00	3.49E+00	0	<b>2.15E-15</b>	<b>3.48E-15</b>	100%
	500	<b>4.44E-15</b>	<b>3.05E-15</b>	100%	1.26E+01	1.00E+01	0	1.28E+01	2.28E+01	0	8.74E-12	5.39E-12	100%
	1000	<b>8.88E-16</b>	<b>2.88E-15</b>	100%	1.28E+01	1.44E+01	0	1.46E+01	2.03E+01	0	1.59E-07	1.63E-07	100%
$f_9$	100	<b>0.00E+00</b>	<b>0.00E+00</b>	100%	1.52E-01	2.44E-01	0	4.54E-01	3.16E-01	0	<b>0.00E+00</b>	<b>0.00E+00</b>	100%
	500	<b>0.00E+00</b>	<b>0.00E+00</b>	100%	2.82E+02	2.13E+02	0	3.92E+02	1.08E+02	0	1.84E-16	8.44E-17	100%
	1000	<b>0.00E+00</b>	<b>0.00E+00</b>	100%	1.10E+03	1.34E+03	0	3.61E+03	1.17E+03	0	2.31E-13	3.29E-14	100%
$f_{10}$	100	8.16E-03	4.32E-03	100%	1.10E+01	2.19E+01	0	1.74E+01	1.33E+01	0	<b>2.73E-07</b>	<b>3.11E-07</b>	100%
	500	2.52E-02	1.99E-02	80%	5.16E+01	3.33E+01	0	3.97E+05	2.96E+05	0	<b>8.12E-05</b>	<b>4.98E-06</b>	100%
	1000	1.43E-02	1.03E-02	80%	1.00E+05	2.13E+04	0	6.96E+06	1.34E+06	0	<b>6.33E-04</b>	<b>3.77E-03</b>	100%



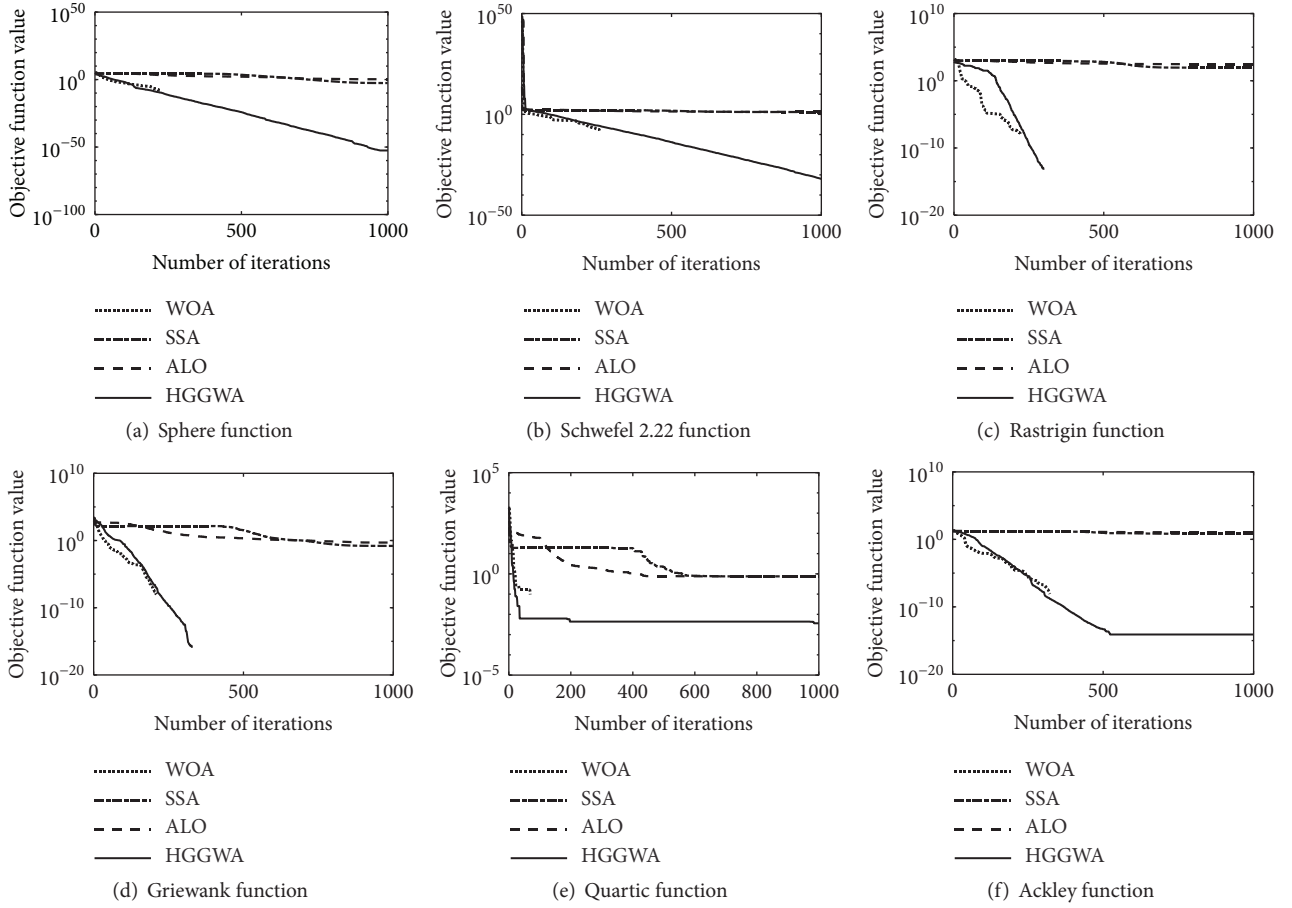


FIGURE 2: Convergence curves of WOA, SSA, ALO, and HGGWA algorithms.

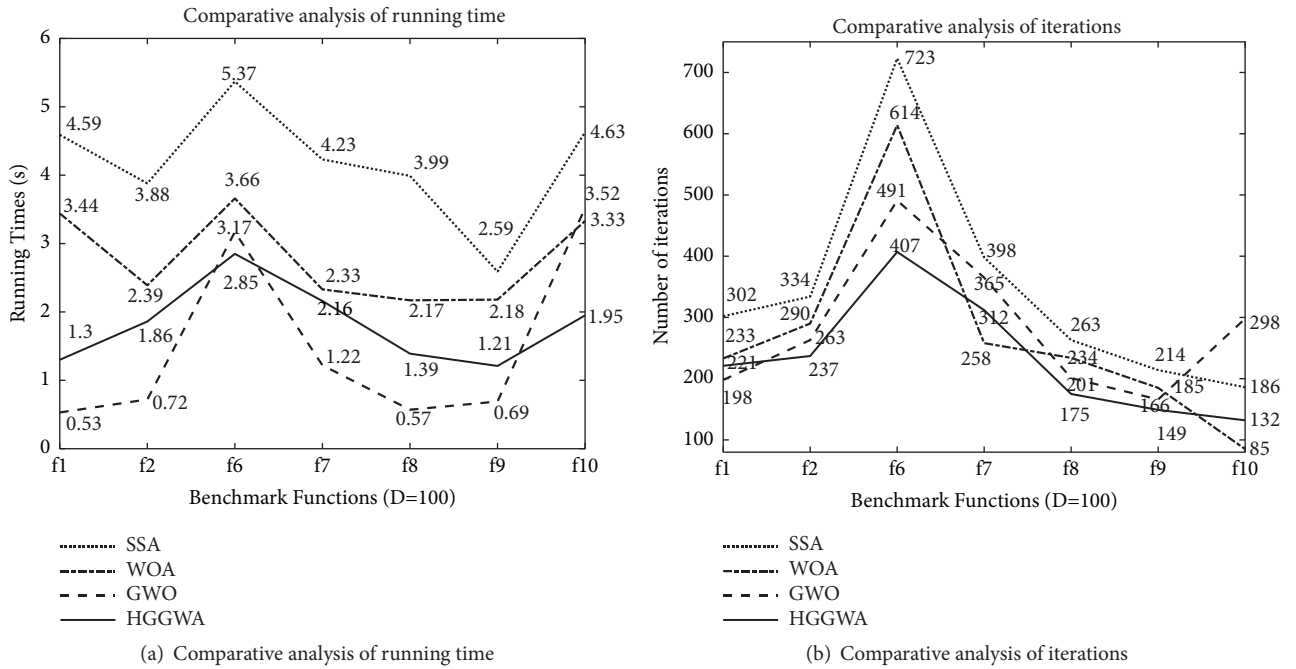


FIGURE 3: The average running times and iterations of SSA, WOA, GWO, and HGGWA on different test functions.

TABLE 5: Comparison results of 7 CEC'2008 benchmark functions by five algorithms.

Function	Dim	CCPSO2			DEwSAcc			MLCC			EPUS-PSO			HGGWA		
		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std	
F1	100	773E-14	3.23E-14		5.68E-14	0.00E+00		6.82E-14	2.32E-14		7.47E-01	1.70E-01		3.06E-17	2.93E-17	
	500	3.00E-13	796E-14		2.09E-09	4.61E-09		4.29E-13	3.31E-14		8.45E+01	6.40E+00		2.75E-15	2.13E-15	
	1000	<b>5.18E-13</b>	<b>9.61E-14</b>		8.78E-03	5.27E-03		8.45E-13	5.00E-14		5.53E+02	2.86E+01		1.06E-10	1.83E-11	
F2	100	6.08E+00	7.83E+00		8.25E+00	5.32E+00		2.52E+01	8.72E+00		1.86E+01	2.26E+00		2.71E+00	2.39E+00	
	500	5.79E+01	4.21E+01		7.57E+01	3.04E+00		6.66E+01	5.69E+00		4.35E+01	5.51E-01		3.12E+01	4.91E+00	
	1000	7.82E+01	4.25E+01		9.60E+01	1.81E+00		1.08E+02	4.75E+00		4.66E+01	4.00E-01		2.71E+01	5.66E+00	
F3	100	4.23E+02	8.65E+02		1.45E+02	5.84E+01		1.49E+02	5.72E+01		4.99E+03	5.35E+03		9.81E+01	2.25E-01	
	500	7.24E+02	1.54E+02		1.81E+03	2.74E+02		9.24E+02	1.72E+02		5.77E+04	8.04E+03		4.96E+02	6.23E-01	
	1000	1.33E+03	2.63E+02		9.14E+03	1.26E+03		1.79E+03	1.58E+02		8.37E+05	1.52E+05		9.97E+02	3.58E+01	
F4	100	3.98E-02	1.99E-01		4.37E+00	7.65E+00		4.38E-13	9.21E-14		4.71E+02	5.94E+01		0.00E+00	0.00E+00	
	500	3.98E-02	1.99E-01		3.64E+02	5.23E+01		1.79E-11	6.31E-11		3.49E+03	1.12E+02		0.00E+00	0.00E+00	
	1000	1.99E-01	4.06E-01		1.82E+03	1.37E+02		1.37E-10	3.37E-10		7.58E+03	1.51E+02		4.93E-11	2.87E-12	
F5	100	3.43E-03	4.88E-03		3.07E-14	7.86E-15		3.41E-14	1.16E-14		3.72E-01	5.60E-02		0.00E+00	0.00E+00	
	500	1.18E-03	4.61E-03		6.90E-04	2.41E-03		2.12E-13	2.47E-14		1.64E+00	4.69E-02		2.16E-16	7.56E-17	
	1000	1.18E-03	3.27E-03		3.58E-03	5.73E-03		<b>4.18E-13</b>	<b>2.78E-14</b>		5.89E+00	3.91E-01		5.49E-13	2.35E-14	
F6	100	1.44E-13	3.06E-14		1.13E-13	1.53E-14		1.11E-13	7.86E-15		2.06E+00	4.40E-01		3.21E-15	2.48E-15	
	500	5.34E-13	8.61E-14		4.80E-01	5.73E-01		<b>5.34E-13</b>	<b>7.01E-14</b>		6.64E+00	4.49E-01		6.74E-12	4.39E-12	
	1000	<b>1.02E-12</b>	<b>1.68E-13</b>		2.29E+00	2.98E-01		1.06E-12	7.68E-14		1.89E+01	2.49E+00		2.39E-07	2.13E-07	
F7	100	<b>-1.50E+03</b>	<b>1.04E+01</b>		-1.37E+03	2.46E+01		-1.54E+03	2.52E+00		-8.55E+02	1.35E+01		-7.00E+02	6.64E-01	
	500	-7.23E+03	4.16E+01		-5.74E+03	1.83E+02		-7.43E+03	8.03E+00		-3.51E+03	2.10E+01		-8.83E+03	3.54E+00	
	1000	-1.43E+04	8.27E+01		-1.05E+04	4.18E+02		-1.47E+04	1.51E+01		-6.62E+03	3.18E+01		-3.69E+04	2.16E+02	

CCPSO2 for function F7 in the case of 100-dimensional; the two algorithms obtain similar optimization results for the function F6 in the case of 500-dimensional. In addition, at first glance it might seem that HGGWA and MLCC have similar performance. However, the HGGWA achieves better convergence accuracy than MLCC under the same number of iterations. In particular, the HGGWA algorithm's optimization results are superior to DEwSAcc and EPUS-PSO in all dimensions. The result of HGGWA scaled very well from 100-D to 1000-D on F1, F4, F5, and F6, outperforming DEwSAcc and EPUS-PSO on all the cases.

**5.4. Sensitivity Analysis of Nonlinear Adjustment Coefficient  $k$ .** This section mainly discusses the effect of nonlinear adjustment coefficients on the convergence performance of the algorithm. In the HGGWA algorithm, the role of parameter  $a$  is to balance global exploration capabilities and local exploitation capabilities. In (12), the nonlinear adjustment coefficient  $k$  is a key parameter and is mainly used to control the range of variation of the convergence factor. Therefore, we selected four different values for numerical experiments to analyze the effect of nonlinear adjustment coefficients on the performance of the algorithm. The four different values are 0.5, 1, 1.5, 2, and the results of the comparison are shown in Table 6. Among them, black bold represents the best result of solving in several algorithms.

In general, the value of the nonlinear adjustment coefficient  $k$  has no significant effect on the performance of the HGGWA algorithm. On closer inspection, one can see that besides the function  $f_3$ , the optimization performance of the algorithm is optimal when the nonlinear adjustment coefficient  $k$  is 0.5. But for function  $f_3$ ,  $k = 1.5$  is the best choice. For functions  $f_7$  and  $f_9$ , the HGGWA algorithm obtains the optimal solution 0 in the case of several different values of the coefficient  $k$ . Therefore, for most test functions, the optimal value of the nonlinear adjustment factor  $k$  is 0.5.

**5.5. Global Convergence Analysis of HGGWA.** In the HGGWA algorithm, the following four improved strategies are used to ensure the global convergence of the algorithm: (1) initial population strategy based on Opposition-Based Learning; (2) selection operation based on optimal retention; (3) crossover operation based on population partitioning mechanism; and (4) mutation operation for elite individuals.

The idea of solving the high-dimensional function optimization problem by HGGWA algorithm is as follows. Firstly, for the initial population strategy, the basic GWO generates the initial population in a random manner. However, it will have a great impact on the search efficiency of the algorithm for high-dimensional function optimization problems. The algorithm cannot effectively search the entire problem solution space if the initial grey wolf population is clustered in a small range. Therefore, the Opposition-Based Learning strategy is used to initialize the population, which can make the initial population evenly distributed in the solution space, thus laying a good foundation for the global search of the algorithm. Secondly, the optimal retention strategy is used to preserve the optimal individual of the parent population

in the process of population evolution. As a result, the next generation of population can evolve in the optimal direction. In addition, the individuals with higher fitness are selected by the gambling roulette operation, which maintains the dominance relationship between individuals in the population. This dominance relationship makes the algorithm with good global convergence. Thirdly, the crossover operation is completed under the condition of population partitioning in order to achieve the purpose of dimension reduction and maintain the diversity of the population. Finally, the search direction of the grey wolf population is guided by the mutation operation of the elite individuals, which effectively prevents the algorithm from falling into the local optimal solution. All in all, through the improvement of these four different strategies, the HGGWA algorithm shows good global convergence in solving high-dimensional function optimization problems.

## 6. Conclusion

In order to overcome the shortcomings of GWO algorithm for solving large-scale global optimization problems which are easy to fall into local optimum, this paper proposes a Hybrid Genetic Grey Wolf Algorithm (HGGWA) by integrating three genetic operators into the algorithm. The improved algorithm initializes the population based on the Opposition-Based Learning strategy for improving the search efficiency of the algorithm. The strategy of population division based on cooperative coevolution reduces the scale of the problem, and the mutation operation of the elite individuals effectively prevents the algorithm from falling into the local optima. The performance of HGGWA has been evaluated using 10 classical benchmark functions and 7 CEC'2008 high-dimensional functions. From our experimental results, several conclusions can be drawn.

The results have shown that it is capable of grouping interacting variables with great accuracy for the majority of the benchmark functions. A comparative study among the HGGWA and other state-of-the-art algorithms was conducted and the experimental results showed that the HGGWA algorithm exhibits better global convergence whether it is solving separable functions or nonseparable functions.

(1) By testing 10 classical benchmark functions, compared with the results of WOA, SSA, and ALO, the HGGWA algorithm has been greatly improved in convergence accuracy. In addition to the Schwefel problem and Rosenbrock function, the HGGWA algorithm achieves an 80%-100% successful ratio for all benchmark functions with 100 dimensions, which proves the effectiveness of HGGWA for solving high-dimensional functions.

(2) By comparing with the optimization results on seven CEC'2008 large-scale global optimization problems, the results show that HGGWA algorithm achieves the global optimal value for the separable functions F1, F4, and F6, but the effect on other nonseparable problems is not very satisfactory. However, HGGWA algorithm still exhibits better global convergence than the other algorithms: CCPSO2, DEwSAcc, MLCC, and EPUS-PSO on most of the functions.

TABLE 6: Comparison of optimization performance of different  $k$  values for HGGWA.

Function	$k = 0.5$			$k = 1$			$k = 1.5$			$k = 2$		
	Mean	Std		Mean	Std		Mean	Std		Mean	Std	
$f_1$	<b>8.43E-60</b>	5.36E-60		2.25E-54	2.03E-54		5.58E-53	4.31E-53		2.31E-51	3.16E-51	
$f_2$	<b>2.89E-34</b>	2.31E-34		2.06E-33	1.89E-33		4.70E-32	3.65E-32		5.76E-30	4.22E-30	
$f_3$	1.11E+00	1.02E+00		3.14E+01	2.19E+01		<b>4.03E-01</b>	2.38E-01		3.60E+01	2.16E+01	
$f_4$	<b>9.78E+01</b>	8.36E+01		9.82E+01	7.43E+01		9.81E+01	7.34E+01		9.83E+01	4.61E+01	
$f_5$	<b>2.18E-03</b>	1.34E-03		8.87E-02	6.67E-02		7.81E+00	6.13E+00		1.74E-02	2.34E-02	
$f_6$	<b>1.31E-03</b>	1.13E-03		1.62E-03	2.43E-04		2.99E-03	1.04E-03		1.84E-03	1.34E-03	
$f_7$	<b>0.00E+00</b>	0.00E+00		0.00E+00	0.00E+00		0.00E+00	0.00E+00		0.00E+00	0.00E+00	
$f_8$	<b>1.58E-14</b>	1.23E-15		2.93E-14	1.37E-14		2.22E-14	1.49E-14		2.51E-14	1.08E-14	
$f_9$	<b>0.00E+00</b>	0.00E+00		0.00E+00	0.00E+00		0.00E+00	0.00E+00		0.00E+00	0.00E+00	
$f_{10}$	<b>2.99E-02</b>	2.49E-02		3.38E-02	1.24E-03		8.42E-02	6.14E-02		7.29E-02	8.13E-03	

(3) By analyzing the running time of several algorithms, the results show that the convergence time of HGGWA algorithm has obvious advantages compared with SSA, WOA, etc. algorithms which are proposed recently. For the functions  $f_1, f_2, f_6, f_7, f_8, f_9$  and  $f_{10}$ , the running time of the HGGWA algorithm is kept between 1s and 3s under the specified convergence accuracy, which shows the excellent stability of the HGGWA.

(4) In the future, we are planning to investigate more efficient population partition mechanism to adapt to different nonseparable problems in the process of crossover operation. We are also interested in applying HGGWA to real-world problems to ascertain its true potential as a valuable optimization technique for large-scale optimization such as the setup of large-scale multilayer sensor networks [73] in the Internet of Things.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Authors' Contributions

Qinghua Gu and Xuexian Li contributed equally to this work.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant Nos. 51774228 and 51404182), Natural Science Foundation of Shaanxi Province (2017JM5043), and Foundation of Shaanxi Educational Committee (17JK0425).

## References

- [1] R. Zhang and C. Wu, "A hybrid approach to large-scale job shop scheduling," *Applied Intelligence*, vol. 32, no. 1, pp. 47–59, 2010.
- [2] M. Gendreau and C. D. Tarantilis, Solving large-scale vehicle routing problems with time windows: The state-of-the-art[M]. Montreal: Cirrelet, 2010.
- [3] M. B. Liu, S. K. Tso, and Y. Cheng, "An extended nonlinear primal-dual interior-point algorithm for reactive-power optimization of large-scale power systems with discrete control variables," *IEEE Power Engineering Review*, vol. 22, no. 9, p. 56, 2002.
- [4] I. Fister, I. Fister Jr., J. Brest, and V. Žumer, "Memetic artificial bee colony algorithm for large-scale global optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '12)*, pp. 1–8, IEEE, Brisbane, Australia, June 2012.
- [5] A. Zamuda, J. Brest, B. Bošović, and V. Žumer, "Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, IEEE, June 2008.
- [6] T. Weise, R. Chiong, and K. Tang, "Evolutionary optimization: pitfalls and booby traps," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 907–936, 2012.
- [7] L. Lafleur, *Descartes: Discourse On Method*, Pearson Schweiz Ag, Zug, Switzerland, 1956.
- [8] Y.-S. Ong, M. H. Lim, and X. Chen, "Memetic Computation—Past, Present & Future [Research Frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.
- [9] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005.
- [10] S. Jiang, M. Lian, C. Lu, Q. Gu, S. Ruan, and X. Xie, "Ensemble prediction algorithm of anomaly monitoring based on big data analysis platform of open-pit mine slope," *Complexity*, vol. 2018, Article ID 1048756, 13 pages, 2018.
- [11] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [13] E. Daniel, J. Anitha, K. K. Kamaleshwaran, and I. Rani, "Optimum spectrum mask based medical image fusion using gray wolf optimization," *Biomedical Signal Processing and Control*, vol. 34, pp. 36–43, 2017.
- [14] A. A. M. El-Gaafary, Y. S. Mohamed, A. M. Hemeida, and A.-A. A. Mohamed, "Grey wolf optimization for multi input multi output system," *Universal Journal of Communications and Network*, vol. 3, no. 1, pp. 1–6, 2015.
- [15] G. M. Komaki and V. Kayvanfar, "Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *Journal of Computational Science*, vol. 8, pp. 109–120, 2015.
- [16] S. Jiang, M. Lian, C. Lu et al., "SVM-DS fusion based soft fault detection and diagnosis in solar water heaters," *Energy Exploration & Exploitation*, 2018.
- [17] Q. Gu, X. Li, C. Lu et al., "Hybrid genetic grey wolf algorithm for high dimensional complex function optimization," *Control and Decision*, pp. 1–8, 2019.
- [18] M. A. Potter and K. A. D. Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel Problem Solving from Nature—PPSN III*, vol. 866 of *Lecture Notes in Computer Science*, pp. 249–257, Springer, Berlin, Germany, 1994.
- [19] M. A. Potter, *The Design and Analysis of a Computational Model of Cooperative Coevolution*, George Mason University, Fairfax, Va, USA, 1997.
- [20] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [21] M. El-Abd, "A cooperative approach to the artificial bee colony algorithm," in *Proceedings of the 2010 6th IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, IEEE, Spain, July 2010.
- [22] Y. Shi, H. Teng, and Z. Li, "Cooperative co-evolutionary differential evolution for function optimization," in *Proceedings of the 1st International Conference on Natural Computation (ICNC '05)*, *Lecture Notes in Computer Science*, pp. 1080–1088, August 2005.
- [23] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.



- [24] X. Li and X. Yao, "Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms," in *Proceedings of the Congress on Evolutionary Computation, CEC '9*, pp. 1546–1553, IEEE Computational Intelligence Magazine, Trondheim, Norway, May 2009.
- [25] M. N. Omidvar, X. Li, Z. Yang, and X. Yao, "Cooperative co-evolution for large scale optimization through more frequent random grouping," in *Proceedings of the 2010 6th IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, Spain, July 2010.
- [26] T. Ray and X. Yao, "A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC)*, pp. 983–989, IEEE, Trondheim, Norway, May 2009.
- [27] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6239, pp. 300–309, 2010.
- [28] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.
- [29] M. N. Omidvar, X. Li, and X. Yao, "Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms," in *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference, GECCO'11*, pp. 1115–1122, IEEE, Ireland, July 2011.
- [30] R. Storn, "On the usage of differential evolution for function optimization," in *Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS '96)*, pp. 519–523, June 1996.
- [31] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.
- [32] K. Weicker and N. Weicker, "On the improvement of coevolutionary optimizers by learning variable interdependencies," in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, pp. 1627–1632, IEEE, July 1999.
- [33] E. Sayed, D. Essam, and R. Sarker, "Using hybrid dependency identification with a memetic algorithm for large scale optimization problems," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7673, pp. 168–177, 2012.
- [34] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 1101–1108, IEEE, Seoul, South Korea, 2001.
- [35] E. Sayed, D. Essam, and R. Sarker, "Dependency identification technique for large scale optimization problems," in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation, CEC 2012*, Australia, June 2012.
- [36] Y. Ren and Y. Wu, "An efficient algorithm for high-dimensional function optimization," *Soft Computing*, vol. 17, no. 6, pp. 995–1004, 2013.
- [37] J. Liu and K. Tang, "Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution," in *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning – IDEAL 2013*, vol. 8206 of *Lecture Notes in Computer Science*, pp. 350–357, Springer, Berlin, Germany, 2013.
- [38] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 1663–1670, June 2008.
- [39] M. N. Omidvar, Y. Mei, and X. Li, "Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 1305–1312, China, July 2014.
- [40] S. Mahdavi, E. M. Shiri, and S. Rahnamayan, "Cooperative Co-evolution with a new decomposition method for large-scale optimization," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014.
- [41] B. Kazimpour, M. N. Omidvar, X. Li, and A. K. Qin, "A sensitivity analysis of contribution-based cooperative co-evolutionary algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2015*, pp. 417–424, Japan, May 2015.
- [42] Z. Cao, L. Wang, Y. Shi et al., "An effective cooperative coevolution framework integrating global and local search for large scale optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2015*, pp. 1986–1993, Japan, May 2015.
- [43] X. Peng, K. Liu, and Y. Jin, "A dynamic optimization approach to the design of cooperative co-evolutionary algorithms," *Knowledge-Based Systems*, vol. 109, pp. 174–186, 2016.
- [44] H. K. Singh and T. Ray, "Divide and conquer in coevolution: a difficult balancing act," in *Agent-Based Evolutionary Search*, vol. 5 of *Adaptation, Learning, and Optimization*, pp. 117–138, Springer, Berlin, Germany, 2010.
- [45] X. Peng and Y. Wu, "Large-scale cooperative co-evolution using niching-based multi-modal optimization and adaptive fast clustering," *Swarm & Evolutionary Computation*, vol. 35, 2017.
- [46] M. Yang, M. N. Omidvar, C. Li et al., "Efficient resource allocation in cooperative co-evolution for large-scale global optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 493–505, 2017.
- [47] M. N. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC 2010)*, pp. 1–8, Barcelona, Spain, July 2010.
- [48] X. Peng and Y. Wu, "Enhancing cooperative coevolution with selective multiple populations for large-scale global optimization," *Complexity*, vol. 2018, Article ID 9267054, 15 pages, 2018.
- [49] S.-T. Hsieh, T.-Y. Sun, C.-C. Liu, and S.-J. Tsai, "Solving large scale global optimization using improved particle swarm optimizer," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pp. 1777–1784, China, June 2008.
- [50] T. Hatanaka, T. Korenaga, N. Kondo et al., *Search Performance Improvement for PSO in High Dimensional Space*, InTech, 2009.
- [51] J. Fan, J. Wang, and M. Han, "Cooperative coevolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 829–839, 2014.
- [52] A.-R. Hedar and A. F. Ali, "Genetic algorithm with population partitioning and space reduction for high dimensional problems," in *Proceedings of the 2009 International Conference on Computer Engineering and Systems, ICCES'09*, pp. 151–156, Egypt, December 2009.

- [53] J. Q. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [54] L. M. Hvattum and F. Glover, "Finding local optima of high-dimensional functions using direct search methods," *European Journal of Operational Research*, vol. 195, no. 1, pp. 31–45, 2009.
- [55] C. Liu and B. Li, "Memetic algorithm with adaptive local search depth for large scale global optimization," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 82–88, China, July 2014.
- [56] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1257–1263, 2015.
- [57] A. Zhu, C. Xu, Z. Li, J. Wu, and Z. Liu, "Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC," *Journal of Systems Engineering and Electronics*, vol. 26, no. 2, pp. 317–328, 2015.
- [58] V. Chahar and D. Kumar, "An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems," *Advances in Engineering Software*, vol. 112, pp. 231–254, 2017.
- [59] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of large scale power system using quasi-oppositional grey wolf optimization algorithm," *Engineering Science and Technology, an International Journal*, vol. 19, no. 4, pp. 1693–1713, 2016.
- [60] C. Lu, S. Xiao, X. Li, and L. Gao, "An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production," *Advances in Engineering Software*, vol. 99, pp. 161–176, 2016.
- [61] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.
- [62] W. Long, J. Jiao, X. Liang, and M. Tang, "Inspired grey wolf optimizer for solving large-scale function optimization problems," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 60, pp. 112–126, 2018.
- [63] S. Gupta and K. Deep, "Performance of grey wolf optimizer on large scale problems," in *Proceedings of the Mathematical Sciences and Its Applications*, American Institute of Physics Conference Series, Uttar Pradesh, India, 2017.
- [64] R. L. Haupt and S. E. Haupt, "Practical genetic algorithms," *Journal of the American Statistical Association*, vol. 100, no. 100, p. 253, 2005.
- [65] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC '05)*, pp. 695–701, Vienna, Austria, November 2005.
- [66] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.
- [67] H. Wang, S. Rahnamayan, and Z. Wu, "Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems," *Journal of Parallel and Distributed Computing*, vol. 73, no. 1, pp. 62–73, 2013.
- [68] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [69] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2006.
- [70] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [71] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [72] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [73] Q. Gu, S. Jiang, M. Lian, and C. Lu, "Health and safety situation awareness model and emergency management based on multi-sensor signal fusion," *IEEE Access*, vol. 7, pp. 958–968, 2019.

## Review Article

# Review of the Complexity of Managing Big Data of the Internet of Things

David Gil <sup>1</sup>, Magnus Johnsson <sup>2,3,4</sup>, Higinio Mora <sup>1</sup>, and Julian Szymański <sup>5</sup>

<sup>1</sup>University of Alicante, Alicante, Spain

<sup>2</sup>Malmö University, Malmö, Sweden

<sup>3</sup>Department of Intelligent Cybernetic Systems, NRNU MEPhI, Moscow, Russia

<sup>4</sup>AI Research AB, Höör, Sweden

<sup>5</sup>Gdansk University of Technology, Gdansk, Poland

Correspondence should be addressed to David Gil; [david.gil@ua.es](mailto:david.gil@ua.es)

Received 9 November 2018; Revised 4 January 2019; Accepted 15 January 2019; Published 3 February 2019

Academic Editor: Danilo Comminiello

Copyright © 2019 David Gil et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There is a growing awareness that the complexity of managing Big Data is one of the main challenges in the developing field of the Internet of Things (IoT). Complexity arises from several aspects of the Big Data life cycle, such as gathering data, storing them onto cloud servers, cleaning and integrating the data, a process involving the last advances in ontologies, such as Extensible Markup Language (XML) and Resource Description Framework (RDF), and the application of machine learning methods to carry out classifications, predictions, and visualizations. In this review, the state of the art of all the aforementioned aspects of Big Data in the context of the Internet of Things is exposed. The most novel technologies in machine learning, deep learning, and data mining on Big Data are discussed as well. Finally, we also point the reader to the state-of-the-art literature for further in-depth studies, and we present the major trends for the future.

## 1. Introduction

The fast-developing and expanding area known as the Internet of Things (IoT) [1–3] involves expanding the Internet beyond such standard devices as computers, smartphones, and tablets to also include the connection of other physical devices and objects. This allows for a variety of devices, sensors, etc. to be monitored and controlled, and to interact and communicate via the Internet. This means that an abundance of opportunity for brand new and revolutionary types of services and applications arises. As a result, we are now witnessing a technological revolution where millions of people are connecting and generate tremendous amounts of data through the increasing use of a wide variety of devices. These include smart devices and any type of wearable that are connected to the Internet, powering novel connected applications and solutions. The cost of technology has sharply decreased making it possible for everybody to access the Internet and to gather data and an abundance of real-time information.

One immediate consequence of this revolutionary emergence of novel technological opportunities is the urgent need for the development and adaptation of other related areas to further enable the development of the IoT field. Thus, new words, as well as new expressions, have started to emerge, such as Big Data [4, 5], cloud computing [6], and Data science. Data science has been defined as a “concept to unify statistics, data analysis, machine learning and their related methods” to “understand and analyse actual phenomena” with data [7, 8], and there is now a strong demand for professional data scientists in a multitude of sectors [9–12].

This article aims at providing a review of IoT related surveys in order to highlight the opportunities and the challenges, as well as the state-of-the-art technologies related to Big Data. There will be a particular focus on how to address the arising problems of managing the ensuing increased complexity. Since it is such a complex area, we have divided the Big Data procedure into several different stages to establish the most important points in each, while highlighting to



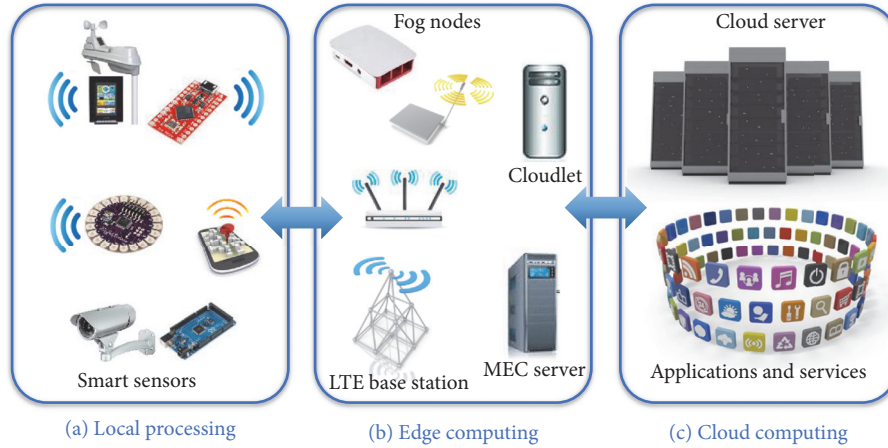


FIGURE 1: A schematic depiction of three different approaches to handle the complexity of the intensive data processing arising as a consequence of the tremendous amounts of collected IoT data.

the reader the most relevant papers related to every stage. Due to the complexity of managing Big Data, we have created separate sections in regard to the aforementioned stages of Big Data procedure. Our contribution explicitly indicates the advantages of every stage in the knowledge discovering procedure in contrast to approaches that offer more general visions. The advantage of this proposal is to be able to understand as well as analyse the challenges and opportunities in every particular phase.

The remainder of this article is structured as follows: first the next section discusses a set of general approaches to handle the complexity of managing Big Data in the context of the IoT as well as the future trends in the development of these approaches; then a section follows that discusses the knowledge discovering procedure in data gathered from a large number of diverse devices in the context of the IoT; finally, we provide a conclusion that summarises the article and points out major future trends.

## 2. The Internet of Things and Complexity Handling: Architectures for Big Data

The Internet of Things (IoT) paradigm has brought a great revolution to our society [13–15]. It is a technology that makes our world better. It allows us to get information about the physical environment around us, and from this data valuable knowledge can be inferred about how the world works. This knowledge enables the deployment of new real-world applications, and it makes it easier for smart decisions to improve the quality of life of the citizens of our society. There are many examples of how this novel technology runs. The smart city concept is a representative use case, where many applications have been developed for its ecosystem [16–19].

An important source of complexity within the IoT paradigm comes from the great amount of data collected. In most cases, the data also need to be processed in order to be converted into useful knowledge.

In view of the recent proposals on how to handle the complexity of Big Data, there are three general approaches

to carry out the ensuing very intensive data processing: (A) local processing; (B) edge computing; and (C) cloud computing. Figure 1 shows a schematic overview of these approaches, and Table 1 summarises a representative set of ways and aspects of handling the complexity arising from the IoT. Table 1 also provides references to corresponding papers, categorised under the headings of the three general approaches mentioned above. In the following subsections, brief descriptions of each of these approaches are presented, and finally their main future trends are introduced.

**2.1. Local Processing.** This approach basically consists of processing the data where the data is collected. In this way, no raw data need to be communicated to remote servers. Instead, only the useful and relevant information is centralised to make smart decisions [20, 21]. In addition, deploying the first-level intelligence closer to the sensors produces an increase in the overall energy efficiency and significantly reduces the communication needs of many IoT applications.

This approach develops the concept of ‘smart sensor,’ which was initially defined as ‘smart transducer’ [22]. A smart sensor is a sensor with computing and communication capabilities to make computations with the acquired data, make decisions, and store information for further use and perform two-way communications [23]. Smart sensors are becoming integral parts of intelligent systems and they are indispensable enablers of the IoT paradigm and the corresponding development of advanced applications. A typical example of these developed sensors is the ‘smart wearable.’ This device can acquire several biosignals, process them, show elaborated information to the user, and send the relevant information to, for example, external platforms for medical supervision [23–25]. Other important applications come from the logistics [26] and industrial fields [27]. Indeed, the new computation and communication capabilities of the IoT paradigm allow for the implementation of intelligent manufacturing systems giving rise to the next generation of industry, the so-called ‘Industry 4.0’ [28].

TABLE 1: A representative set of ways and aspects of handling the complexity arising from the IoT, together with references to corresponding papers, categorized under the headings of three general approaches (local processing, edge computing and cloud computing) to carry out intensive data processing of Big Data.

Work	Main contributions
<b>(A) Local processing</b>	
Smart sensing for IoT applications [21]	Discusses emerging trends of smart sensing.
Sensor Fusion and Smart Sensor in Sports and Biomedical Applications [24]	An overview of smart sensors and sensor fusion.
High-level modelling and synthesis of smart sensor networks for Industrial Internet of Things [27]	Efficient design process and methodology for complex industrial applications.
Smart Sensing Devices for Logistics Application [26]	Analysis of the logistics sector and Cyber Physical Systems (CPS) as smart connected solutions.
Intelligent Manufacturing in the Context of Industry 4.0: A Review [28]	Review of key technologies such as the IoT and cyber-physical systems.
<b>(B) Edge computing</b>	
Edge Computing: Vision and Challenges [34]	Challenges and opportunities in the field of edge computing are described.
Collaborative Working Architecture for IoT-Based Applications [20]	Network design, which combines sensing and processing capabilities based on the MCC paradigm.
IoT-Based Computational Framework [25]	Distributed framework based on the IoT paradigm for real-time monitoring.
Edge Computing [35]	Analysis of the edge computing paradigm.
Fog Computing [48]	The Fog Computing framework.
Secure Multi-Tier Mobile Edge Computing Model [49]	Formal framework to handle the security level of edge computing environments.
Mobile Edge Computing [38]	Analysis of opportunities, solutions, and challenges of the MEC paradigm.
Cloudlets [39]	Introduction to the cloudlet concept for offloading computations.
Future Edge Cloud and Edge Computing for Internet of Things Applications [40]	Discussion of Edge Cloud and Edge Computing research efforts.
<b>(C) Cloud computing</b>	
A Smart Sensing Architecture for Domestic Monitoring [50]	Integrated sensor network deployment with advanced Cloud Computing Data Mining algorithms.
IoT-as-a-Service [43]	Strategy for evaluating the information quality in delivering IoT-as-a-Service.
The shift to Cloud Computing [41]	This paper analyses the impact of the shift to the Cloud-based model.
Accessibility analysis in smart cities [46]	Comprehensive system for monitoring urban accessibility in smart cities.
Big data analytics framework for smart cities [47]	Smart City Data Analytics Panel for Big Data analytics.
Sensing and Actuation as a Service Delivery Model [44]	A novel system model for Sensing and Actuation as a Service (SAaaS).
User Quality-Of-Experience and Service Provider Profit in 5G [51]	The Quality-of-Experience (QoE) and the Profit-aware Resource Allocation problems are analysed.
Orchestrated Platform for Cyber-Physical Systems [45]	Discussion on the scalability of the sensor data back-end and the predictive simulation architecture for CPS.

In these environments, network virtualization plays a significant role in providing flexibility and better manageability to Internet [29]. This is a way for reducing the complexity of the infrastructure since network resources can be managed as logical services, rather than physical resources. This feature enables us to implement smart scheduling methods for network usage and dataflows routing from IoT applications [30].

In order to properly carry out this resource management, network performance monitoring needs to be performed

in effective and efficient ways. However, it remains a challenge for network operators [31] since active monitoring techniques used to dynamically acquire it can introduce overheads in the network [32]. In general, existing methods are hard to use in practice and further research is needed in this area. Nevertheless, a promising idea to address this challenge consists in reducing the data measurement by implementing intelligent measurement schemes based on inference techniques from partial direct monitoring data [33].

**2.2. Edge Computing.** Edge computing is a novel paradigm which has spawned great interest recently. It consists of the deployment of storage and computing capabilities at the ‘Edge’ of the Internet. The ‘Edge’ of the Internet can be defined as the portion of the network between sensors or data sources and cloud data centres [34]. The edge computing paradigm aims at deploying computing, storage, and network resources in this portion. The physical proximity of the computing platforms to where the data acquisition happens makes it easier to achieve lower end-to-end latency, high bandwidth, and low jitter to services [35].

There are several ways to implement edge computing that have in turn led to different approaches, such as Fog Computing, Mobile Edge Computing (MEC), and Cloudlet Deployment. Fog Computing consists in using the network devices such as routers, switches, and gateways as Fog Nodes to provide storage and computing resources [36]. In addition, network virtualization has significantly contributed to developing this paradigm by considering the fog devices as virtual network nodes. This trend increases the deployment flexibility of Fog Computing services and their integration with mobile devices and ‘things’ [37]. MEC is a novel paradigm based on deploying cloud computing capabilities in the base stations of the telecom operators [38]. Finally, Cloudlet Deployment consists in the same concept as Cloud Computing, but without the Wide Area Network (WAN) inconveniences. The servers are installed within the local networks where the data sources are connected. These servers are known as cloudlets [39].

Applications for edge computing, such as in Virtual Reality and Gaming Applications [40], cannot tolerate high latency, or its unpredictability. This is something that remote cloud servers cannot deliver.

**2.3. Cloud Computing.** The Cloud Computing paradigm is one of the most disruptive technological innovations in the last few years. It makes available to anyone a flexible amount of computing resources under per-use payment methods, the so-called ‘as-a-service’ model. Currently, more and more software and hardware solutions are redesigned for this cloud paradigm [41].

The cloud computing model favours the development of large-sized data centres where the resources are optimised through virtualization and efficient management systems. This technology gives the IoT applications the possibility to work in different environments in a very agile way using the same infrastructure [42]. In such a way, combining the cloud computing paradigm with IoT forms a new type of distributed system able to provide IoT-as-a-Service (IoTaaS) [43]. This concept allows for the integration of powerful computing resources with different types of devices such as sensors, actuators, and other embedded devices to deliver advanced services and applications based on the gathered data. A particular instance of this idea is the Sensing and Actuation Cloud where the connected IoT devices are mainly sensors and actuators [44], or the Cloud Cyber Physical Systems (CPS) composed of sensors or sensor networks [45].

There are a great variety of successful examples of this trend in many areas, where the data are analysed in the cloud

through Big Data and data mining methods to infer valuable knowledge from them and deliver rich and smart services to the stakeholders. For example, the smart city concept, mentioned above, is in part made possible by a centralised cloud-based data analysis and service provision [41, 46, 47].

In addition, a combination of these options can be designed taking several aspects into account, such as power consumption, communication networks, and the availability of computing platforms. Dynamic solutions can easily adapt to the more favourable approach to better handle the complexity and meet the operation constraints.

**2.4. Future Trends.** Regarding the future trends of the developments of these three general approaches to intensive data processing of IoT related Big Data, there are developments at several fronts. The following is a summary of those most relevant.

When it comes to local processing, the efforts are directed towards the continuous improvement of smart sensor devices. We can distinguish several research lines here. One is the efforts to increase the performance of the devices while simultaneously reducing their power consumption. Another is the integration of multiple sensing modalities on the same chip. Still another is the efforts directed towards the improvement of the methods employed for the extraction of useful information from the raw data [21].

Edge computing has a promising future since it decentralises the computing power along the network and produces clear benefits when it comes to response time and reliability [34]. The research lines in this field aim at reaching a smooth engagement with the IoT ecosystem, mainly by reducing the management complexity of dispersed edge resources and developing mechanisms to maintain the security perimeter for the data and applications [49].

The cloud computing paradigm has triggered a strong growth of computing services around the world. For this reason, there is intensive ongoing research on expanding cloud services and solutions to new fields of application. These tasks seek to simplify business and make services easier for stakeholders. In this way, the new 5G protocol will facilitate access for services and applications in the cloud improving the Quality-of-Experience [51].

### 3. Knowledge Discovering Procedure

In Figure 2, a classical procedure of discovering knowledge from the data gathered from a large number of diverse devices is depicted. In this figure, we get an overview of all the stages involved in such a process. There are many challenges involved in these stages that will be described next.

**3.1. IoT Data Gathering.** The gathering of data for IoT architectures involves collection from different sources like social networks, the web, various devices, software applications, humans, and not the least various kinds of sensors. In addition to physical sensors, there are also virtual sensors that are created by the combination and fusion of data from different physical sensors in the cloud [52]. When it

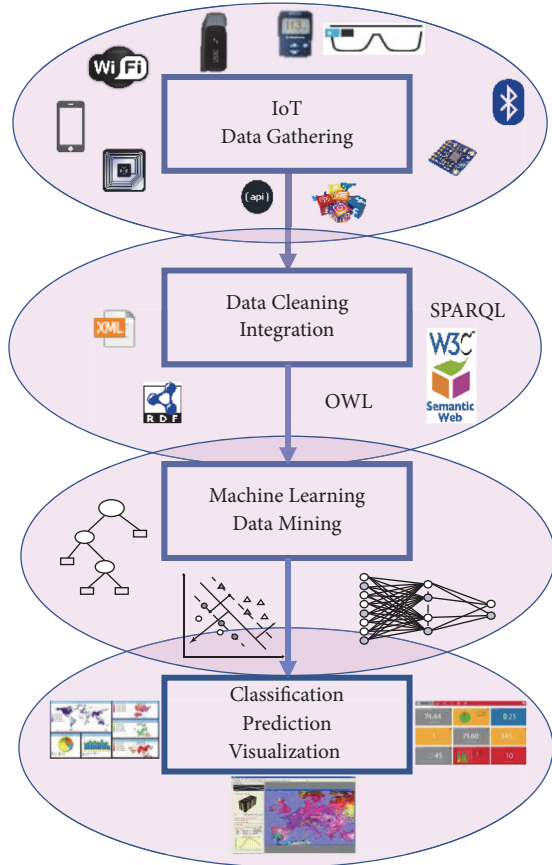


FIGURE 2: A classical procedure for the discovery of knowledge based on data gathered from a large number of diverse devices.

comes to the gathering of data from sensors, not only the raw sensor data are collected and stored, but these are also often linked to, for example, relevant contextual information, which increases the value of the data [53]. All these different sources engender large amounts of various types of data that, of course, also increases the requirements for storage capacity. The increasingly affordable storage resources that have recently become available mitigates this problem to some extent though.

Sensor networks are central for realising the IoT and in order to handle large amounts of polymorphous, heterogeneous sensor data on a large scale. Very Large-Scale Sensor Networks are employed using Cloud Computing [54]. Some of the main challenges regarding Very Large-Scale Sensor Networks are to handle the sensor resources and the computational resources and to store and process the sensor data.

Table 2 provides references to papers focused on the gathering of data in the context of the IoT.

**3.2. Data Cleaning and Integration.** A consequence of the way information is gathered through various sources and devices within IoT is that the information varies broadly in structure and type. This leads to a need for integration,

which can be defined as a set of techniques used to combine data from disparate sources into meaningful and valuable information.

Integration is one of the most challenging issues of Big Data, which is also associated with one of the most difficult Vs of Big Data, i.e., the variety of data. Table 3 shows a summary of papers that are focused on the problem of variety of information in Big Data.

Moreover, given the current context in which companies are organized, it is not enough to work with internal, local, and private databases. In most cases, there is also a need for the World Wide Web where many diverse databases and other data sources must interact and interoperate. This circumstance leads us to concepts such as heterogeneity and uncertainty.

Table 4 summarizes papers that deal with integration by means of a diversity of techniques and methods like XML, ontological constructs from knowledge representation, uncertainty, and data provenance.

**3.3. Data Mining and Machine Learning.** As more devices, sensors, etc. generate large amounts of data within the IoT, the question arises whether there are possibilities of finding hidden information in that data.

Data mining is a process that detects interesting knowledge from information repositories. This process is partly based on methods derived from modern machine learning algorithms adapted to fit Big Data and that extracts hidden information from, e.g., databases, data warehouses, data streams, time series, sequences, text, the web, and the large amount or valuable data generated by the IoT. Data mining aims at creating efficient predictive and descriptive models of large amounts of data that also generalize to new data [78]. It includes methods such as clustering, classification, time series analysis, association rule mining, and outlier analysis [79]. The precise choice among diverse data mining and machine learning techniques often depends on the taxonomy of the dataset.

Clustering includes unsupervised learning and uses the available structure to group data based on various kinds of similarity measures. Some examples of clustering methods are hierarchical clustering and partitioning algorithms, e.g., K-Means.

Classification is the process of finding models/functions describing classes that allow the prediction of class membership for new data. Some examples of classification methods are the K-Nearest Neighbour algorithm, Artificial Neural Networks, Decision Trees, Support Vector Machines, Bayesian Methods, and Rule-Based Methods.

In time series analysis meaningful properties are extracted from data over time, and in association rule mining, association rules are detected based on attribute-value conditions that are found frequently in the dataset.

Outlier analysis detects patterns that differ significantly from the main part of the data. The methods used are based on properties such as the density distribution or the distances between the instances in the data.



TABLE 2: The table summarizes and refers to a representative set of papers focusing on the gathering of data in the context of the IoT.

Reference	Title of paper	Description/Objective
[2]	Sensing as a Service and Big Data	Examines new approaches of IoT architectures, big sensor network applications, sensor data, and context-aware capturing techniques.
[55]	Internet of Things (IoT): A vision, architectural elements, and future directions	Describes an approach based on the cloud for worldwide implementation of Internet of Things.
[56]	Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges	Emphasis on the opportunities and challenges for the IoT and its future perspective in the health care area.
[15]	Internet of Things: A review of Surveys based on Context Aware Intelligent Services	A review of IoT studies that offer integrated and context-aware intelligent services.
[57]	Compressed sensing signal and data acquisition in wireless sensor networks and Internet of Things	Discusses how new insights can be supplied by compressed sensing into data sampling and acquisition for IoT.
[58]	A Computational Architecture Based on RFID Sensors for Traceability in Smart Cities	A novel approach of a distributed system to represent as well as providing the pathway and movement of people in densely geographical areas by means of a smart sensor network based on RFID.
[59]	Introducing a Novel Hybrid Artificial Intelligence Algorithm to Optimize Network of Industrial Applications in Modern Manufacturing	General modelling to evaluate and optimize nonlinear RFID network planning problems utilizing artificial intelligence techniques.

TABLE 3: The table summarizes and refers to a representative set of papers focusing on the variety of information in the context of the IoT.

Reference	Title of paper	Description/Objective
[60]	Data-intensive applications, challenges, techniques and technologies: A survey on Big Data	Survey on Big Data: applications, opportunities challenges, techniques and technologies.
[61]	The rise of “big data” on cloud computing: Review and open research issues	Important concepts of Big Data are introduced in this study and relationships among those concepts are provided. Finally, it summarizes the open research areas and how they need to be addressed.
[62]	Deep learning applications and challenges in big data analytics	In this paper, it is indicated how beneficial Deep Learning could be for several aspects of Big Data pattern recognition, analytics, semantic, etc.
[63]	On the use of MapReduce for imbalanced big data using Random Forest	In this experimental study, the performances with Random Forest classifier and MapReduce scheme have been used in order to deal with Imbalanced dataset.

Table 5 provides a summary of, and references to, papers focusing on machine learning and data mining in the context of Big Data.

**3.4. Deep Learning.** In recent years, deep learning has become an important technology for solving a wide range of machine learning tasks [85]. There are applications for natural language processing [86], signal processing [87], and video analysis that allows for the achievement of significantly better results than the state-of-the-art baselines. Also, deep learning is a very useful tool for processing large volumes of data [62]. Because of high efficiency of processing data obtained from complex sensing environments at different spatial and temporal resolutions, deep learning is a suitable tool for analysing real-world IoT data. According to Gartner's Top 10 Strategic Technology Trends for 2017 ([https://www.gartner.com/smarterwithgartner/gartners-top-](https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/)

[10-technology-trends-2017/](https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/)), deep learning and IoT will become one of the most strategic technological two-way relationships: from the IoT side there are large volumes of data produced that require advanced analytics offered by the deep learning side. A wide range of deep learning architectures [88] finds applications for processing the data from IoT environments: convolutional networks for image analysis, recurrent networks for signal processing, autoencoders for denoising, feed forward networks for classification, and regression. Figure 3 represents a general architecture of deep learning.

Usually, the data are processed in dedicated frameworks such as Tensorflow (<https://www.tensorflow.org/>), Theano (<http://deeplearning.net/software/theano/>), Caffe (<http://caffe.berkeleyvision.org/>), H2O (<https://www.h2o.ai/>), and Torch (<http://torch.ch/>). Often GPUs or clusters of GPU servers are used for the processing [78, 79].

TABLE 4: The table summarizes and refers to a representative set of papers focusing on Data Integration in the context of the IoT.

Reference	Title of paper	Description/Objective
[64]	Principles of Data Integration	This paper brings up the notion that new services have to be able to share data among several applications and organizations, as well as integrating the data efficiently and flexibly.
[65]	Answering queries using views: A survey	This paper presents a survey of important methods that are employed to answer queries using views.
[66]	MiniCon: A scalable algorithm for answering queries using views	In this paper a survey of methods for efficient and comprehensive answering of queries using views is presented.
[67]	XQuery: the XML query language	This paper introduces the XML query language XQuery.
[68]	From semistructured data to XML: Migrating the Lore data model and query language	This paper discusses the adaptation to XML of databases and semistructured languages.
[69]	Querying XML streams	In this paper a construct called TurboXPath, similar to x-scan, is used for processing hierarchical “native XML” data pages written to disk.
[70]	Semantic integration: a survey of ontology-based approaches	This paper provides a survey of ontology-based approaches to semantic integration.
[71]	Learning to map between ontologies on the semantic web	This paper presents assisting tools for the mapping between ontologies on the semantic web.
[72]	Containment of conjunctive queries on annotated relations	This paper indicates the relationships between different provenance formalisms.
[73]	Perm: Processing provenance and data on the same data model through query rewriting	This paper presents a provenance model similar to that of semi-rings focusing on supporting other operators such as semi-joins.
[74]	Google fusion tables: web-centered data management and collaboration	A presentation of a cloud-based system that facilitates the integration of data on the web. Datasets, e.g. in the form of CSV files or spreadsheets, can be uploaded to the system and made public or shared with collaborators.
[75]	Global detection of complex copying relationships between sources	Methods that are developed to detect copying relationships between sources in order to find the number of independent occurrences of facts are discussed in this paper.
[76]	Crowdsourcing systems on the world-wide web	A survey to get a global picture of crowdsourcing systems on the Web is presented in this paper.
[77]	A Novel Multidimensional Approach to Integrate Big Data in Business Intelligence	In this paper, an approach for integrating different formats into the recent RDF Data Cube format is presented. The approach is based on a MapReduce paradigm.

TABLE 5: The table summarizes and refers to a representative set of papers focusing on Machine Learning and Data Mining in the context of Big Data.

Reference	Title of paper	Description/Objective
[4]	Big data: A survey	This paper discusses technical challenges and reviews advances of the value chain (data generation, data acquisition, data storage, and data analysis) of Big Data
[80]	Data mining with big data	A HACE theorem that characterizes features of the Big Data revolution is presented, and a Big Data processing model, from a Data Mining perspective, is proposed in this paper.
[81]	Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners	A look into how to leverage Big Data analytics efficiently in order to foster positive change.
[82]	Mining Big Data: current status, and forecast to the future	This paper forecasts the future and presents the current status and controversies when it comes to some of the most interesting state-of-the-art topics in Big Data Mining.
[83]	Complex Power System Status Monitoring and Evaluation Using Big Data Platform and Machine Learning Algorithms: A Review and a Case Study	This paper reviews intelligent machine learning methods for complex power systems and key technologies in Big Data management.
[84]	Mining Outlier Data in Mobile Internet-Based Large Real-Time Databases	In this paper a novel mining outlier data method for analysing real-time data features is presented.

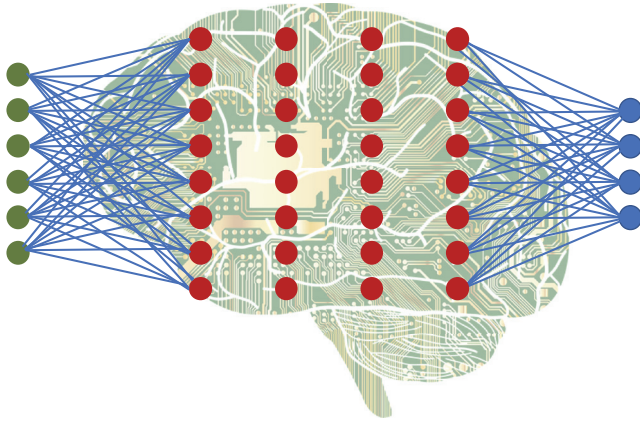


FIGURE 3: As can be seen in the figure, a Deep Learning Neural Network contains several hidden layers. Often the heavy computations are run on GPUs or clusters of GPU servers.

They offer different execution models as standalones or utilize high-performance computing based on, e.g., Hadoop, or Spark Cluster that allows a reduced time of computations. The frameworks have been widely compared and the reviews can be found online (<https://dzone.com/articles/8-best-deep-learning-frameworks>) (<https://www.exastax.com/deep-learning/a-comparison-of-deep-learning-frameworks/>). It should be noticed that these frameworks implement a processing model where the data are transferred to a server performing the analysis and in a final stage the response is returned. This model is subject to latency that could not be acceptable in some applications where there are requirements for high reliability, like, for example, when it comes to autonomous cars [89]. Thus, if efficiency constraints require real-time data processing, then a particular implementation of the algorithm is made on a local node. In its basic setting, this solution does not allow the use of information from other sources. An example of on the node-processing has been presented in [90], where on the node spectral domain preprocessing is used before the data is passed onto the deep learning framework for Human Activity Recognition.

For the IoT the deep analytics are made on large data collections and are usually based on creating more descriptive features of processed objects. For example, in temporal data processing for indoor location prediction [91], a Semisupervised Deep Extreme Learning Machine algorithm has been proposed that improves the localisation performance. The wireless positioning method has been improved with the usage of the Stacked Denoising Autoencoder and that also improves the performance by creating reliable features from a large set of noisy samples [92]. The prediction of home electricity power consumption has been analysed with a deep learning system that automatically extracts features from the captured data and optimises the electricity supply of the smart grid [93].

In Edge Computing with the analytics performed by a deep learning cluster [94], the resource consumption has been efficiently reduced [95]. Convolutional neural networks with automatically created features appeared to be a very

good solution for privacy preservation [96]. Also in the security domain, deep learning finds many applications, e.g., it allows the construction of a model-based attack detection architecture for the IoT for cyber-attack detection in fog-to-things computing [97].

Video analysis integrated in IoT networks is strongly supported by neural networks, e.g., deep learning-based visual food recognition allows for the construction of a system employing an edge computing-based service for accurate dietary assessment [98]. RTFace, a mechanism for denaturing video streams, has been based on a Deep Neural Network for face detection [99]. It selectively blurs faces and enables privacy management for live video analytics.

**3.5. Classification, Prediction, and Visualization.** This section discusses the final stage in the chain of the “Procedure for Knowledge Discovery,” which is the obtainment of the final knowledge extracted from the raw data.

When employing machine learning methods for classification and prediction, it is important to use methods with good ability to generalize. The reason for this is that when we apply any of the aforementioned techniques, and after they have been trained on the original data, we want them to make good classifications and predictions of novel data rather than on the data used for training.

After machine learning methods have been applied, it is crucial to know how to interpret their outputs and understand what these mean and how they improve the knowledge in each application area. To that end, visualization methods are employed. Such methods are widely used within Big Data scenarios as they are very helpful for all types of graphical interpretations when the Volume, Variety, or Velocity are complex. In Table 6, we present a summary of, and referral to, papers that deal with visualization.

## 4. Conclusion

As indicated by the journal articles and the conference papers we have reviewed in this article, the complexity of Big Data is an urgent topic and the awareness of this is growing. Consequently, there is a lot of research carried out on this, and we will in all likelihood find more and more progress in this field during the next few years.

Additionally, a key issue that we really want to emphasize in this study is the aspects related to Big Data which transcend the academic area and that, therefore, are reflected in the company. An observation is that more than 50% out of 560 enterprises thinks Big Data will help them increase their operational efficiency as well as other things [60]. This indicates that there are a lot of opportunities for Big Data. However, it is also clear that there are many challenges in every phase of the knowledge discovery procedure that need to be addressed in order to achieve a continued and successful progress within the field of Big Data.

As is shown in Figure 1, there are three general approaches when carrying out intensive data processing in IoT architectures: (a) local processing, (b) edge computing, and (c) cloud

TABLE 6: The table summarizes and refers to a representative set of papers focusing on Visualization and Prediction in the context of Big Data.

Reference	Title of paper	Description/Objective
[100]	Beyond the hype: Big Data concepts, methods, and analytics	The main objective described in this paper is analytic methods and how are they used for Big Data, in particular, the ones related to unstructured data.
[101]	Key Performance Indicators: Developing, Implementing, and Using Winning KPIs	This book represents a guide with tools and procedures to discover the KPIs and how they are developed and used.
[102]	The visual organization: data visualization, Big Data, and the quest for better decisions	The paper describes data visualization myths, such as: that all data must be visualized, when in fact only good data should be visualized; visualization will always manifest the right decision or action; and that visualization will lead to certainty.
[103]	Big Data and Visualization: Methods, Challenges and Technology Progress	The paper presents applications, technological progress of Big Data visualization, and discusses challenges of it.
[104]	Big-Data Visualization	A special issue which focus on the current situation and new trends of Big Data Visualization.

computing. The text explained each of these approaches more in detail.

We also explained the knowledge discovery procedure by dividing it into several stages as shown in Figure 2. These steps are IoT Data Gathering, Data Cleaning, Integration, Machine Learning, Data Mining, Classification, Prediction, and Visualization.

We have also discussed that many research papers are focused on the variety of information because this is in itself, in conjunction with integration, one of the most challenging issues when it comes to the IoT. This is also the reason why it is very often also associated with one of the most difficult Vs of Big Data, which is the variety of data.

The trend for the future seems to be that more investigations will be carried out in such areas as (a) techniques for data integration, again the V of Variety; (b) more efficient machine learning techniques on big data, such as Deep Learning and frameworks such as Apache's Hadoop and Spark, that will probably have a crucial importance; and (c) the visualization of the data, with, e.g., dashboards, and more efficient techniques for the visualization of indicators.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors acknowledge the support from the research center Internet of Things and People (IOTAP) at Malmö University in Sweden. This work was also supported by the Spanish Research Agency (AEI) and the European Regional Development Fund (ERDF) under project Cloud-Driver4Industry TIN2017-89266-R.

## References

- [1] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, *Vision and Challenges for Realising the Internet of Things The meaning of things lies not in the things themselves, but in our attitude towards them. Antoine de Saint-Exupéry*, 2010.
- [2] A. Zaslavsky, C. Perera, and D. Georgakopoulos, "Sensing as a Service and Big Data," in *Proceedings of the International Conference on Advances in Cloud Computing (ACC)*, pp. 21–29, 2012.
- [3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [4] M. Chen, S. Mao, and Y. Liu, "Big data: a survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [5] S. Sagioglu and D. Sinanc, "Big data: a review," in *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS '13)*, pp. 42–47, May 2013.
- [6] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [7] C. Hayashi, *What is Data Science? Fundamental Concepts and a Heuristic Example*, 1998.
- [8] V. Dhar, "Data science and prediction," *Communications of the ACM*, vol. 56, no. 12, pp. 64–73, 2013.
- [9] L. Vangelova, "Data scientist," *Scientific Teaching*, vol. 79, no. 6, pp. 66–67, 2012.
- [10] J. Hardin, R. Hoerl, N. J. Horton et al., "Data science in statistics curricula: preparing students to "think with data"," *The American Statistician*, vol. 69, no. 4, pp. 343–353, 2015.
- [11] T. Hey, S. Tansley, and K. Tolle, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, 2009.
- [12] P. Bevington and D. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*, 1993.
- [13] I. C. L. Ng and S. Y. L. Wakenshaw, "The Internet-of-Things: Review and research directions," *International Journal of Research in Marketing*, vol. 34, no. 1, pp. 3–21, 2017.
- [14] T. Saarikko, U. H. Westergren, and T. Blomquist, "The Internet of Things: Are you ready for what's coming?" *Business Horizons*, vol. 60, no. 5, pp. 667–676, 2017.
- [15] D. Gil, A. Ferrández, H. Mora-Mora, and J. Peral, "Internet of things: a review of surveys based on context aware intelligent services," *Sensors*, vol. 16, no. 7, article 1069, 2016.
- [16] R. Pérez-delHoyo, C. García-Mayor, H. Mora, V. Gilart-Iglesias, and M. D. Andújar-Montoya, "Improving urban accessibility: A methodology for urban dynamics analysis in smart, sustainable and inclusive cities," *International Journal of Sustainable Development and Planning*, vol. 12, no. 3, pp. 357–367, 2017.



- [17] Z. Lv, X. Li, W. Wang, B. Zhang, J. Hu, and S. Feng, "Government affairs service platform for smart city," *Future Generation Computer Systems*, vol. 81, pp. 443–451, 2018.
- [18] J. Macke, R. M. Casagrande, J. A. R. Sarate, and K. A. Silva, "Smart city and quality of life: Citizens' perception in a Brazilian case study," *Journal of Cleaner Production*, vol. 182, pp. 717–726, 2018.
- [19] H. March, "The Smart City and other ICT-led technomimaginations: Any room for dialogue with Degrowth?" *Journal of Cleaner Production*, vol. 197, pp. 1694–1703, 2018.
- [20] H. Mora, M. Signes-Pont, D. Gil, and M. Johnsson, "Collaborative Working Architecture for IoT-Based Applications," *Sensors*, vol. 18, no. 6, p. 1676, 2018.
- [21] W. Lee and A. Sharma, "Smart sensing for IoT applications," in *Proceedings of the 13th IEEE International Conference on Solid-State and Integrated Circuit Technology, ICSICT 2016*, pp. 362–364, October 2016.
- [22] Institute of Electrical and Electronics Engineers, *IEEE Std 1451.0™ 2007, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats*, 2007.
- [23] T. Islam, S. C. Mukhopadhyay, and N. K. Suryadevara, "Smart Sensors and Internet of Things: A Postgraduate Paper," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 577–584, 2017.
- [24] J. Mendes Jr., M. Vieira, M. Pires, and S. Stevan Jr., "Sensor Fusion and Smart Sensor in Sports and Biomedical Applications," *Sensors*, vol. 16, no. 10, p. 1569, 2016.
- [25] H. Mora, D. Gil, R. M. Terol, J. Azorín, and J. Szymanski, "An IoT-Based Computational Framework for Healthcare Monitoring in Mobile Environments," *Sensors*, vol. 17, no. 10, p. 2302, 2017.
- [26] M. Masoudinejad, A. K. R. Venkatapathy, J. Emmerich, and A. Riesner, *Smart Sensing Devices for Logistics Application*, Springer, Cham, Switzerland, 2017.
- [27] C. Chen, M. Lin, and X. Guo, "High-level modeling and synthesis of smart sensor networks for Industrial Internet of Things," *Computers & Electrical Engineering*, vol. 61, pp. 48–66, 2017.
- [28] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the context of industry 4.0: a review," *Engineering Journal*, vol. 3, no. 5, pp. 616–630, 2017.
- [29] Y. Su, X. Meng, Q. Kang, and X. Han, "Dynamic Virtual Network Reconfiguration Method for Hybrid Multiple Failures Based on Weighted Relative Entropy," *Entropy*, vol. 20, no. 9, p. 711, 2018.
- [30] C.-W. Tseng, F.-H. Tseng, Y.-T. Yang, C.-C. Liu, and L.-D. Chou, "Task Scheduling for Edge Computing with Agile VNFs On-Demand Service Model toward 5G and Beyond," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7802797, 13 pages, 2018.
- [31] A. Yassine, H. Rahimi, and S. Shirmohammadi, "Software defined network traffic measurement: Current trends and challenges," *IEEE Instrumentation & Measurement Magazine*, vol. 18, no. 2, pp. 42–50, 2015.
- [32] H. Tahaei, R. Salleh, S. Khan, R. Izard, K.-K. R. Choo, and N. B. Anuar, "A multi-objective software defined network traffic measurement," *Measurement*, vol. 95, pp. 317–327, 2017.
- [33] X. Wang, C. Xu, G. Zhao, K. Xie, and S. Yu, "Efficient Performance Monitoring for Ubiquitous Virtual Networks Based on Matrix Completion," *IEEE Access*, vol. 6, pp. 14524–14536, 2018.
- [34] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [35] M. Satyanarayanan, "The emergence of edge computing," *The Computer Journal*, vol. 50, no. 1, pp. 30–39, 2017.
- [36] *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are What You Will Learn*, 2015.
- [37] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, pp. 27–42, 2017.
- [38] E. Ahmed and M. H. Rehmani, "Mobile Edge Computing: Opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2017.
- [39] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The Case for VM-based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [40] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.
- [41] L. J. M. Nieuwenhuis, M. L. Ehrenhard, and L. Prause, "The shift to Cloud Computing: The impact of disruptive technology on the enterprise software business ecosystem," *Technological Forecasting & Social Change*, vol. 129, pp. 308–313, 2018.
- [42] A. Celesti, D. Mulfari, M. Fazio, M. Villari, and A. Puliafito, "Exploring Container Virtualization in IoT Clouds," in *Proceedings of the 2nd IEEE International Conference on Smart Computing, SMARTCOMP 2016*, pp. 1–6, May 2016.
- [43] M. Giacobbe, R. Di Pietro, A. Longo Minnola, and A. Puliafito, "Evaluating Information Quality in Delivering IoT-as-a-Service," in *Proceedings of the 2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 405–410, June 2018.
- [44] S. Satpathy, B. Sahoo, and A. K. Turuk, "Sensing and Actuation as a Service Delivery Model in Cloud Edge centric Internet of Things," *Future Generation Computer Systems*, vol. 86, pp. 281–296, 2018.
- [45] R. Lovas, A. Farkas, A. C. Marosi et al., "Orchestrated Platform for Cyber-Physical Systems," *Complexity*, vol. 2018, Article ID 8281079, 16 pages, 2018.
- [46] H. Mora, V. Gilart-Iglesias, R. Pérez-del Hoyo, and M. Andújar-Montoya, "A Comprehensive System for Monitoring Urban Accessibility in Smart Cities," *Sensors*, vol. 17, no. 8, p. 1834, 2017.
- [47] A. M. Osman, "A novel big data analytics framework for smart cities," *Future Generation Computer Systems*, vol. 91, pp. 620–633, 2019.
- [48] J. Santos, B. Volckaert, T. Wauters, and F. de Turck, "Fog Computing: Enabling the Management and Orchestration of Smart City Applications in 5G Networks," *Entropy*, vol. 20, no. 1, p. 4, 2017.
- [49] F. Mora-Gimeno, H. Mora-Mora, D. Marcos-Jorquera, and B. Volckaert, "A Secure Multi-Tier Mobile Edge Computing Model for Data Processing Offloading Based on Degree of Trust," *Sensors*, vol. 18, no. 10, p. 3211, 2018.
- [50] A. Monteriù, M. Prist, E. Frontoni et al., "A Smart Sensing Architecture for Domestic Monitoring: Methodological Approach and Experimental Validation," *Sensors*, vol. 18, no. 7, p. 2310, 2018.
- [51] M. Afrin, M. Razzaque, I. Anjum, M. Hassan, and A. Alamri, "Tradeoff between User Quality-Of-Experience and Service Provider Profit in 5G Cloud Radio Access Network," *Sustainability*, vol. 9, no. 11, p. 2127, 2017.

- [52] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure—physical sensor management with virtualized sensors on cloud computing," in *Proceedings of the 13th International Conference on Network-Based Information Systems (NBIS '10)*, pp. 1–8, September 2010.
- [53] J. J. Calbimonte, H. Jeung, O. Corcho, and K. Aberer, "Semantic Sensor Data Search in a Large-Scale Federated Sensor Network," *Semantic Sensor Networks*, pp. 14–29, 2011.
- [54] J. Liu, J. Chen, L. Peng, X. Cao, R. Lian, and P. Wang, "An open, flexible and multilevel data storing and processing platform for very large scale sensor network," in *Proceedings of the 2012 14th International Conference on Advanced Communication Technology (ICACT)*, 2012.
- [55] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [56] M. Hassanaliyagh, A. Page, T. Soyata et al., "Health monitoring and management using internet-of-things (IoT) sensing with cloud-based processing: opportunities and challenges," in *Proceedings of the IEEE International Conference on Services Computing, SCC 2015*, pp. 285–292, IEEE, July 2015.
- [57] S. Li, L. D. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, 2013.
- [58] H. Mora-Mora, V. Gilart-Iglesias, D. Gil, and A. Sirvent-Llamas, "A computational architecture based on RFID sensors for traceability in smart cities," *Sensors*, vol. 15, no. 6, pp. 13591–13626, 2015.
- [59] F. Liu, Y. Liu, D. Jin, X. Jia, and T. Wang, "Research on Workshop-Based Positioning Technology Based on Internet of Things in Big Data Background," *Complexity*, vol. 2018, Article ID 875460, 11 pages, 2018.
- [60] C. L. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Information Sciences*, vol. 275, pp. 314–347, 2014.
- [61] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of 'big data' on cloud computing: review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.
- [62] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, pp. 1–21, 2015.
- [63] S. del Río, V. López, J. M. Benítez, and F. Herrera, "On the use of MapReduce for imbalanced big data using Random Forest," *Information Sciences*, vol. 285, pp. 112–137, 2014.
- [64] A. Halevy, A. Doan, and Z. Ives, *Principles of Data Integration*, Elsevier, 2012.
- [65] A. Y. Halevy, "Answering queries using views: A survey," *The VLDB Journal*, vol. 10, no. 4, pp. 270–294, 2001.
- [66] R. Pottinger and A. Halevy, "MiniCon: A scalable algorithm for answering queries using views," *The VLDB Journal*, vol. 10, no. 2–3, pp. 182–198, 2001.
- [67] M. Brundage, *Xquery: The XML Query Language*, 2004.
- [68] R. Goldman, J. McHugh, and J. Widom, "From semistructured data to XML: Migrating the Lore data model and query language," *Markup Languages: Theory and Practice*, vol. 2, no. 2, pp. 153–163, 1999.
- [69] V. Josifovski, M. Fontoura, and A. Barta, "Querying XML streams," *The VLDB Journal*, vol. 14, no. 2, pp. 197–210, 2005.
- [70] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," *ACM SIGMOD Record*, vol. 33, no. 4, pp. 65–70, 2004.
- [71] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to map between ontologies on the semantic web," in *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*, pp. 662–673, ACM, May 2002.
- [72] T. J. Green, "Containment of Conjunctive Queries on Annotated Relations," *Theory of Computing Systems*, vol. 49, no. 2, pp. 429–459, 2011.
- [73] B. Glavic and G. Alonso, "Perm: Processing provenance and data on the same data model through query rewriting," in *Proceedings of the 25th IEEE International Conference on Data Engineering, ICDE 2009*, pp. 174–185, China, April 2009.
- [74] H. Gonzalez, A. Halevy, C. S. Jensen et al., "Google fusion tables: web-centered data management and collaboration," in *Proceedings of the 1st ACM symposium*, p. 175, June 2010.
- [75] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava, "Global detection of complex copying relationships between sources," *Proceedings of the VLDB Endowment*, vol. 3, no. 1–2, pp. 1358–1369, 2010.
- [76] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, 2011.
- [77] A. Maté, H. Llorens, E. De Gregorio et al., "A novel multidimensional approach to integrate big data in business intelligence," *Journal of Database Management*, vol. 26, no. 2, pp. 14–31, 2015.
- [78] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: part I," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, 2014.
- [79] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the internet of things: Literature review and challenges," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.
- [80] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.
- [81] J. Dean, *Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners*, John Wiley & Sons, 2014.
- [82] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1–5, 2012.
- [83] Y. Guo, Z. Yang, S. Feng, and J. Hu, "Complex Power System Status Monitoring and Evaluation Using Big Data Platform and Machine Learning Algorithms: A Review and a Case Study," *Complexity*, vol. 2018, Article ID 8496187, 21 pages, 2018.
- [84] X. Liu, Y. Zhou, and X. Chen, "Mining Outlier Data in Mobile Internet-Based Large Real-Time Databases," *Complexity*, vol. 2018, Article ID 9702304, 12 pages, 2018.
- [85] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [86] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [87] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

- [88] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.
- [89] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the the 40th International Conference on Software Engineering*, pp. 303–314, May 2018.
- [90] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 56–64, 2017.
- [91] Y. Gu, Y. Chen, J. Liu, and X. Jiang, "Semi-supervised deep extreme learning machine for Wi-Fi based localization," *Neurocomputing*, vol. 166, pp. 282–293, 2015.
- [92] W. Zhang, K. Liu, W. Zhang, Y. Zhang, and J. Gu, "Deep Neural Networks for wireless localization in indoor and outdoor environments," *Neurocomputing*, vol. 194, pp. 279–287, 2016.
- [93] L. Li, K. Ota, and M. Dong, "When Weather Matters: IoT-Based Electrical Load Forecasting for Smart Grid," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 46–51, 2017.
- [94] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.
- [95] Y. Huang, X. Ma, X. Fan, J. Liu, and W. Gong, "When deep learning meets edge computing," in *Proceedings of the 2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pp. 1–2, October 2017.
- [96] S. Sharma, K. Chen, and A. Sheth, "Towards practical privacy-preserving analytics for IoT and cloud based healthcare systems," *IEEE Internet Computing*, vol. 22, pp. 42–51, 2018.
- [97] A. Abeshu and N. Chilamkurti, "Deep Learning: The Frontier for Distributed Attack Detection in Fog-To-Things Computing," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 169–175, 2018.
- [98] C. Liu, Y. Cao, Y. Luo et al., "A New Deep Learning-Based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 249–261, 2018.
- [99] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "A scalable and privacy-aware IoT service for live video analytics," in *Proceedings of the 8th ACM Multimedia Systems Conference, MMSys 2017*, pp. 38–49, June 2017.
- [100] A. Gandomi and M. Haider, "Beyond the hype: big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.
- [101] D. Parmenter, *Key Performance Indicators: Developing, Implementing, And Using Winning KPIs*, 2015.
- [102] P. Simon, *The visual organization: data visualization, Big Data, and the quest for better decisions*, 2014.
- [103] L. Wang, G. Wang, and C. A. Alexander, "Big data and visualization: methods, challenges and technology progress," *Digital Technologies*, vol. 1, no. 1, pp. 33–38, 2015.
- [104] D. Keim, H. Qu, and K. Ma, "Big-Data Visualization," *IEEE Computer Graphics and Applications*, vol. 33, no. 4, pp. 20–21, 2013.

## Review Article

# Recent Progress of Anomaly Detection

**Xiaodan Xu,<sup>1,2</sup> Huawen Liu,<sup>1,3</sup> and Minghai Yao <sup>2</sup>**

<sup>1</sup>Department of Computer Science, Zhejiang Normal University, Jinhua 321004, China

<sup>2</sup>College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310000, China

<sup>3</sup>Shanghai Key Laboratory of Intelligent Information Processing, Shanghai 200433, China

Correspondence should be addressed to Minghai Yao; ymh@zjut.edu.cn

Received 10 October 2018; Revised 11 December 2018; Accepted 31 December 2018; Published 13 January 2019

Guest Editor: David Gil

Copyright © 2019 Xiaodan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Anomaly analysis is of great interest to diverse fields, including data mining and machine learning, and plays a critical role in a wide range of applications, such as medical health, credit card fraud, and intrusion detection. Recently, a significant number of anomaly detection methods with a variety of types have been witnessed. This paper intends to provide a comprehensive overview of the existing work on anomaly detection, especially for the data with high dimensionalities and mixed types, where identifying anomalous patterns or behaviours is a nontrivial work. Specifically, we first present recent advances in anomaly detection, discussing the pros and cons of the detection methods. Then we conduct extensive experiments on public datasets to evaluate several typical and popular anomaly detection methods. The purpose of this paper is to offer a better understanding of the state-of-the-art techniques of anomaly detection for practitioners. Finally, we conclude by providing some directions for future research.

## 1. Introduction

Anomaly analysis is of great interest to diverse research fields, including data mining and machine learning. It aims to identify those regions from data whose behaviours or patterns do not conform to expected values [1]. The unexpected behaviours, which are significantly different from those of the remainder of the given data, are commonly called anomalies. Notwithstanding, there is no widely acceptable formal definition of this concept. In the literature, an anomaly is also referred to as an outlier, a discordant object, an exception, an aberration, or a peculiarity, depending on specific application scenarios [1–5].

Identifying interesting or unexpected patterns is very important to many domains, such as decision making, business intelligence, and data mining. For example, an abnormal network transmission may imply that a computer system is attacked by hackers or viruses, an anomalous transaction of a credit card may imply unauthorized usage, and unexpected geological activity in nature can be a precursor of an earthquake or tsunami. Due to this fact, anomaly detection has a wide variety of applications, including public medical health,

credit card fraud and network intrusion, and data cleaning [3, 5].

With the emergence of new technologies, data collected from real-world scenarios are becoming larger and larger, not only in size, but also in dimensionality. The high-dimensional property makes the data objects almost equidistant to each other. This implies that any data objects become very close as the dimensionality of data increases, resulting in the meaningless nature of their respective distances [4]. In this case, traditional anomaly detection methods cannot effectively handle high-dimensional data. In addition, most of the traditional detection methods assume that the data have the same type of features. However, the data in reality often have different feature types, such as numerical, binary, categorical, or nominal. This leads to an increased difficulty in anomaly detection.

Since anomaly detection has a wide range of potential applications, a great number of detection algorithms have been witnessed during the past decades. In this paper, we briefly review the latest works and place especial focuses on the ones for those complex data with high dimensionalities and mixed types. Generally, the existing anomaly detection



TABLE 1: A brief description of the anomaly detection methods.

Types	Descriptions & Typical methods	Advantages	Disadvantages
Neighbour-based detection	Identifying anomalies by using neighbourhood information. Typical examples include $k$ NN[9], $k$ NNW[10], LOF[11], LoOP[12], ODIN[13], RBDA[6], etc.	(i) Independent of the data distributions (ii) Intuitively understood and easily interpreted	(i) Sensitive to parameters (ii) Relatively poor performance
Subspace-based detection	Finding anomalies by sifting through different feature subsets. Representative examples include SOD[7], Zhang et al. [14, 15], RODS[16], OR[17], Muller et al. [18], etc.	(i) High efficiency (ii) Very effectiveness in some cases	(i) Finding the relevant feature subspaces for outliers is nontrivial and difficult
Ensemble-based detection	Integrating various anomaly detection results to achieve a consensus. Representatives are FB [19], HiCS [8], Stein et al. [20], Zimek et al. [21], Passillas et al. [22], and so on.	(i) High accuracy (ii) Less sensitive	(i) Inefficient (ii) Choosing the right meta-detectors is difficult
Mixed-type detection	Making a unified model for different data types, or taking each data type separately. Classical examples have LOADED [23], ODMAD [24], Zhang et al. [25], Lu et al. [26], Do et al. [27], and so on.	(i) Capable of handling the data with different types (ii) Relatively high accuracy	(i) Obtaining the correlation structures of features is difficult (ii) High complexity

techniques can be grouped into three categories: neighbour-based, subspace-based, and ensemble-based detection methods, depending on the techniques used. Table 1 summaries brief descriptions of the anomaly detection algorithms, including their definitions, pros, and cons.

In the literature, there are several survey papers (e.g., [1–5]) proposed for anomaly detection. However, they concern different aspects of anomaly detection. For example, [1] only reviews traditional outlier detection algorithms, while [2] places its focus on ensemble learning ones. The detection methods for specific application domains like network data and temporal data have been overviewed in [5] and [3], respectively. Unlike the surveys above, this paper only involves the latest and popular anomaly detection methods for the data with high dimensionality and mixed types, on which the classical detection methods cannot handle very well. Besides, this paper also offers more information related to anomaly detection, such as public datasets and widely used metrics. These aspects, however, have not been considered in the other papers. Additionally, this paper has made a comprehensively experimental comparison of several popular detection algorithms. The paper aims to help practitioners to better understand the state-of-the-art techniques of anomaly detection.

The remainder of this paper is organized as follows. Section 2 presents a survey of anomaly detection for the complicated data, including neighbour-based, subspace-based, and ensemble-based detection techniques. Section 3 provides evaluation metrics commonly used in the anomaly detection techniques, followed by experimental comparisons of the popular detection methods in Section 4. Section 5 concludes the paper.

## 2. Methodology

How to effectively identify outliers from the high-dimensional or mixed-type data is a fundamental and challenging issue in outlier detection. Recently, a rich number of detection algorithms have been developed to alleviate the problems. Roughly speaking, they can be divided into three categories, that is, neighbour-based (e.g., RBDA [6]), subspace-based (e.g., SOD [7]), and ensemble-based methods (e.g., HiCS [8]). The neighbour-based outlier detection methods mainly exploit the neighbourhood information of a given data object to determine whether it is far from its neighbours or its density is low or not. The subspace-based detection methods identify anomalies by sifting through different feature subsets in an ordered way. Unlike the routine algorithms, the ensemble-based ones combine the outputs of several detection algorithms or base detectors into a unified output by using integrated strategies. Table 1 briefly summarizes descriptions of the anomaly detection techniques.

**2.1. Neighbour-Based Detection.** The basic idea of the neighbour-based anomaly detection methods is to identify outliers by virtue of the neighbourhood information. Given a data object, the anomaly score is defined as the average distance ( $k$ NN [9]) or weighted distance ( $k$ NNW [10]) to its  $k$  nearest neighbours. Another strategy is to take the local outlier factor (LOF) [11] as the measurement of anomaly degree, in which the anomaly score was measured relative to its neighbourhoods. Based on LOF and LoOP [12] provided for each object an outlier probability as score, which is easily interpretable and can be compared over one data set. In



ODIN (Outlier Detection using Indegree Number) [13], an object is defined as an outlier if it participates in most neighbourhoods in  $k$ NN graph.

Note that all the neighbour-based detection methods mentioned above are independent of the distributions of the data and capable of detecting isolated objects. However, their performance heavily relies on the distance measures, which become unstable or meaningless in high-dimensional spaces. To cope with this problem, a feasible solution is to consider the ranking of neighbours, because, for each object, the ranking of its nearest neighbours is still meaningful to the nature of high-dimensional data. The underlying assumption is that two objects would most likely become nearest neighbours or have similar neighbours if they were generated from the same mechanism [7]. Following this idea, RBDA (Rank-Based Detection Algorithm) [6] takes the ranks of each object in its neighbours as the proximity degree of the object. For each object  $s \in D$ , let  $N_k(s)$  be the  $k$  nearest neighbours of  $s$ . The anomaly degree of  $s$  is defined as follows:

$$A_k(s) = \frac{\sum_{p \in N_k(s)} r_p(s)}{\|N_k(s)\|} \quad (1)$$

where  $r_p(s)$  is the rank of  $s$  among the neighbours of  $p$ . According to Eq. (1), one may observe that if  $s$  ranks behind the neighbours  $N_k(s)$ , it has a higher anomaly degree and would have a high probability of being considered an anomaly. RBDA does not consider the distance information of objects with regard to their neighbours, which would be useful in some cases; MRD (Modified-Ranks with Distance) [28] does. MRD takes both the ranks and the distances into account when estimating the anomaly scores of objects.

A special kind of the nearest neighbour, called the reverse neighbour, is also used to represent the proximate relationship among objects. For any object  $s$ ,  $p$  is called a reverse neighbour of  $s$  if  $s$  is one of the nearest neighbours of  $p$ , and vice versa, that is,  $s \in N_k(p)$  and  $p \in N_k(s)$ . The intuitive idea is that if an object has fewer reverse nearest neighbours, it is more likely to be an anomaly. Radovanovic et al. [29] adopted the reverse nearest neighbours to estimate the anomaly scores for each object. Bhattacharya et al. [30] continued this method even further by adopting both the reverse neighbours and the ranks of nearest neighbours to measure the anomaly score for each candidate object. Zhang et al. [31] estimated the anomaly scores using the number of the shared nearest neighbours of objects. Tang and He [32] exploited three kinds of neighbourhoods, including  $k$  nearest neighbours, reverse nearest neighbours, and shared nearest neighbours, to determine the anomaly scores in the local kernel density estimation. The neighbour ranking-based methods are sensitive to  $k$ , where different  $k$  values will yield different results. In addition, assigning the right value to  $k$  for a specific application is not trivial. To this end, Ha et al. [33] adopted a heuristic strategy to select an appropriate value for  $k$  using an iterative random sampling procedure. The assumption is that outlying objects are less likely to be selected than inlying objects in random sampling. Thus, greater inlier scores, called the observability factor (OF), should be given to the selected objects in each sampling. After

several iterations of random sampling, the OF score of each object is estimated by counting its occurrence times in its neighbourhood. Based on the OF scores, the value of  $k$  can be appropriately assigned as the entropy of the observability factors.

**2.2. Subspace-Based Detection.** Anomalies often exhibit unusual behaviours in one or more local or low-dimensional subspaces. The low-dimensional or local abnormal behaviours would be masked by full dimensional analysis [34]. Zimek et al. [4] noted that, for an object in a high-dimensional space, only a subset of relevant features offers valuable information, while the rest are irrelevant to the task. The existence of the irrelevant features may impede the separability of the anomaly detection model. However, the anomaly detection techniques discussed so far identify anomalous objects from the whole data space with full dimensions. Thus, identifying anomalies from appropriate subspaces appears to be more interesting and efficient.

Subspace learning is a popular technique to handle high-dimensional problems in the literature. It is also extensively studied in anomaly analysis. The anomaly detection methods based on subspace techniques aim at finding anomalies by sifting through different subsets of dimensions in an ordered way. These methods have two kinds of representations: the sparse subspace methods [14, 16, 35, 36] and the relevant subspace methods [7, 15, 17, 18, 37].

The sparse subspace techniques project all objects in a high-dimensional space onto one or more low-dimensional and sparse subspaces. The objects falling into the sparse subspaces are considered anomalies because the sparse subspaces have abnormally lower densities. It is noted that exploring the sparse projections from the entire high-dimensional space is a time-consuming process. To alleviate this problem, Aggarwal and Yu [36] exploited an evolutionary algorithm to improve the exploration efficiency, where a subspace with the most negative scarcity coefficients was considered a space projection. However, the performance of the evolutionary algorithm heavily relies on some factors, such as the initial populations, the fitness functions, and selection methods.

Subspace representation and encoding are another studied topic for sparse subspace techniques. As a typical example, Zhang et al. [14] utilized the concept of lattice to represent the relationship of subspaces, where the subspaces with low density coefficients are regarded as sparse ones. This kind of method shows advantages in the performance and the completeness. However, constructing the concept lattice of subspaces is complex, leading to low efficiency. Dutta et al. [16] leveraged the technique of sparse encoding to project objects to a manifold space with a linear transformation, making the space sparse.

The relevant subspace methods exploit local information represented as relevant features to identify anomalies. For instance, OR (Out Ranking) [17] extend a subspace clustering model to rank outliers in heterogeneous high-dimensional data. SOD (Subspaces Anomaly Detection) [7] is a typical example of the relevant subspace learning methods. It first explores several correlation datasets by using the shared nearest neighbours for each object and then determines an

axis-parallel subspace on each correlation dataset by linear correlation such that each feature has low variance in the subspace. Unlike SOD, Muller et al. [37] used the relevant relationships of features from the correlation dataset to determine the subspace. Specifically, they obtained relevant subspaces by examining the relevant relationships of features with the Kolmogorov-Smirnov test [38]. Then, the anomaly degree of the object was calculated by multiplying the local anomaly scores in each relevant subspace. It can be easily observed that this kind of detection method is computationally expensive. The limitation of this method is that it requires a great number of local data to detect the trend of deviation.

**2.3. Ensemble-Based Detection.** Ensemble learning is widely studied in machine learning [39, 40]. Since it has a relatively better performance than other related techniques, ensemble learning is also frequently used for anomaly detection. As we know, none of the outlier detection methods can discover all anomalies in a low-dimensional subspace due to the complexity of the data. Thus, different learning techniques or even multiple subspaces are required simultaneously, where the potential anomalies are derived by ensemble techniques. In the literature, there are two ensemble strategies frequently adopted for anomaly analysis, that is, summarizing the anomaly scores and selecting the best one after ranking. For anomaly analysis, feature bagging and subsampling are extensively studied.

The FB (Feature Bagging) detection method [19] aims to train multiple models on different feature subsets sampled from a given high-dimensional space and then combines the model results into an overall decision. A typical example of this technique is the work done by Lazarevic and Kumar [19], in which feature subsets are randomly selected from the original feature space. On each feature subset, the score of each object is estimated with an anomaly detection algorithm. Then, the scores for the same object are integrated as the final score. Nguyen et al. [41] used different detection techniques, rather than the same one, to estimate anomaly scores for each object on random subspaces.

Keller et al. [8] proposed a flexible anomaly detection method that decouples the process of anomaly mining into two steps, that is, subspace search and anomaly ranking. The subspace search aims at obtaining high contrast subspaces (HiCS) using the Monte Carlo sampling technique, and, then, the LOF scores of objects are aggregated upon the obtained subspaces. Stein [20] extended this by first gathering the relevant subspaces of HiCS and then calculated the anomaly scores of objects using local anomaly probabilities (LoOP) [12], in which the neighbourhood is selected in the global data space.

The subsampling technique obtains training objects from a given collection of data without replacement. If implemented properly, it can effectively improve the performance of detection methods. For example, Zimek et al. [21] applied the technique of random subsampling to obtain the nearest neighbours for each object and then estimated its local density. This ensemble method, coupled with an anomaly detection algorithm, has a higher efficiency and provides a diverse set of results.

There are several anomaly detection methods that consider both feature bagging and subsampling. For example, Pasillas-Diaz et al. [22] obtained different features at each iteration via feature bagging and then calculated the anomaly scores for different subsets of data via subsampling. However, the variance of objects is difficult to obtain using feature bagging, and the final results tend to be sensitive to the size of subsampled datasets.

**2.4. Mixed-Type Detection.** It is worthy of remark that most of the anomaly detection methods mentioned above can only handle numerical data, resulting in poor robustness. In real-world applications, categorical and nominal features are ubiquitous; that is, categorical and numerical features are mixed within the same dataset [34]. Such mixed-type data pose great challenges to the existing detection algorithms. For mixed-type data, a common and simple strategy is to discretize numerical features and then treat them as categorical ones so that the detection methods for categorical data can be applied directly. While this practice seems to be a good solution, it may lose important information, that is, the correlations between features, leading to poor performance.

By now, a great number of detection methods have been developed to handle categorical data in the literature [42]. For example, He et al. [43] proposed a frequent pattern-based anomaly detection algorithm, where the potential anomalies were measured using a frequent pattern anomaly factor. As a result, the data objects that contained infrequent patterns could be considered anomalies. Contrastively, Otey et al. [44] developed a nonfrequent item set-based anomaly detection algorithm. Despite the pattern-based methods being suitable for handling categorical data, they are time consuming for general cases. Wu and Wang [45] estimated the frequent pattern anomaly factors based on nonexhaustive methods by mining a small number of patterns instead of all the frequent patterns. Koufakou and Georgiopoulos [46] considered the condensed representation of nonderivable item sets in their algorithm, which is a compact representation and can be obtained less expensively.

There are a lot of studies attempting to handle mixed-type data directly in the literature. Typical examples include LOADED [23], RELOADED, and ODMAD [24]. For instance, LOADED calculates an anomaly score for each object by using the support degrees of item sets for categorical features and correlation coefficients for numerical features [23]. RELOAD employs naive Bayes classifiers to predict abnormalities of categorical features. Finally, ODMAD treats categorical and numerical features separately. Specifically, it first calculates anomaly scores for categorical features using the same classification algorithm as LOADED. The objects, which are not identified as anomalies at this step, will be examined over numerical features with the cosine similarity [24]. Bouguessa [47] modelled the categorical and numerical feature space by using a mixture of bivariate beta distributions. The objects having a small probability of belonging to any components are regarded as anomalies.

The correlations of features have also been taken into consideration. For example, Zhang and Jin [25] exploited the concept of patterns to determine anomalies. In this method,

a pattern is a subspace formed by a particular category and all numerical features. Within this context, the patterns are learned via logistic regression. The objects would be considered anomalies if the probability returned by the model is far from a specific pattern. Lu et al. [26] took pairwise correlations of mixed-type features into consideration and presented a generalized linear model framework for anomaly analysis. Additionally, the t-student distribution was also used to capture variations of anomalies from normal objects. More recently, Do et al. [27] calculated anomaly scores for each object using the concept of free energy derived from a mixed-variant restricted Boltzmann machine. Since this well captured the correlation structures of mixed-type features through the factoring technique, it has a relatively high performance.

### 3. Evaluation Measurements

Unlike the problems of classification, evaluating the performance of the anomaly detection algorithms is more complicated. On the one hand, the ground truth of anomalies is unclear because real anomalies are rare in nature. On the other hand, the anomaly detection algorithms often output an anomalous score for each object. The objects with relatively large anomalous scores are considered anomalies if they are larger than a given threshold. Setting a proper threshold for each application in advance is relatively difficult. If the threshold is set too large, true anomalies would be missed; otherwise, some objects that are not true anomalies would be mistakenly taken as potential anomalies.

In general, the following measurements have often been used to evaluate the performance of the anomaly detection methods.

- (1) **Precision at  $t$  ( $P@t$ )** [48]: given a dataset  $D$  consisting of  $N$  objects,  $P@t$  is defined as the proportion of the true anomalies,  $A \subseteq D$ , to the top  $t$  potential anomalies identified by the detection method; that is,

$$P@t = \frac{|a \in A \mid \text{rank}(a) \leq t|}{t} \quad (2)$$

It is noticeable that the value of  $t$  is difficult to set for each specific application. A commonly used strategy is to set  $t$  as the number of anomalies in the ground truth.

- (2) **R-precision** [49]: this measurement is the proportion of true anomalies within the top  $t$  potential anomalies identified, where  $t$  is the number of ground truth anomalies. Since the number of true anomalies is relatively small in comparison to the size of the dataset, the value of R-precision would be very small. Thus, it contains less information.
- (3) **Average precision (AP)** [50]: instead of evaluating the precision individually, this measurement refers to

the mean of precision scores over the ranks of all anomaly objects:

$$AP = \frac{1}{|a|} \sum_{t=1}^{|a|} P@t. \quad (3)$$

where  $P@t$  is the precision at  $t$ , that is, Eq. (2).

- (4) **AUC** [4]: the receiver operating characteristic (ROC) curve is a graphical plot of the true positive rate against the false positive rate, where the true (false) positive rate represents the proportion of anomalies (inliers) ranked among the top  $t$  potential anomalies. Zimek et al. [4] noted that, for a random model, the ROC curve tends to be diagonal, while, for a good ranking model, it will output true anomalies first, leading to the area under the corresponding curve (AUC) covering all available space. Thus, the AUC is often used to numerically evaluate the performances of anomaly detection algorithms.
- (5) **Correlation coefficient**: correlation coefficients, such as Spearman's rank similarity and Pearson correlation, are also taken as evaluation measurements. This kind of measurement places more emphasis on the potential anomalies ranked at the top by incorporating weights. More details about the measurements of correlation coefficients can be found in [51] and references therein.
- (6) **Rank power (RP)**: Both the precision and AUC criteria do not consider characteristics of anomaly ranking. Intuitively, an anomaly ranking algorithm will be regarded as more effective if it ranks true anomalies in the top and normal observations in the bottom of the list of anomaly candidates. The rank power is such a metric and evaluates the comprehensive ranking of true anomalies. The formal definition is

$$\text{RankPower} = \frac{n(n+1)}{2 \sum_{i=1}^n R_i} \quad (4)$$

where  $n$  is the number of anomalies in the top  $t$  potential objects and  $R_i$  is the rank of the  $i$ -th true anomaly. For a fixed value of  $t$ , a larger value indicates better performance. When the  $t$  anomaly candidates are true anomalies, the rank power equals one.

### 4. Experimental Comparisons

As discussed above, various anomaly detection algorithms have been developed. For better understanding the characters of the detection methods, in this section we make an experimental comparison of the popular anomaly detection algorithms.

**4.1. Experimental Settings.** In the literature, two kinds of data, that is, synthetic and real-world datasets, were often reported to evaluate the performance of the anomaly detection methods. The former is generated under the contexts of specific

TABLE 2: Experimental datasets used in our experiments.

Dataset	N(A)	Attribute	Anomalies	Source
ALOI	50000(1508)	27	The 1508 of rare objects	UCI [53]
Arcene	100(44)	10000	The cancer patients	UCI [53]
Ionosphere	351(126)	32	The ‘bad’ class	UCI [53]
KDDCup99	48113(200)	38	The U2R class	UCI [53]
PenDigits	9868(20)	16	The fourth class	UCI [53]
Sonar	208(97)	60	The rock object	UCI [53]
WDBC	367(10)	30	The malignant class	UCI [53]
Waveform	3443(100)	21	The ‘0’ class	UCI [53]
Ann-thyroid	7129(534)	21	The hyper and subnormal classes	ELKI [54]
Arrhythmia	450(206)	259	The arrhythmia class	ELKI [54]
HeartDisease	270(120)	13	The affected patients class	ELKI [54]
Pima	768(268)	8	The Diabetes class	ELKI [54]
SpamBase	4209(1681)	57	The non-spam class	ELKI [54]
ALLAML	38(11)	7129	The AML class	EBD [56]
DLBCL	77(19)	7129	The FL morphology class	EBD [56]
Gisette	550(50)	5000	The normal class	EBD [56]
Lung_MPM	181(31)	12533	The MPM class	EBD [56]
Ovarian	253(162)	15154	The Ovarian Cancer class	EBD [56]

constraints or conditions. Wang et al. [52] provided several synthetic datasets with anomalies for different scenarios. The real-world datasets are offered at public sources such as UCI Machine Learning Repository [53] and ELKI toolkits [54]. However, the datasets publicly available are initially used for classification purposes. Hence, they should be preprocessed, making them suitable for the anomaly detection tasks. Two strategies are frequently adopted during the preprocessing stage [55]. The classes with rare data will be regarded as anomalies and the remaining as normal ones, if they have explicitly semantic meanings. Otherwise, one of the classes will be randomly selected as the anomalies.

To make a fair comparison, our experiments were carried out on 18 real-world datasets. They were downloaded from the UCI Machine Learning Repository [53], the ELKI toolkit [54], and ELVIRA Biomedical Dataset Repository (EBD) [56]. A brief summary of the datasets is presented in Table 2, where the “N (A)” column refers to the numbers of normal objects and anomalies, respectively. We performed the pre-processed operation on the datasets as suggested in [55]. For example, the fourth class (‘4’) in *PenDigits* consisting of 9,868 objects was considered anomalies, while the remaining as normal objects in our experiments.

The experiments compared nine popular anomaly detection algorithms, including *k*NN (*k* Nearest Neighbours) [9], LOF (Local Anomaly Factor) [11], LoOP (Local Anomaly Probabilities) [12], ODIN (Outlier Detection using Indegree Number) [13], RBDA (Rank-Based Detection Algorithm) [6], OR (Out Rank) [17], SOD (Subspace Anomaly Degree) [7], FB (Feature Bagging) [19], and HiCS (High Contrast Subspaces) [8]. They stand for the three kinds of the anomaly detection methods as mentioned above. For example, *k*NN, ODIN, LOF, LoOP, and RBDA belong to the neighbour-based detection methods and OR and SOD are the subspace-based

detection methods. FB and HiCS are the ensemble-based detection methods.

In our experiments, two metrics, that is, R-precision and AUC, were adopted to evaluate the detection algorithms. For the remaining four metrics, we have not presented here, because similar conclusions were found. The comparison experiments were conducted with the ELKI toolkit. The parameters involved within the anomaly detection algorithms were assigned to default values as recommended in the literature. The experiments were performed on a PC with 2.8 GHz of CPU clock rate and 4 GB of main memory.

**4.2. Experimental Results.** Table 3 provides the R-precision performance of the anomaly detection algorithms on the experimental datasets. Since the main memory was quickly consumed when RBDA, FB, and OR run on the *ALOI* and *KDDCup99* datasets, their experimental results were unavailable and presented as “/” in Table 3.

From the experimental results in Table 3, one may observe that the neighbour-based methods had relatively stable performance, while the ensemble-based methods, for example, HiCS, performed unsteadily in many cases. For instance, *k*NN and RBDA achieved relatively good performance on eight datasets. Even HiCS had worse performance on four of them, for example, *PenDigits*, *KDDCup99*, *Ann-thyroid*, and *DLBCL*, but it achieved the highest R-precisions on *Waveform*, *WDBC*, and *Ovarian*. The reason is that the ensemble-based detection methods tend to be sensitive to the size of datasets subsampled from the original ones. Since OR is heavily dependent on the quantities of feature subspaces, it obtained the highest values on *Ann-thyroid* and the lowest values on *Sonar*, *Waveform*, *Arrhythmia*, and *Spambase*. For the high-dimensional datasets, that is, *Arcene*, *ALLAML*, *DLBCL*, *Gisette*, *Lung\_MPM*, and *Ovarian*, *k*NN,



TABLE 3: R-precisions of the anomaly detection algorithms where  $k=7$  for the neighbours.

Dataset	$k$ NN	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	0.16	0.24	0.20	0.22	/	0.06	0.21	/	/
Ionosphere	0.65	0.57	0.46	0.65	0.88	0.15	0.69	0.77	0.69
KDDCup99	0.09	0.24	0.15	0.20	/	/	0.44	/	/
PenDigits	0.01	0.05	0.05	0.05	0.01	0.01	0.05	0.01	0.01
Sonar	0.46	0.55	0.55	0.63	0.53	0.45	0.52	0.49	0.57
WDBC	0.30	0.30	0.40	0.30	0.41	0.6	0.60	0.7	0.70
Waveform	0.07	0.05	0.06	0.04	0.10	0.03	0.06	0.13	0.21
Arrhythmia	0.67	0.64	0.66	0.66	0.73	0.60	0.65	0.65	0.61
Ann-thyroid	0.05	0.04	0.04	0.04	0.06	0.14	0.01	0.04	0.01
HeartDisease	0.57	0.53	0.49	0.56	0.48	0.49	0.52	0.43	0.50
Pima	0.50	0.45	0.39	0.52	0.41	0.39	0.52	0.35	0.46
SpamBase	0.52	0.43	0.40	0.45	0.38	0.38	0.43	0.39	0.48
Arcene	0.52	0.29	0.39	0.42	0.43	0.45	0.48	0.40	0.45
ALLAML	0.40	0.36	0.36	0.36	0.39	0.36	0.36	0.36	/
DLBCL	0.21	0.09	0.21	0.21	0.24	0.21	0.21	0.21	0.16
Gisette	0.14	0.12	0.10	0.14	0.15	0.14	0.16	0.16	0.10
Lung_MPM	0.52	0.29	0.39	0.42	0.54	0.45	0.48	0.49	0.45
Ovarian	0.54	0.59	0.56	0.60	0.61	0.60	0.57	0.53	0.62

RBDA, SOD, and OR had good performance, where OR and SOD were more stable. Indeed, these contain many irrelevant features, which makes those subspace-based methods more effective. It can also be observed that RBDA was better than  $k$ NN and ODIN, for RBDA took the neighbour ranking, instead of the distances, into account which is more suitable for the high-dimensional datasets.

The compared algorithms, except OR, take  $k$ NN as their baseline. As we know,  $k$ NN heavily relies on the number of neighbours  $k$ . To reveal the impact of  $k$  on the performance, we performed a comparison experiment among these methods with different  $k$  values. Tables 4 and 5 show the AUC scores of the anomaly detection algorithms with  $k=10$  and  $k=50$ , respectively. Since the experimental results on the high-dimensional datasets (i.e., *Arcene*, *ALLAML*, *DLBCL*, *Gisette*, *Lung\_MPM*, and *Ovarian*) were still unavailable after three hours' running, they were not provided in Table 5.

According to the results, we know that the detection performance of the comparison algorithms was heavily dependent on the number of neighbours and varied greatly when  $k$  assigned different values. To further illustrate this fact, we conducted additional experiments by performing the detection algorithms on *Arrhythmia*, *Waveform*, and *WDBC* with  $k$  varying from 10 to 50. The experimental results are illustrated as Figure 1.

As shown in Figure 1, the performance of RBDA, FB, and SOD was relatively stable, although they took use of  $k$ NN as their baselines. In fact, SOD exploits  $k$ NN to obtain the relative subspace information, while FB ensembles all informative features found by  $k$ NN. As a result,  $k$  had less impact on them. On the other hand,  $k$ NN, ODIN, LoOP, and LOF heavily relied on the values of  $k$ . For example, the AUC values from ODIN varied greatly on all three datasets with the different values of  $k$ . HiCS had unsteady performance in

many cases. For instance, it was less affected by  $k$  on *WDBC*, while sensitive to  $k$  on *Arrhythmia*. The reason is that, in our experiments, the basis detector of HiCS was also  $k$ NN, leading to its performance relying on  $k$ , although it is an ensemble anomaly detection algorithm.

Another interesting fact is that, on the *WDBC* and *Waveform* datasets, the AUC values of the compared algorithms varied greatly. Indeed, the ratios of anomalies to normal objects within these two datasets are relatively small (2.7% and 2.9% for *WDBC* and *Waveform*, respectively). Consequently, the anomaly detection algorithms were more sensitive to  $k$ . In contrast, on the datasets with high anomaly proportions, for example, *Arrhythmia* (45.7% anomalies), the AUC scores of the anomaly detection algorithms were less sensitive to  $k$ . Similar situations can be found for the other datasets. Due to the limitation of space, they will not be presented here one by one.

Computational efficiency is another important aspect for the practical applications of the anomaly detection methods. We carried out an additional experiment to compare the computational efficiencies of the anomaly detection algorithms. Table 6 records the elapsed time (s) of the anomaly detection algorithms on the experimental datasets.

The elapsed time in Table 6 shows that the neighbour-based detection methods, using the metrics of both distances (e.g.,  $k$ NN and ODIN) and densities (e.g., LOF and LoOP), had relatively higher efficiencies. However, the ensemble-based detection methods, especially HiCS, took too much time to detect anomalies. As a matter of fact, they construct lots of individual detectors before identifying outliers. For the subspace-based detection algorithms, their efficiencies are dependent on the techniques adopted. For example, SOD, which exploits neighbours to explore relative subspaces, is more efficient than OR.



TABLE 4: AUC of the anomaly detection algorithms with  $k=10$  for the neighbours.

Dataset	$kNN$	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	0.66	0.80	0.78	0.80	/	0.57	0.72	/	/
Ionosphere	0.49	0.51	0.57	0.71	0.89	0.24	0.76	0.88	0.80
KDDCup99	0.70	0.60	0.59	0.81	/	/	0.91	/	/
PenDigits	0.90	0.88	0.90	0.88	0.56	0.47	0.91	0.80	0.81
Sonar	0.60	0.60	0.61	0.66	0.60	0.49	0.51	0.57	0.59
WDBC	0.64	0.80	0.69	0.76	0.89	0.96	0.90	0.94	0.98
Waveform	0.53	0.52	0.48	0.54	0.70	0.57	0.63	0.73	0.73
Arrhythmia	0.75	0.68	0.73	0.72	0.73	0.68	0.71	0.73	0.69
Ann-thyroid	0.52	0.50	0.50	0.52	0.69	0.54	0.47	0.72	0.54
HeartDisease	0.52	0.48	0.46	0.59	0.52	0.55	0.61	0.52	0.46
Pima	0.59	0.49	0.49	0.62	0.58	0.54	0.65	0.50	0.54
SpamBase	0.58	0.50	0.51	0.53	0.47	0.46	0.55	0.48	0.52
Arcene	0.46	0.46	0.45	0.46	0.46	0.52	0.47	0.40	0.49
ALLAML	0.71	0.66	0.69	0.69	0.70	0.70	0.72	0.69	/
DLBCL	0.40	0.39	0.40	0.42	0.40	0.41	0.36	0.41	0.40
Gisette	0.56	0.55	0.58	0.56	0.57	0.58	0.71	0.58	0.44
Lung_MPM	0.80	0.63	0.73	0.73	0.71	0.69	0.71	0.73	0.75
Ovarian	0.32	0.38	0.38	0.46	0.43	0.43	0.37	0.38	0.44

TABLE 5: AUC of the anomaly detection algorithms with  $k=50$  for the neighbours, where the performance on *Arcene*, *ALLAML*, *DLBCL*, *Gisette*, *Lung\_MPM*, and *Ovarian* was not given, for it was still unavailable after three hours' running.

Dataset	$kNN$	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	0.59	0.75	0.74	0.77	/	0.57	0.71	/	/
Ionosphere	0.48	0.50	0.55	0.63	0.89	0.24	0.77	0.86	0.75
KDDCup99	0.67	0.66	0.62	0.65	/	/	0.89	/	/
PenDigits	0.65	0.66	0.76	0.84	0.92	0.47	0.88	0.96	0.86
Sonar	0.56	0.55	0.55	0.58	0.59	0.49	0.56	0.54	0.61
WDBC	0.61	0.51	0.61	0.50	0.90	0.96	0.90	0.92	0.98
Waveform	0.48	0.48	0.48	0.51	0.73	0.57	0.63	0.73	0.74
Arrhythmia	0.75	0.71	0.74	0.73	0.74	0.68	0.72	0.75	0.61
Ann-thyroid	0.51	0.52	0.53	0.51	0.66	0.54	0.47	0.65	0.52
HeartDisease	0.53	0.47	0.46	0.57	0.56	0.55	0.60	0.64	0.46
Pima	0.57	0.53	0.52	0.62	0.62	0.54	0.65	0.62	0.61
SpamBase	0.63	0.50	0.52	0.53	0.50	0.46	0.55	0.39	0.54

## 5. Conclusion

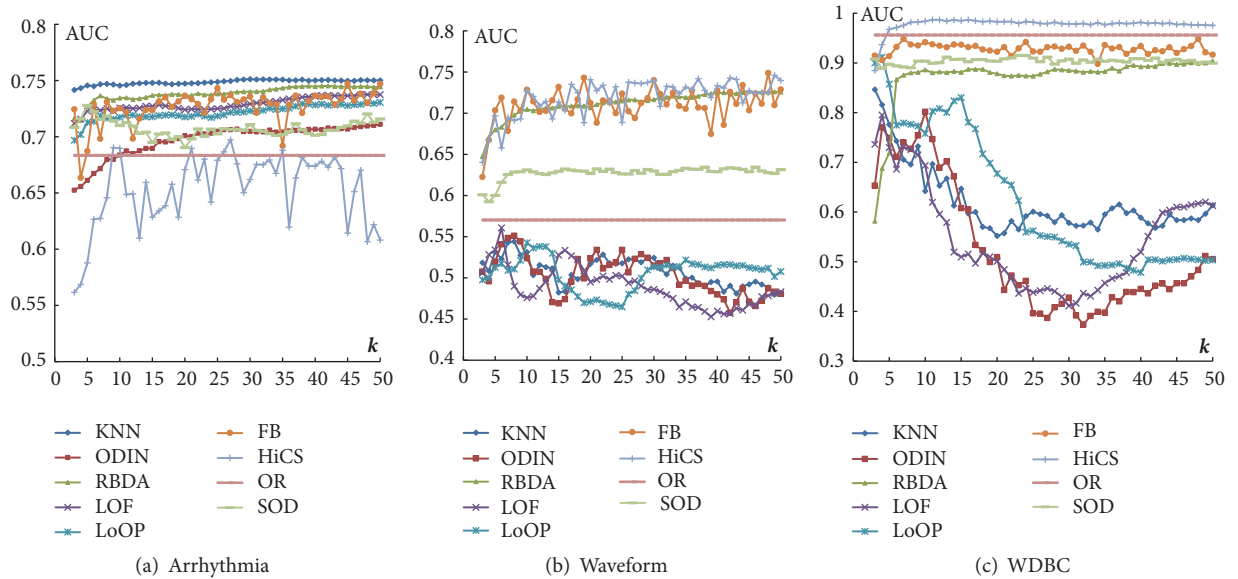
The data collected from real-world applications are becoming larger and larger in size and dimension. As the dimensionality increases, the data objects become sparse, resulting in identifying anomalies being more challenging. Besides, the conventional anomaly detection methods cannot work effectively and efficiently. In this paper, we have discussed typical problems of anomaly detection associated with the high-dimensional and mixed-type data and briefly reviewed the techniques of anomaly detection. To offer a better understanding of the anomaly detection techniques for practitioners, we conducted extensive experiments on publicly available datasets to evaluate the typical and popular anomaly detection methods. Although the progresses of anomaly detection for the high-dimensional and mixed-type data have

been achieved to some extent, there are also several open issues shown as follows that further need to be addressed:

- (1) The traditional distance metrics in the neighbour-based methods cannot work very well for the high-dimensional data because of the equidistant characteristics. The mixed-type features make anomaly detection more difficult. Introducing effective distance metrics for the high-dimensional and mixed-type data is necessary.
- (2) The neighbour-based anomaly detection algorithms are sensitive to nearest neighbours selected for the models. Determining the right number of neighbours is a challenging issue for the neighbour-based methods.

TABLE 6: Time cost (s) of the anomaly detection algorithms, where  $k=10$ .

Dataset	$k$ NN	ODIN	LOF	LoOP	RBDA	OR	SOD	FB	HiCS
ALOI	163.3	131.2	129.8	134.9	/	1650.8	273.3	/	/
Ionosphere	0.02	0.03	0.03	0.05	0.04	2.10	0.05	1.07	171.70
KDDCup99	164.1	166.9	168.4	168	/	/	325.1	/	/
PenDigits	2.13	2.07	2.11	2.20	2.20	414.52	13.8	156.20	2549.47
Sonar	0.02	0.02	0.02	0.02	0.04	1.318	0.04	0.125	17.37
WDBC	0.09	0.05	0.09	0.05	0.11	2.51	0.08	0.34	58.05
Waveform	0.13	0.83	0.81	0.70	0.16	400.09	3.78	60.28	5924.89
Arrhythmia	0.27	0.16	0.17	0.17	0.30	35.68	0.11	2.45	64.74
Annthyroid	1.42	1.46	1.46	1.47	1.50	349.59	4.93	261.23	4514.92
HeartDisease	0.02	0.01	0.03	0.02	0.04	0.45	0.03	0.34	126.21
Pima	0.04	0.04	0.04	0.04	0.07	2.55	0.37	0.51	95.78
SpamBase	2.32	3.85	3.73	2.22	2.80	589.59	4.56	151.48	10453.15
Arcene	0.59	0.60	0.60	0.60	0.62	1158.85	0.70	19.44	7581.04
ALLAML	0.06	0.07	0.06	0.07	0.08	3.73	0.10	0.42	/
DLBCL	0.26	0.26	0.27	0.261	0.32	28.55	0.36	1.69	3496.79
Gisette	9.81	9.06	9.08	9.12	10.80	56754	9.65	67.87	11370.13
Lung_MPM	2.41	2.46	2.46	2.47	2.88	850.21	2.81	17.32	96292.66
Ovarian	5.83	5.79	5.80	5.80	6.76	1521.78	6.26	40.76	1367821.09

FIGURE 1: AUC of the anomaly detection algorithms with different  $k$  varying from 3 to 50 on *Arrhythmia*, *Waveform*, and *WDBC*.

- (3) The subspace-based and ensemble-based methods have relatively good performance if the diversity of the subspaces or base learners is large. For these kinds of anomaly detection methods, how to choose the right subspaces or base learners, as well as their quantities and their combining strategies, is still an open issue.
- (4) Since anomalies are relatively rare and the ground truth is often unavailable in real scenarios, how to effectively and comprehensively evaluate the detection performance is also a challenging issue.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Funding

This work was supported by the National Natural Science Foundation (NSF) of China (61871350, 61572443); the Natural Science Foundation of Zhejiang Province of China (LY14F020019); and Shanghai Key Laboratory of Intelligent Information Processing, Fudan University (IIP-2016-001).

## References

- [1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
- [2] C. C. Aggarwal, "Outlier ensembles," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 49–80, 2017.
- [3] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2014.
- [4] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining*, vol. 5, no. 5, pp. 363–387, 2012.
- [5] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *The Computer Journal*, vol. 54, no. 4, pp. 570–588, 2011.
- [6] H. Huang, K. Mehrotra, and C. K. Mohan, "Rank-based outlier detection," *Journal of Statistical Computation and Simulation*, vol. 83, no. 3, pp. 518–531, 2013.
- [7] H. P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data," in *Proceedings of the Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 831–838, Springer-Verlag, 2009.
- [8] F. Keller, E. Müller, and K. Böhm, "HiCS: High contrast subspaces for density-based outlier ranking," in *Proceedings of the IEEE 28th International Conference on Data Engineering, ICDE 2012*, pp. 1037–1048, USA, April 2012.
- [9] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000.
- [10] F. Angiulli and C. Pizzuti, "Fast Outlier Detection in High Dimensional Spaces," in *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 15–26, Springer-Verlag, Heidelberg, Berlin, Germany, 2002.
- [11] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93–104, 2000.
- [12] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities," in *Proceedings of the ACM 18th International Conference on Information and Knowledge Management (CIKM '09)*, pp. 1649–1652, ACM Press, November 2009.
- [13] H. Ville, I. Karkkainen, and P. Franti, "Outlier Detection Using k-Nearest Neighbour Graph," in *Proceedings of the IEEE International Conference on Pattern Recognition*, vol. 3, pp. 330–433, 2004.
- [14] J. Zhang, Y. Jiang, K. H. Chang, S. Zhang, J. Cai, and L. Hu, "A concept lattice based outlier mining method in low-dimensional subspaces," *Pattern Recognition Letters*, vol. 30, no. 15, pp. 1434–1439, 2009.
- [15] J. Zhang, X. Yu, Y. Li, S. Zhang, Y. Xun, and X. Qin, "A relevant subspace based contextual outlier mining algorithm," *Knowledge-Based Systems*, vol. 99, no. 72, pp. 1–9, 2016.
- [16] J. K. Dutta, B. Banerjee, and C. K. Reddy, "RODS: Rarity based Outlier Detection in a Sparse Coding Framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 483–495, 2016.
- [17] E. Müller, I. Assent, U. Steinhausen, and T. Seidl, "OutRank: Ranking outliers in high dimensional data," in *Proceedings of the 2008 - IEEE 24th International Conference on Data Engineering Workshop, ICDE'08*, pp. 600–603, Mexico, April 2008.
- [18] E. Müller, M. Schiffer, and T. Seidl, "Adaptive outlieriness for subspace outlier ranking," in *Proceedings of the 19th International Conference on Information and Knowledge Management and Co-located Workshops, CIKM'10*, pp. 1629–1632, Canada, October 2010.
- [19] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceedings of the KDD-2005: 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 157–166, USA, August 2005.
- [20] B. Van Stein, M. Van Leeuwen, and T. Back, "Local subspace-based outlier detection using global neighbourhoods," in *Proceedings of the 4th IEEE International Conference on Big Data, Big Data 2016*, pp. 1136–1142, USA, December 2016.
- [21] A. Zimek, M. Gaudet, R. J. G. B. Campello, and J. Sander, "Subsampling for efficient and effective unsupervised outlier detection ensembles," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013*, pp. 428–436, USA, August 2013.
- [22] J. R. Pasillas-Diaz and S. Ratte, "Bagged subspaces for unsupervised outlier detection," *International Journal of Computational Intelligence*, vol. 33, no. 3, pp. 507–523, 2017.
- [23] A. Ghoting, M. E. Otey, and S. Parthasarathy, "LOADED: Link-based outlier and anomaly detection in evolving data sets," in *Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM 2004*, pp. 387–390, UK, November 2004.
- [24] A. Koufakou and M. Georgiopoulos, "A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes," *Data Mining and Knowledge Discovery*, vol. 20, no. 2, pp. 259–289, 2010.
- [25] K. Zhang and H. Jin, "An effective pattern based outlier detection approach for mixed attribute data," in *AI 2010: Advances in Artificial Intelligence*, vol. 6464 of *Lecture Notes in Computer Science*, pp. 122–131, Springer, Berlin, Germany, 2010.
- [26] Y.-C. Lu, F. Chen, Y. Wang, and C.-T. Lu, "Discovering anomalies on mixed-type data using a generalized Student-t based approach," *Expert Systems with Applications*, vol. 28, no. 10, pp. 1–10, 2016.
- [27] K. Do, T. Tran, D. Phung, and S. Venkatesh, "Outlier detection on mixed-type data: an energy-based approach," in *Advanced Data Mining and Applications*, pp. 111–125, Springer International Publishing, Cham, Switzerland, 2016.
- [28] H. Huang, K. Mehrotra, and C. K. Mohan, "Outlier detection using modified-ranks and other variants," *Electrical Engineering and Computer Science* 72, 2011, [https://surface.syr.edu/eecs\\_techreports/72/](https://surface.syr.edu/eecs_techreports/72/).
- [29] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1369–1382, 2015.
- [30] G. Bhattacharya, K. Ghosh, and A. S. Chowdhury, "Outlier detection using neighborhood rank difference," *Pattern Recognition Letters*, vol. 60, pp. 24–31, 2015.
- [31] L. Zhang, Z. He, and D. Lei, "Shared nearest neighbors based outlier detection for biological sequences," *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 12, pp. 1–10, 2012.
- [32] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.

- [33] J. Ha, S. Seok, and J.-S. Lee, "A precise ranking method for outlier detection," *Information Sciences*, vol. 324, pp. 88–107, 2015.
- [34] C. C. Aggarwal, "High dimensional outlier detection: the subspace method," in *Outlier Analysis*, pp. 135–167, Springer, New York, NY, USA, 2013.
- [35] J. Zhang, S. Zhang, K. H. Chang, and X. Qin, "An outlier mining algorithm based on constrained concept lattice," *International Journal of Systems Science*, vol. 45, no. 5, pp. 1170–1179, 2014.
- [36] C. C. Aggarwal and S. Yu, *An Effective and Efficient Algorithm for High-Dimensional Outlier Detection*, Springer-Verlag, New York, NY, USA, 2005.
- [37] E. Muller, M. Schiffer, and T. Seidl, "Statistical selection of relevant subspace projections for outlier ranking," in *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE 2011*, pp. 434–445, Germany, April 2011.
- [38] M. A. Stephens, "Use of the kolmogorov-smirnov, cramér-von mises and related statistics without extensive tables," *Journal of the Royal Statistical Society: Series B*, vol. 32, no. 1, pp. 115–122, 1970.
- [39] A. Zimek, R. J. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: challenges and research questions," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 1, pp. 11–22, 2014.
- [40] C. C. Aggarwal and S. Sathe, "Theoretical Foundations and Algorithms for Outlier Ensembles," *ACM SIGKDD Explorations Newsletter*, vol. 17, no. 1, pp. 24–47, 2015.
- [41] H. V. Nguyen, H. H. Ang, and V. Gopalkrishnan, "Mining outliers with ensemble of heterogeneous detectors on random subspaces," in *Database Systems for Advanced Applications*, vol. 5981, pp. 368–383, Springer, Berlin, Germany, 2010.
- [42] A. Giacometti and A. Soulet, "Frequent pattern outlier detection without exhaustive mining," *Advances in Knowledge Discovery and Data Mining*, pp. 196–207, 2016.
- [43] Z. He, X. Xu, Z. Huang, and S. Deng, "FP-outlier: Frequent pattern based outlier detection," *Computer Science and Information Systems*, vol. 2, no. 1, pp. 103–118, 2005.
- [44] M. E. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets," *Data Mining and Knowledge Discovery*, vol. 12, no. 2-3, pp. 203–228, 2006.
- [45] S. Wu and S. Wang, "Information-theoretic outlier detection for large-scale categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 3, no. 25, pp. 589–602, 2013.
- [46] A. Koufakou, J. Secretan, and M. Georgiopoulos, "Non-derivable itemsets for fast outlier detection in large high-dimensional categorical data," *Knowledge and Information Systems*, vol. 29, no. 3, pp. 697–725, 2011.
- [47] M. Bouguessa, "A practical outlier detection approach for mixed-attribute data," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8637–8649, 2015.
- [48] N. Craswell, "Precision at n," in *Encyclopaedia of Database Systems*, L. Liu and M. Ozsu, Eds., pp. 2127–2128, Springer, Berlin, Germany, 2009.
- [49] N. Craswell, "R-precision," in *Encyclopaedia of Database Systems*, L. Liu and M. Ozsu, Eds., p. 2453, Springer, Berlin, Germany, 2009.
- [50] E. Zhang and Y. Zhang, "Average precision," in *Encyclopaedia of Database Systems*, L. Liu and M. Ozsu, Eds., pp. 192–193, Springer, Berlin, Germany, 2009.
- [51] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*, pp. 1047–1058, USA, April 2012.
- [52] X. Wang, X. L. Wang, Y. Ma, and D. M. Wilkes, "A fast MST-inspired kNN-based outlier detection method," *Information Systems*, vol. 48, pp. 89–112, 2015.
- [53] "UCI Machine Learning Repository," 2007, <http://archive.ics.uci.edu/ml/>.
- [54] "ELKI," 2016, <https://elki-project.github.io/releases/>.
- [55] G. O. Campos, A. Zimek, J. Sander et al., "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.
- [56] "ELVIRA Biomedical DataSet Repository," 2005, <http://leo.ugr.es/elvira/DBCRepository/>.

## Research Article

# Secure UAV-Based System to Detect Small Boats Using Neural Networks

**Moisés Lodeiro-Santiago** <sup>1</sup>, **Pino Caballero-Gil** <sup>1</sup>,  
**Ricardo Aguasca-Colomo** <sup>2</sup> and **Cándido Caballero-Gil** <sup>1</sup>

<sup>1</sup>*Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, Tenerife 38200, Spain*

<sup>2</sup>*Instituto Universitario SIANI-Edificio Central del Parque Científico y Tecnológico, Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain*

Correspondence should be addressed to Moisés Lodeiro-Santiago; [mlodeirs@ull.edu.es](mailto:mlodeirs@ull.edu.es)

Received 17 October 2018; Revised 3 December 2018; Accepted 10 December 2018; Published 2 January 2019

Guest Editor: Magnus Johnsson

Copyright © 2019 Moisés Lodeiro-Santiago et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a system to detect small boats (pateras) to help tackle the problem of this type of perilous immigration. The proposal makes extensive use of emerging technologies like Unmanned Aerial Vehicles (UAV) combined with a top-performing algorithm from the field of artificial intelligence known as Deep Learning through Convolutional Neural Networks. The use of this algorithm improves current detection systems based on image processing through the application of filters thanks to the fact that the network learns to distinguish the aforementioned objects through patterns without depending on where they are located. The main result of the proposal has been a classifier that works in real time, allowing the detection of pateras and people (who may need to be rescued), kilometres away from the coast. This could be very useful for Search and Rescue teams in order to plan a rescue before an emergency occurs. Given the high sensitivity of the managed information, the proposed system includes cryptographic protocols to protect the security of communications.

## 1. Introduction

According to research in the area of political geography, EU governments are immersed in a difficult battle against irregular migration [1]. This phenomenon was fuelled by the 9/11 attacks and is becoming identified as a “vector of insecurity,” so some countries are using it to justify drastic acts of immigration measures [2]. On the other hand, the so-called Transnational Clandestine Actors [3] operate across national borders, evading state laws, becoming rich at the cost of the despair suffered by many people living in “poor countries” and violating their basic human rights. Thus, this scenario leads to catastrophic consequences most times, with innumerable loss of human lives, mainly because of the vulnerability of the means used to travel [4]. Data from the European External Borders Agency FRONTEX [5] indicate that between 2015 and 2016 more than 800,000 people irregularly passed through the Mediterranean to Europe seeking refuge [6]. The number of irregular immigrants who

cross the sea increases every year compared to the number of them who do it on foot [7]. To face this situation, the EU has been investing more and more resources in the detection of these flows [5].

Various advances in research and various works have been recently presented that deal with the problem of irregular immigration, [8–12]. These works make use of various image processing techniques for the detection of people and boats in the sea, demonstrating that it is feasible to use technology in combination with UAV systems to face these problems.

This work presents a system to cope with the aforementioned problem, making use of a system based on a UAV for capturing several sequences of images with a smartphone on it. The UAV uses an optimal route planning system such as the one presented in [13] adapted to marine and coastal environments. These images are sent in real time through antennas using LTE/4G coverage to a remote cloud server, where they are processed by a Convolutional Neural Network



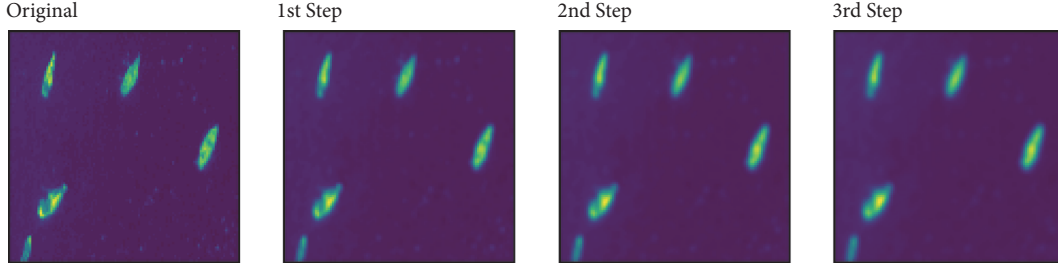


FIGURE 1: 3-step process on 5x5 kernels for noise removal.

(CNN) that has been previously trained to detect three types of objects: ships, pateras (or cayucos), and people (on land or at sea). These images may be used in a system for the detection and alert of various security and emergency. For this purpose, an Automatic Identification System (AIS) is used to compare each image of a detected ship, according to its GPS position, with a marine traffic database in order to find out whether it is a registered ship or not.

The security of the application against manipulation or attacks is structured in different levels depending on the used technology. For transmission via LTE (4G) in coastal areas with coverage, the SNOW 3G [14] algorithm is used for integrity protection and flow encryption [15]. Furthermore, in order to avoid image manipulation by inserting watermarks that may disable the ability to identify images [16], an algorithm has been designed that first adds white noise to the image and then compresses it using a JPEG compression [17]. This proposal not only prevents the attack but also, according to performed tests, increases the accuracy of the network.

Furthermore, to protect data transmission systems, an Attribute-Based Encryption (ABE) is used. In the bibliography, several proposals can be found that use ABE as a light cryptographic technique to deal with problems different from the one described in this work. On the one hand, in the paper [18], ABE is used to access scalable media where the complete subcontracting process returns plaintext to smartphone users. On the other hand, in [19], ABE is proposed to access health care records using a mobile phone with decryption process outsourced to cloud servers.

The present document is structured as follows: Section 2 discusses the use of neural networks, particularly convolutional ones. Section 3 defines the different stages of the proposed system and some experiments during data collection, training, and obtaining results. Section 4 describes the security layer, with emphasis on possible attacks and countermeasures applied to this type of system. Finally, Section 5 closes the paper with some conclusions.

## 2. Image Processing

Image processing is the first essential step of the proposed solution to the aforementioned problem. Image processing is a methodology that has been widely applied in the field of research for the identification of objects, tracking of objects, detection of diseases, etc. For many years in the field of

artificial intelligence, neural networks have gained strength and, in image processing, the CNN have been used.

CNNs are a type of network created specifically for image and video processing. The relationship between CNNs and neural networks is quite simple because both have the same elements (neurons, weights, and biases). Mainly, the operation in these networks is based on taking the inputs and encoding some properties of the architecture CNNs passing the results from layer to layer in order to obtain classification data.

The special thing about is the mathematical convolution that is applied. A  $C$  convolution is a mathematical operation on two functions  $f$  and  $g$  to produce into a new function that represents the magnitude in which  $f$  is a superimposed and a transferred and inverted version of  $g$ . For example, the convolution of  $f$  and  $g$  is denoted by  $f * g$  and is defined as the integral of the product of both functions after moving one of them at a distance of  $t$ . Thus, the  $C$  convolution is defined as  $C = (f * g) = \int_{-\infty}^{\infty} f(\eta)g(t - \eta)d\eta$ . In CNN, the first argument of the convolution usually refers to the input and the second argument refers to the kernel (a fixed-size matrix with positive or negative numerical coefficients, with an anchor point within the matrix that, as a general rule, is located in the middle of the matrix). The common output of applying a convolution with a kernel is treated as a new feature map  $H$  such that  $H(x, y) = \sum_{i=0}^{M_i-1} \sum_{j=0}^{M_j-1} I(x + i - a_i, y + j - a_j)K(i, j)$ .

Figure 1 illustrates an example of the evolution when applying a 3-step process on a 5x5 kernel with values of 0.04 for removing residual noise. Although at first glance there are no significant differences, the image becomes blurrier as the different steps (from 1 to 3) are applied (this can be seen better on the edges of the pateras).

The layers of compression or pooling layers are applied along the neural network to reduce the space of the representation by making use of a number of parameters and the same computation of the network. This process is applied independently in each step in depth within the network, taking as reference the inputs. It is also used for the reduction of data overfitting. There are different types of pooling although among the most best is the one known as Max-Pooling. In the Max-Pooling process, having as input an array of  $N \times N$  and  $M \times M$  grids is taken such that  $M \subseteq \{1..N\}$ . The resulting number of horizontal and vertical steps will determine the discard threshold for the new layers.

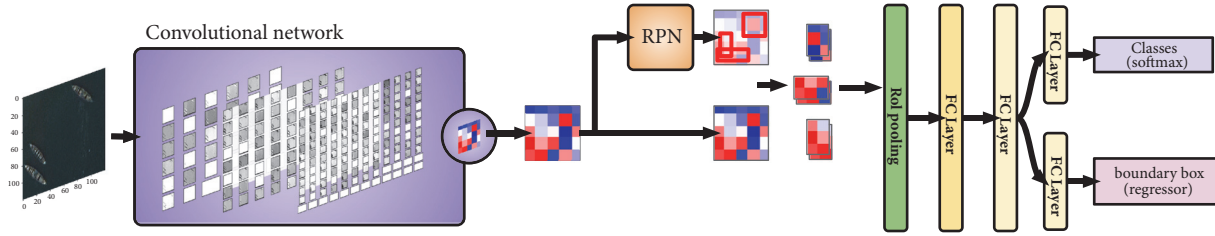


FIGURE 2: Faster R-CNN flow.

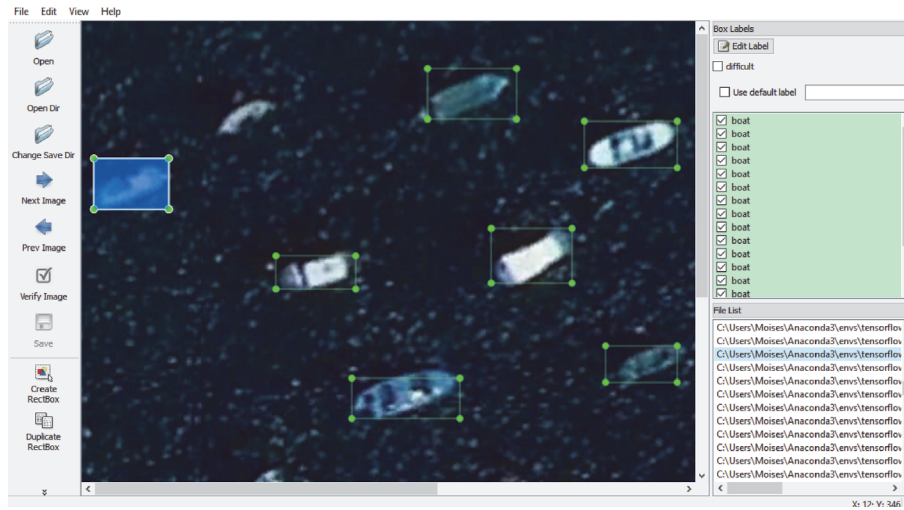


FIGURE 3: Classification tool.

In order to get the model used in this work performance improvements, modern object detector based on CNNs knowing as Faster R-CNN [20, 21]. This model depends in part on an external region used for selective search [22]. The Faster R-CNN model has a design similar to that of Fast R-CNN [23], so that it jointly optimises classification and bounding box regression task. Moreover, the proposed region is replaced by a deep learning network and the Region of Interest (RoI) is replaced by features maps. Thus, the new Region Proposal Network (RPN) is more efficient for the generation of RoIs because for every window location, multiple possible regions are generated based on a bounding box ratio. In other words, a visualisation is made on each location in the characteristics map, considering a number  $k$  of different boxes centred on it (a longer area, a fatter one, a longer one, etc.). This is shown in Figure 2 in an example, where a softmax classifier composes a Fully Connected (FC) Layer.

### 3. Proposed System

This section describes the procedures performed after the acquisition of the data, explaining the processing of the images as well as the detailed training process, providing information on each of the obtained results and discussing why to choose one or other result to continue the experiments. Finally, some conclusions results are provided through a demonstration image with correct classification ratios.

**3.1. Data Collection and Classification.** For the collection of images of pateras, due to the lack of accessibility to boats of the type patera or cayuco in a massive way, we have opted for the gathering of information from of an image, by using Google Earth software, always looking for a height with respect to sea level of 100 meters (height at which the drone would fly in the experiment) and maintaining a totally perpendicular view. After obtaining a dataset of 3,347 images corresponding to three classes of the problem, we opted for a classification of each of the objects of the various images, taking into consideration that an image can have one or several objects. According to the applied dataset, complexity, and capabilities and based on the available documentation, the majority of the authors refer to the Pareto Principle [24] as the most convenient. Thus, the ratio 80/20, which is the most used has been considered an adequate proportion for the train/test neural network.

For the classification of each image, we have used the software known as Labellmg (see Figure 3), created in Python, for supervised training. This software creates a layer that separates each image into different objects limited by a bounded box corresponding to the position ( $x, y$ ), width, and height of the box. This process has been performed for the three types of object considered in this work.

As a result of this classification, the XML files corresponding to all the objects within the images are obtained. These XML files are used later in the training.

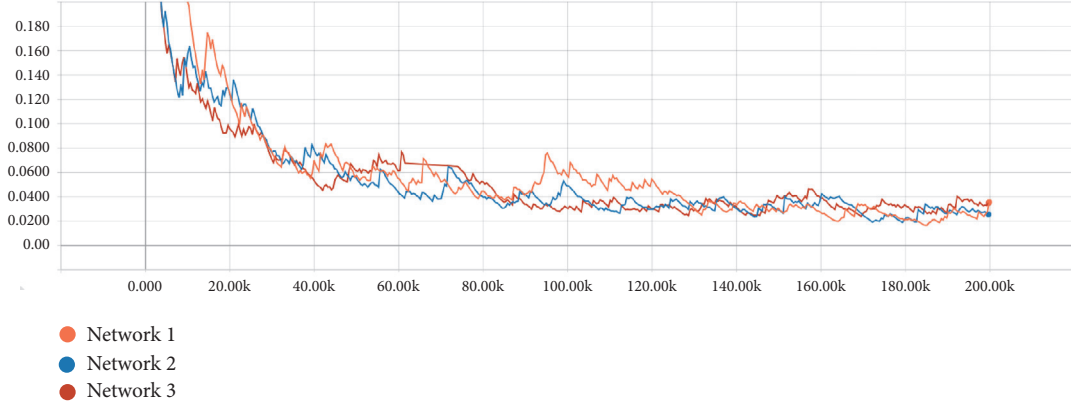


FIGURE 4: First trainings of CNN maintaining coefficients with different datasets.

TABLE 1: Object distribution for imbalanced networks.

Classes	Person	Patera	Boat
Train	1791	284	646
Test	466	96	64

3.2. *Training.* The total time for each training stage of the neural network with the conditions described above has been an average of 16 hours using a GPU Nvidia 1050. The following guidelines were followed in order to obtain the best possible network for detection:

- (i) Train the same neural network three times with the same learning coefficients, regulation coefficients and activation function. The reason for doing only three trainings instead of 5, 10, 30, or more is because there is no rule of thumb that shows an exact trend in the result of the network. Therefore, to rule out strange behaviours, each network was trained 3 times to see empirically that the three results (even starting from a random vector in the direction of the gradient descent) gave similar results. With this, false positives can be discarded when compared with other networks.
- (ii) In all training sessions, a number of 200.0K iterations was established for each of the networks.
- (iii) The reflected values have a tendency of 0.95, which means that they are not real values but that is the value of tendency in each instant calculated from previous values (so it can be higher or lower).
- (iv) In each training, the network changed the dataset on the basis that the dataset has a total of 3347 images divided approximately between a ratio of 80% training and 20% for testing resulting in a distribution as shown in Table 1. For each training of each neural network, the initial set of training and test has been altered to demonstrate the efficiency of the neural network from different datasets. This type of randomness has been applied to demonstrate the functionality and efficiency of the system in methods

TABLE 2: Classification Loss for Networks 1, 2, and 3.

Network	Classification Loss
1	0.03553
2	0.02554
3	0.03494

based on stochastic decisions. Given that the applied methodology is stochastic and random, performing permutations on the dataset allows obtaining different results, which is used to obtain better datasets to be used as a basis for other trainings.

- (v) The used programming language was Python 3 for the machine learning, and the TensorFlow software library for the neural network oriented environment [25].

The use of tools such as TensorFlow (among other frameworks for analysis in convolutional networks) has been widely used in recent years for the detection of patterns in images. One of the most notable current works is its use in medical environments to face deadly diseases such as cancer [26], which slightly improves the performance obtained by specialists in dermatology. Among others, neural networks have been used in the marine environment [27] to identify marine fouling using the same framework. Although according to several studies [28, 29] the use of unbalanced datasets in neural networks is detrimental, we have opted for an approach to a real problem where a balanced data network is not available. At the end of this section, a comparison was made with a balanced network. It can be appreciated that although the results of the balanced network are better, they do not differ too much.

Once the three neural networks finished training (see Figure 4) the final results shown in Table 2 were obtained based on the Classification Loss (CL). The CL equation [21] is optimised for a multitask loss function and represented as  $L(\{p_i\}, \{t_i\}) = (1/Ncls) \sum_i L_{cls}(p_i, p_i^*) + \lambda(1/Nreg) \sum_i L_{reg}(t_i, t_i^*)$ , where the terms of the equation represent the loss in classification over two classes (depending on whether the object exists or not), while the second term is

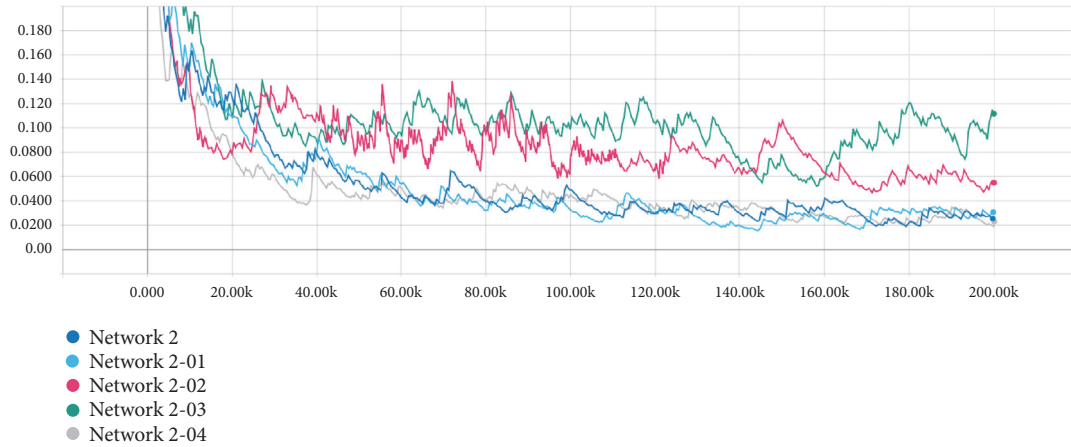


FIGURE 5: Sensitivity analysis based on data from Network 2.

TABLE 3: Learning rates for new trainings using dataset from Network 2.

Network	Learning Rate
2 - 01	0.003
2 - 02	0.00003
2 - 03	0.00001
2 - 04	0.001

the loss of regression of the bounding boxes where an object is found.

When interpreting Figure 4, it must be taken into consideration that the used parameter was the CL. This value is better the closer it gets to zero. Initially, the graph starts with discrete values between 0 and 1, where 1 is a total loss and 0 is a no loss.

Considering these results, the next step in obtaining an improved network was to copy the data from the best network (number 2) and perform a sensitivity analysis on 4 new trainings varying the training coefficients. In the image of Figure 5 we can see the behaviour of the different networks (including the original network number 2). The variation of the coefficients was of multiplicative type, altering the different coefficients of learning rate according to the distribution shown in Table 3.

The learning rate is a measure that represents the size of the vector that is applied in the descent of the gradient when applying the partial derivatives. On the one hand, if the learning rate is very large, the steps will be larger and will approach a solution faster. However, this can be a mistake because it could jump without coming to a good approximation to the solution. On the other hand, if it is very small, it will take longer to train but it will come up with a solution. That is why the study was carried out with different learning rates, to check which learning rates come closest to a good solution in less time. Thus, using these results, the best final coefficient (see Table 4) with respect to the classification loss was the Network 2-04 (grey line in Figure 5), with a

TABLE 4: CL for Networks 1, 2, 3, and 4 trained with original Network 2.

Network	Classification Loss
2 (original)	0.02554
2 - 01	0.03075
2 - 02	0.05487
2 - 03	0.06422
2 - 04	0.02310

TABLE 5: CL for Networks 1 and 3 using the parameters of best Network 2-04.

Network	Classification Loss
2 - 04 (best)	0.02310
1 with 0204 coefficients	0.02559
3 with 0204 coefficients	0.03043

TABLE 6: Balanced training and test dataset.

Classes	Person	Patera	Boat
Train	300	300	300
Test	80	80	80

coefficient lower than 0.02310 which means that in 97.8% cases it produces correct classifications.

Afterwards, the best parameters of the best Network (2-04) were exported to the initial sets of networks 1 and 3 to see if a better result could be obtained by applying the coefficients of the best network so far (see Figure 6). In that image we can see how, although for a short time, the best network is still Network 2 with the fourth training (2-04). However, with these parameters, Networks 1 and 3 (1-0204 and 3-0204) improve slightly with respect to the initial Networks 1 and 3 values (see Table 5 and Figure 7).

After having obtained a result that is feasible in terms of experiments, we decided to make an analysis on a neural network with balanced data (80% training and 20% test), this time having the following random distribution of images (see Table 6).

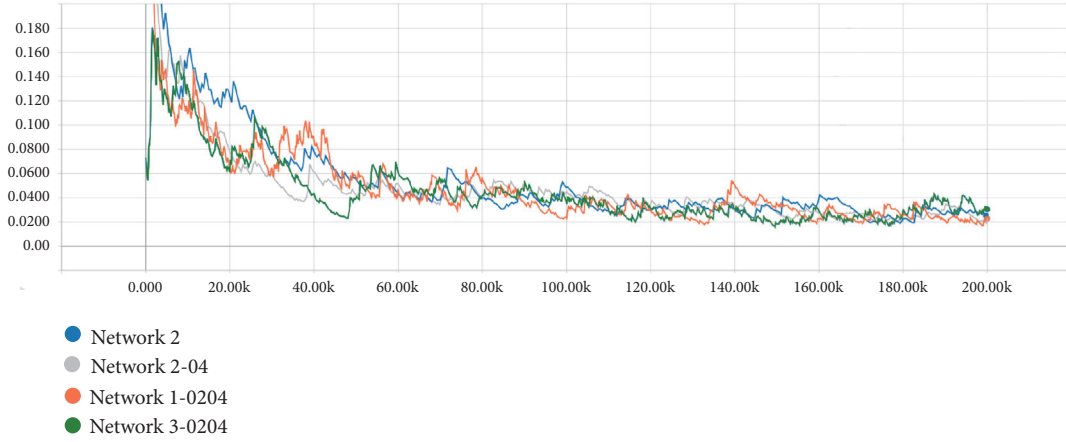


FIGURE 6: Sensitivity analysis of Networks 1 and 3 with data from Network 2-4.

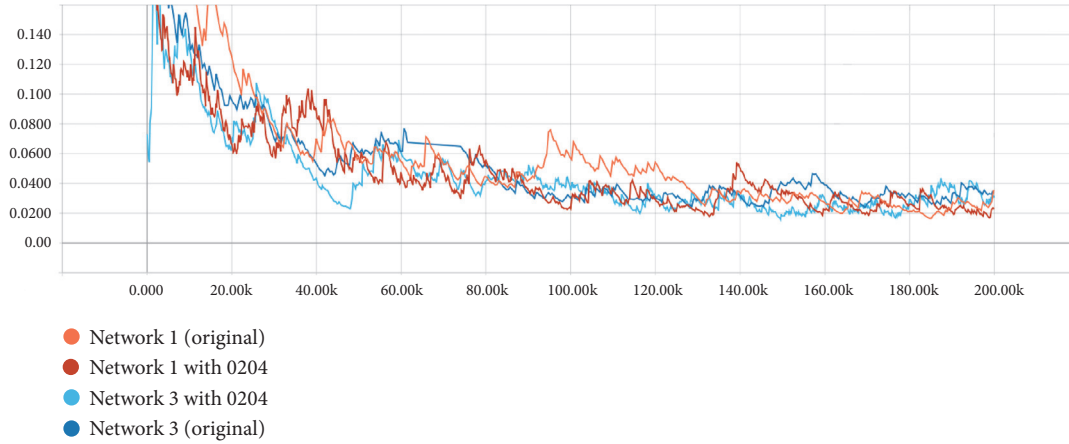


FIGURE 7: Comparison of original Networks 1 and 3 with respect to Network 2-04.

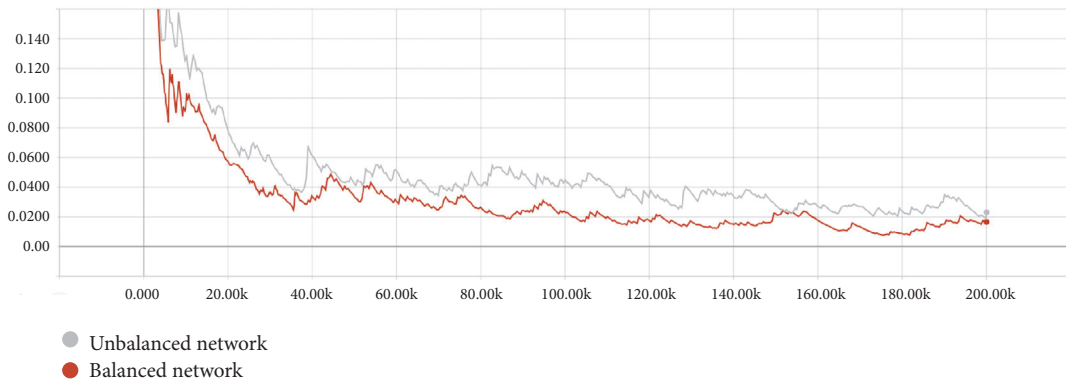


FIGURE 8: Balanced and unbalanced network.

After hours of training with the balanced network, we got a better result than with what had been the best detection network until now (see Figure 8). The results shown in Table 7 mean that the network trains well with these training coefficients and it even improves the results with a balanced network type (although in a real environment it is difficult to find it).

**3.3. Results.** In order to check the efficiency of the best neural network obtained in the previous section, different random frames have been extracted from a video showing different scenarios where pateras and people are seen from a real drone (see Figure 9). It should be noted that all these frames have never been previously seen by the neural network (not even in the testing stage), but are completely new to the neural



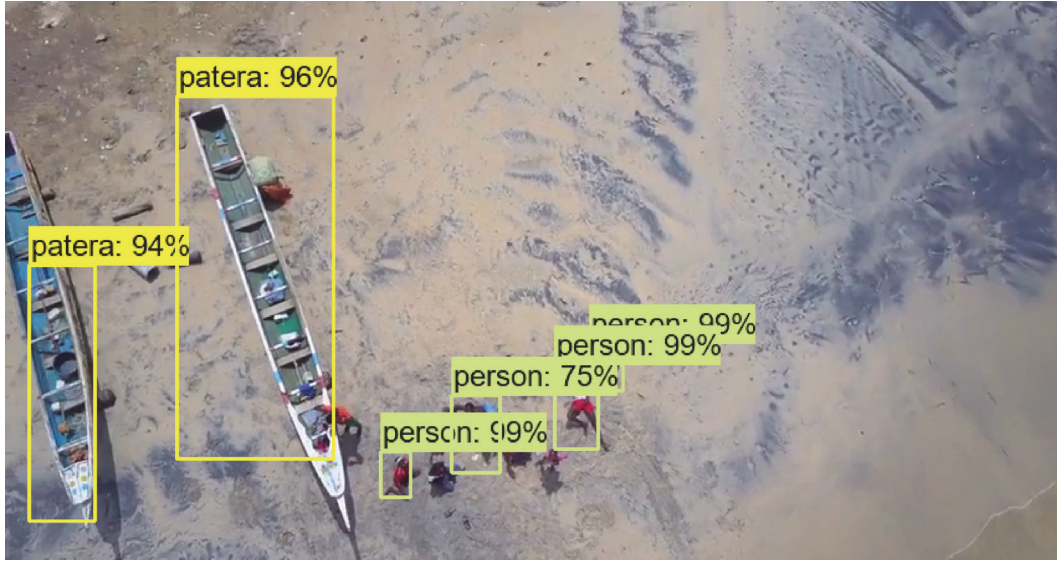


FIGURE 9: Image detection test from validation dataset.

TABLE 7: CL for Network 2-04 compared to the new balanced network.

Network	Classification Loss
Unbalanced Network 2-04	0.02310
Balanced network	0.01658

network. This set of frames is known as validation dataset. On the one hand, as a main result, the proposal produces a correct classification of boats and pateras between 94 and 96 % (although these ratios can vary from 92 to 99 % depending on the frame). On the other hand, a correct classification index for people has been obtained, which is around 98-99 %, although, in certain frames (a video has thousands of frames), this ratio can drop to 73 % due to interference with other objects in the video and the environment.

From the obtained results, we conclude that the defined procedure based on the Faster R-CNN proposed for training can be successfully used to detect boats, people and pateras.

#### 4. Security

In a system, like the defined above whose the results can be the difference between saving a human being saving or not it is essential to have the appropriate mechanisms to ensure that the information is not modified or accessed by illegitimate parties. It is for this reason, a study of possible attack vectors related to neural networks for image detection and problems in wireless communications has been performed, paying special attention in adversarial and Man in the Middle attacks.

**4.1. Adversarial Attacks.** Neural networks are one of the most powerful technological algorithms in the field of artificial intelligence. Among the various networks we can find some

specifically oriented to image detection (as seen throughout this work). Sometimes, the simple behaviour of a network fed with inputs (pixel's images) where the output is a type of classification can lead to error, so that it can be inferred that the network does not act correctly. An adversarial attack [30] is a type of attack within the rising field of artificial intelligence consisting in introducing an imperceptible perturbation that leads to an increased probability of taking the worst possible action.

In the case analysed in this work, this attack involves using a type of images that can be supplied to the network that though represent a certain type of object (for example a ship), for the network they mean something else (like a dog, a toaster.).

In environments where there are thousands or millions of types of classes and classifications it could be a problem. That is the case, for example of Google's Inception V3 [31], could be used to alter the driving of an autonomous vehicle that uses this type of network for altering the images of its environment by applying stickers [16] on traffic signs for the purpose of changing the maximum speed in a road.

The way in which this type of attacks act is through the excitation of the neural network inputs through the inclusion of new figures or noise (generally not perceptible to the human eye) making modifications in the input image (with gradient descent and back propagation techniques) making the network suffer something similar to an optical illusion.

The answer to the question of what this type of attack is looking for is how to maximise the error that can be achieved by entering erroneous information. That is to say, to do the opposite that the neural network expects to do this to minimise the error with the input parameters, all this, taking into account the fact that a formula must be applied to minimise the difference between the added disturbance and the original image with respect to the human eye.

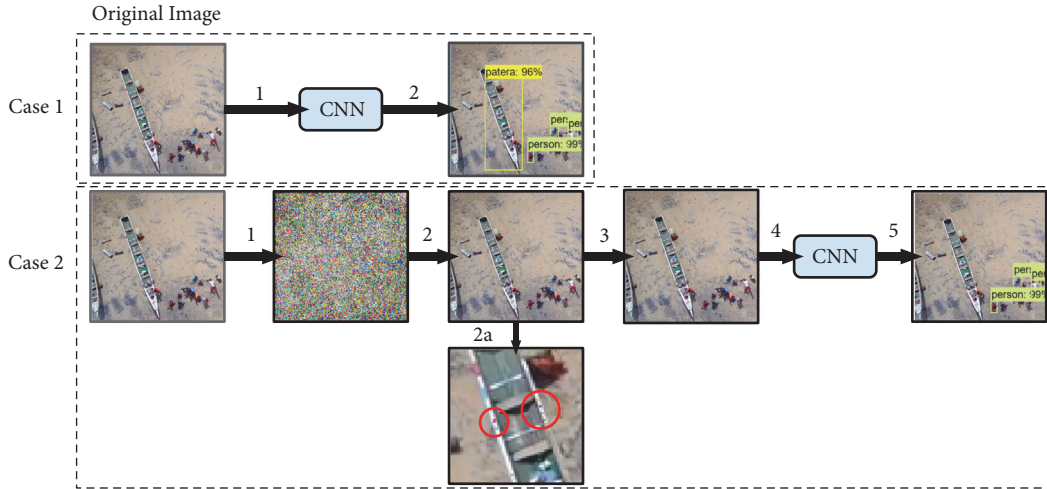


FIGURE 10: Adversarial attack proof of concept.

In the neural network that has been presented in this work, the number of classes has been limited to classify a total of 3 types of objects (ships, pateras, and people) so the margin of error within the possible classification could mean a sort of security system against this type of attack. Because of this, it can be said that, in a controlled environment, this type of attack would have no effect on the proposed system.

However, as a proof of concept, an adversary attack has been created that could modify the behaviour of our network. To do this, we have taken a random frame from a video sequence where we can see a whole patera, a part of another one and people (who could be castaways) in the sand. In Figure 10 it is possible to appreciate two main cases:

#### Case 1.

- (1) Starting from the frame extracted from the video, it has been processed directly by our neural network.
- (2) As a result, we have been obtained a detection of the patera with an index of 0.96 and of the people with an index variant between 0.98 and 0.99.

#### Case 2.

- (1) Based on the same starting image seen in Case 1, training has been carried out with a different neural network to the original one. With this, we demonstrate that adversarial attacks also fulfill a transition property that can affect other networks. The result of this step is the generation of an image with noise. The noise shown in the image has been modified by enlarging the brightness of the image in 10 steps because the original was a black image with little visible noise.
- (2) By applying the original noise to the initial image, a new resulting image is obtained that, with the naked eye, as can be seen in the image 2a of Figure 10, it has some pixels different from the original image.

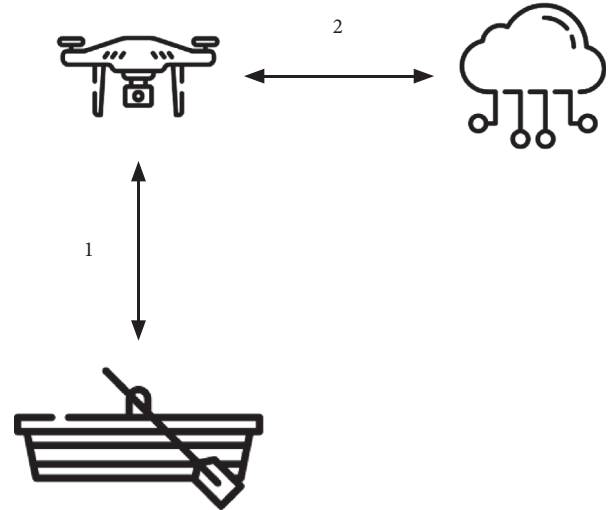


FIGURE 11: Attacking vectors.

- (3) To soften the effect appreciated in point 2, a series of mathematical operations are applied to each pixel to soften the textures and obtain a finished image.
- (4) The image generated in step 3 is sent to the neuronal network for the detection of pateras.
- (5) Finally it can be seen that by applying this new image, which at first sight is the same as the original, the system does not detect the patera.

**4.2. Attack Vectors.** In a possible scenario where an attacker wants to bypass the security measures that have been implemented, he/she could follow one of the following two ways (see Figure 11).

- (1) As discussed in the previous section, there is a type of attack called an adversarial attack that is designed

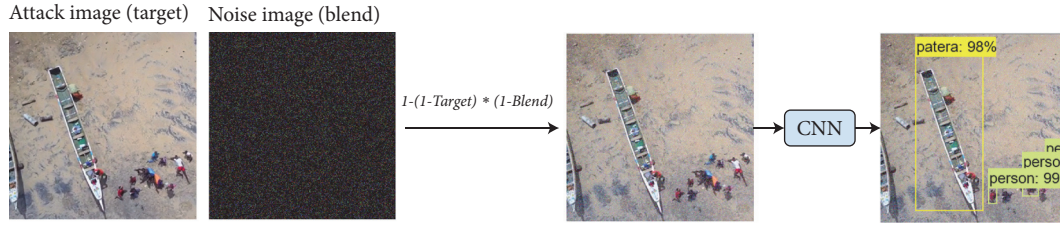


FIGURE 12: Random Gaussian noise blending.

to confuse the neural network. The aforementioned technique that has been used in a real environment of the inclusion of stickers [16] could be applied to the pateras in order to avoid the drone control by pretending the patera look like an unrecognised object or other object. Among the possible countermeasures to mitigate the attack, we have

- (i) JPEG compression method: This method is based on the hypothesis that the input image (i.e., the one taken by the drone) can be manipulated by the aforementioned attack so that the generated image has a noise that confuses the network. For the removal of this malicious noise it is possible to go for an 85% compression using a JPEG compression format [17] that will make the embedded noise blur, while maintaining the basic characteristics of shape in the image.
  - (ii) Noise Inclusion: The drone could have a simple internal image manipulation system to apply a Gaussian random noise so that the noise is imperceptible in the image before being sent to a server for processing. To do this we use an image of noise previously generated (or created in the moment) and then apply the formula of the blending method known as “screen” described with the formula:  $1 - (1 - \text{Target}) * (1 - \text{Blend})$ . The advantage of this compared to the method described above is that the loss of image quality is not affected (depending on the weight and size of the noise). However, it could include a slightly visible noise (see Figure 12).
- (2) Man in the Middle attack (MITM) is a sort of attack where an attacker is placed between sender and receiver. In this case the sender would be the drone and the receiver the server that will do the image processing through the neural network. The communication media can vary depending on the coverage in the area of emission. It is always a wireless connection like 2, 3, 4, or even 5G. In this case, the attacker can intercept the signal with the image in order to modify it on the fly including the necessary noise to make the image undetectable. To deal with such attacks, the system protects the security of the communication system through the cryptographic scheme described in the following section.

**4.3. Attribute-Based Encryption.** In the proposal described in this paper, an encryption is used to protect from unauthorised attackers the confidentiality of the database of the images captured from a smartphone on a UAV, which are labelled with the date when the image was taken, the GPS location of the photograph along with other selected metadata. Smartphones are less powerful than other systems in computations such as image transmission, key generation, and information storage and encryption. In order to reduce the overload of the security protocol, we propose the use of a light cryptographic technique. In addition, to offer the remote server the ability to securely examine all the images captured by UAVs in a region, an Attribute-Based Encryption is proposed. This is a type of public-key encryption in which private keys and encrypted texts depend on certain attributes, and decryption of encrypted text is only accessible to users with the satisfactory attribute configuration. In the proposal described in this document, the used attributes are related to date/time, geopositioning location, linked UAV, etc., so that the private key used in the remote server is restricted to be able to decipher encrypted texts whose attributes coincide with the policy of attributes linked to the UAVs it controls. This private key can be used to decrypt any encrypted text whose attributes match this policy but have no value in deciphering others. This means that each operator in a remote server has a set of UAVs assigned to him/her, so the images captured by any UAV cannot be decrypted either by an unauthorised attacker or by a server operator unrelated to that UAV. Since the used encryption is public-key encryption, its security is based on a mathematically hard problem, and security holds even if an attacker manages to corrupt the storage and obtain any encrypted text. The operations associated with the proposal involve the following phases.

- (1) Setup phase: this phase is where the algorithm takes the implicit security parameter to generate the Public Key (PuK) and Master Key (MaK).
- (2) KeyGen phase: in this phase, a trusted part generates a Transformation Key (TrK) and Private Key (PrK) linked to the smartphone, which are used to decrypt the information sent from it.
- (3) Encrypt Phase: in this phase, the smartphone encrypts the image using PuK and MaK before sending it to the remote server.
- (4) Transformation phase: this phase is where the remote server performs a partial decryption operation of the



encrypted data using TrK to transform the encrypted text into a simple encrypted text (partially decrypted) before sending it to the operators. If the operator's attributes satisfy the access structure associated with the encrypted text, he/she can use the decryption phase to retrieve the plaintext from the transformed ciphertext.

- (5) Decryption phase: as the transformation phase transforms the encrypted text into a simple encryption, finally, the server operator uses this phase to retrieve the plaintext of the transformed ciphertext, using the PrK.

## 5. Conclusions

In this work, a novel proposal has been defined to provide a solution to the problem of the detection of small boats, which are used many times by irregular immigration. For this purpose, a Convolutional Neuronal Network has been created, specifically trained for the detection of three types of objects: boats, people and pateras. This system is used in coordination with a UAV that sends the signals via wireless connection (LTE) to a server that will be responsible for processing the image in the neural network and detecting if it is an anomalous situation. This work describes and includes several security systems that allow us to guarantee the stability of the data so that they cannot be altered either before or after being sent. As a complement to protect data transmission systems using the ABE algorithm, a novel mechanism has been implemented to mitigate adversarial attacks by overlapping Gaussian noise to the possible attacking image noise. In addition, to discard false positives, a compendium of the GPS coordinates of the UAV is made with an AIS system of geolocalised ships. The main contribution is a light neural network with a high rate of detection of objects (reaching up to 99% accuracy), which would be a great help for Search And Rescue or border patrol teams in case of having to perform a rescue. A study with thousands of frames could be done to see the detection ratio and the accuracy of each object, to determine which object is better detected.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

Research was supported by the Spanish National Cybersecurity Institute (INCIBE) under the INCIBEC-2015-02492 and INCIBEI-2015-27338 grants and by the Spanish Ministry of Economy and Competitiveness, the FEDER Fund, and the CajaCanarias Foundation, under TEC2014-54110-R and DIG02-INSITU Projects. The financing granted to the ULL by

the Ministry of Economy, Industry, Commerce and Knowledge, co-financed by the European Social Fund by 85%, is gratefully acknowledged.

## References

- [1] H. van Houtum, "Human blacklisting: The global apartheid of the EU's external border regime," *Environment and Planning D: Society and Space*, vol. 28, no. 6, pp. 957–976, 2010.
- [2] L. Vives, "Unwanted sea migrants across the EU border: The Canary Islands," *Political Geography*, vol. 61, pp. 181–192, 2017.
- [3] P. Andreas, "Redrawing the Line: Borders and Security in the Twenty-first Century," *International Security*, vol. 28, no. 2, pp. 78–111, 2003.
- [4] F. J. de Lucas Martn, "Muertes en el mediterráneo: inmigrantes y refugiados, de infrasujetos de derecho a amenazas para la seguridad," *Quaderns de la Mediterrània= Cuadernos del Mediterráneo*, vol. 22, pp. 272–277, 2015.
- [5] Frontex, *Frontex - European Border And Coast Guard Agency*, 2018, <https://frontex.europa.eu/>.
- [6] UNHCR, *The Un Refugee Agency*, 2018, <http://www.unhcr.org/>.
- [7] P. Soddu, *Ceuta and melilla. security, human rights and frontier control*, Institut Europeu de la Mediterrània (eds) IEMED Mediterranean Yearbook Med, pp. 212–214, 2006.
- [8] M. Díaz-Cabrera, J. Cabrera-Gámez, R. Aguasca-Colomo, and K. Miatliuk, "Photogrammetric analysis of images acquired by an uav," *International Conference on Computer Aided Systems Theory*, pp. 109–116, Springer, 2013.
- [9] A. M. Klimkowska and I. Lee, "A preliminary study of ship detection from UAV images based on color space conversion and image segmentation," in *Proceedings of the 4th ISPRS International Conference on Unmanned Aerial Vehicles in Geomatics, UAV-g 2017*, pp. 189–193, Germany, September 2017.
- [10] T. Giitsidis, E. G. Karakasis, A. Gasteratos, and G. C. Sirakoulis, "Human and fire detection from high altitude UAV images," in *Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2015*, pp. 309–315, Finland, March 2015.
- [11] S. Freitas, C. Almeida, H. Silva, J. Almeida, and E. Silva, "Supervised classification for hyperspectral imaging in UAV maritime target detection," in *Proceedings of the 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 84–90, Torres Vedras, April 2018.
- [12] W. Huo, Y. Huang, J. Pei, Q. Zhang, Q. Gu, and J. Yang, "Ship Detection from Ocean SAR Image Based on Local Contrast Variance Weighted Information Entropy," *Sensors*, vol. 18, no. 4, p. 1196, 2018.
- [13] V. San Juan, M. Santos, and J. M. Andújar, "Intelligent UAV Map Generation and Discrete Path Planning for Search and Rescue Operations," *Complexity*, vol. 2018, Article ID 6879419, 17 pages, 2018.
- [14] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, and A. Fúster-Sabater, "Software implementation of the SNOW 3G generator on iOS and Android platforms," *Logic Journal of the IGPL. Interest Group in Pure and Applied Logics*, vol. 24, no. 1, pp. 29–41, 2016.
- [15] I. Santos-González, A. Rivero-García, P. Caballero-Gil, and C. Hernández-Goya, "Alternative communication system for emergency situations," in *Proceedings of the 10th International Conference on Web Information Systems and Technologies, WEBIST 2014*, pp. 397–402, Spain, April 2014.

- [16] K. Eykholt, I. Evtimov, E. Fernandes et al., “Robust physical-world attacks on deep learning visual classification,” in *Proceedings of the in IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.
- [17] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, 1992.
- [18] K. V. C. Ganesan, “Healthcare monitoring solution with decryption outsourcing by parallel computing in cloud,” *Innovative Research in Computer and Communication Engineering*, vol. 2, pp. 56–64, 2014.
- [19] S. Yu, *Data sharing on untrusted storage with attribute-based encryption [Ph.D. thesis]*, Worcester Polytechnic Institute, 2010.
- [20] J. Huang, V. Rathod, C. Sun et al., “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 3296–3305, USA, July 2017.
- [21] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.
- [22] K. E. A. Van De Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, “Segmentation as selective search for object recognition,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1879–1886, November 2011.
- [23] R. Girshick, “Fast R-CNN,” in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, December 2015.
- [24] V. Pareto, *Cours d'économie politique*, Librairie Droz, Second edition, 1964.
- [25] M. Abadi, P. Barham, J. Chen et al., “Tensorflow: a system for large-scale machine learning,” in *Proceedings of the in Symposium on Operating Systems Design and Implementation*, vol. 16, pp. 265–283, 2016.
- [26] A. Esteva, B. Kuprel, R. A. Novoa et al., “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [27] C. S. Chin, J. T. Si, A. S. Clare, and M. Ma, “Intelligent Image Recognition System for Marine Fouling Using Softmax Transfer Learning and Deep Convolutional Neural Networks,” *Complexity*, vol. 2017, Article ID 5730419, 9 pages, 2017.
- [28] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018.
- [29] L. Mathews and G. Steri, “Learning from imbalanced data,” in *Encyclopedia of Information Science and Technology*, pp. 1825–1834, IGI Global, Fourth edition, 2018.
- [30] T. B. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer, and D. Mané, “Adversarial patch,” *Computer Vision and Pattern Recognition*, 2017.
- [31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2818–2826, July 2016.



## Research Article

# Fuzzy Linguistic Protoforms to Summarize Heart Rate Streams of Patients with Ischemic Heart Disease

**María Dolores Peláez-Aguilera,<sup>1</sup> Macarena Espinilla ,<sup>2</sup> María Rosa Fernández Olmo,<sup>3</sup> and Javier Medina<sup>2</sup>**

<sup>1</sup>*Council of Health for the Andalusian Health Service, Av. de la Constitución 18, 41071 Sevilla, Spain*

<sup>2</sup>*University of Jaén, Department of Computer Science, Campus Las Lagunillas, 23071 Jaén, Spain*

<sup>3</sup>*Cardiac Rehabilitation Unit of the Hospital Complex of Jaén, Av. del Ejército Español 10, 23007 Jaén, Spain*

Correspondence should be addressed to Macarena Espinilla; [mestevez@ujaen.es](mailto:mestevez@ujaen.es)

Received 19 October 2018; Accepted 2 December 2018; Published 1 January 2019

Guest Editor: Higinio Mora

Copyright © 2019 María Dolores Peláez-Aguilera et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cardiac rehabilitation is a key program which significantly decreases mortality rates in high-risk patients with ischemic heart disease. Due to the huge lack of accessibility to such programs at Health Centers, outdoor-based programs for cardiac rehabilitation have been proposed as an excellent tool to improve accessibility for patients at Health Centers. These outdoor-based programs make use of wrist-worn devices for real-time monitoring of rehabilitation sessions based on clinical guidelines. In this way, a greater number of patients can fortunately gain access to the rehabilitation program. However, this advantage also means that the cardiac rehabilitation team has to monitor a greater number of sessions due to the increase of the number of benefited patients, so the team members spend a lot of time analyzing each patient's sessions. In this paper, we present a methodology to evaluate heart rate streams of patients with ischemic heart disease using a linguistic approach. This innovative methodology manages relevant linguistic descriptions (protoforms) for the cardiac rehabilitation team to identify sessions with interest indicators by means of linguistic summaries. Therefore, the analysis process is automated in a comprehensible way, offering short linguistic descriptions to the cardiac rehabilitation team, who promptly provide feedback to their patients. In order to show the great efficiency and effectiveness of the proposed methodology, we have used and applied this methodology to real data provided by patients of an outdoor cardiac rehabilitation program run by the Health Council of the Andalusian Health Service (Spain).

## 1. Introduction

The Internet of Things (IoT) paradigm [1] is based on the idea of multiple devices located around the world working to acquire information and store it in order to subsequently process and analyze this data with the aim of providing intelligent services [2].

In this context, there are open challenges to extract rich information from the huge amount of data from sensor sources in large-scale deployment within the revolutionary paradigm of IoT. To alleviate this limitation, multiple areas in the research field have been involved, such as data filtering, data aggregation, semantic analysis, and information utilization [3].

On the other hand, linguistic descriptions of data generate natural language texts [4] that convey the most relevant

information contained, and sometimes hidden, in the data. Protoforms and fuzzy logic, which were proposed by Zadeh [5, 6] as a useful knowledge model for reasoning [7], summarization [8], and aggregation [9] of data under uncertainty, are modeled by fuzzy sets whose degree of truth to fuzzy sets is defined by membership functions.

Both IoT paradigm and fuzzy linguistic models have been successfully proposed for managing uncertainty and vagueness in an interpretable way, which is a key issue to obtain high performance and results [10]. So, the use of protoforms and fuzzy logic has provided brilliant results in IoT systems in multiple areas with sensor data streams, such as weather forecasting [11], predicting of demand for urgent care in smart cities [12], fever medication control [13], visual scenes [14], or monitoring of patients with preeclampsia in wearable devices [15]. So, the fuzzy logic has been demonstrated as

a useful tool to deal with the uncertainty in the complex Internet of Things systems. In the green multimodal routing problem to improve the reliability of the routes, the fuzzy logic was proposed to model the uncertainty in a piecewise linear function to represent the road traffic congestion [16]. The approaches for activity recognition based on fuzzy logic have provided excellent results in the optimization of the configuration of a heterogeneous architecture of sensors [17]. In the context of robot manipulators' systems with multiple sensors and actuators, a fuzzy control scheme with adaptation algorithms has been proposed to manage the uncertainty of the information [18]. To predict the available maximal data transfer rate of single-pair high-speed digital subscriber line connections from measured frequency dependent electrical parameters of wire pairs, an approach based on fuzzy logic has been proposed [19].

This paper falls in the research field of linguistic descriptions of data with protoforms and fuzzy logic applied to e-health solutions based on complex IoT systems, specifically, in cardiovascular diseases, which represent the main health problem in developed countries according to the World Health Organization (WHO) [20] and where fuzzy logic has been shown to work as an effective modeling tool in cardiac rehabilitation [21, 22].

In the health field, secondary prevention programs and Cardiac Rehabilitation Units (CRU) have been developed in several countries [23, 24], having been proven as the most effective tool to improve prognosis. Cardiac rehabilitation (CR) is defined as the sum of activities required to influence the underlying cause of heart disease favorably, as well as to ensure the best physical, social, and mental condition of patients, enabling them to occupy a normal place in society by their own means [25].

In previous work [22], an outdoor cardiac rehabilitation program (CRP) for patients was embedded in a wrist-worn device with a heart rate sensor for personalized care. Outdoor CRPs have increased the accessibility of cardiac rehabilitation programs due to the fact that they overcome several limitations, such as lack of time, commodities, geographical area, and access to health services [24, 26]. In [22], an outdoor program was designed and supervised remotely by the cardiac rehabilitation team by means of a wearable mobile-cloud platform for collecting and synchronizing data between patients and the cardiac rehabilitation team. To do so, a linguistic approach based on fuzzy logic was proposed in order to model the cardiac rehabilitation protocol and the expert knowledge from the cardiac rehabilitation team.

A great impact under the outdoor CRP is that the number of benefited patients has drastically increased. This positive fact, however, means that the health team must monitor a greater number of sessions. In this way, the team members spend a lot of time analyzing the patients' sessions in a home-based CRP and they are overwhelmed with the huge amounts of information generated by each patient's wrist-worn device.

In order to solve this limitation, in this paper we present a methodology that generates textual information, summaries, from the heart rate streams of patients with ischemic heart disease by means of protoforms and fuzzy logic. So, this paper presents a methodology to summarize patients' rehabilitation

sessions, offering understandable information for the cardiac rehabilitation team. The key points of the proposed methodology are the following:

- (i) To allow the cardiac rehabilitation team to supervise a huge number of sessions and patients by means of linguistic summaries, which integrate an intuitive representation
- (ii) To model a proposed methodology where the linguistic summaries are focused on rich expressiveness, including linguistic temporal terms and linguistic quantifiers by means of linguistic aggregation operators
- (iii) To provide a flexible linguistic methodology where the cardiac rehabilitation team intuitively defines the key interest indicators using protoforms based on expert knowledge in order to recover and dynamically select the rehabilitation sessions that suit and match the expert criteria

The proposed methodology is applied to real data provided by several patients of a cardiac rehabilitation program run by the Health Council of the Andalusian Health Service (Spain) in order to show its efficiency and effectiveness.

The remainder of this paper is structured as follows: Section 2 presents the novel methodology to generate linguistic summaries of the rehabilitation sessions from the heart rate streams of patients with ischemic heart disease for the cardiac rehabilitation team. Section 3 presents a case study to show the utility and applicability of the proposed methodology with real data from the rehabilitation sessions of three patients freely participating in cardiac rehabilitation programs provided by the Health Council of the Andalusian Health Service in the Region of Jaén (Spain). Finally, some concluding remarks are pointed out in conjunction with future works.

## 2. Methodology

In this section, we describe a methodology to generate linguistic summaries of the rehabilitation sessions (RSs) from the heart rate streams (HRS) of patients with ischemic heart disease that follow an outdoor CRP. The HRS data are collected from real patients wearing a high-quality wrist-worn device, which has improved the quality of heart rate measurements and their health applications [27]. It is noteworthy that the linguistic modeling developed in this work has been defined by health experts in the CRP to summarize the sessions with interest indicators in cardiac rehabilitation.

For this purpose, we will present data processing of HRS from rehabilitation sessions through three stages. In the first stage, the raw data from heart rate streams is initially preprocessed using a previous approach [22]. In this stage, a fuzzy model is proposed to monitor the heart rate under a linguistic approach in real time by means of three representative terms and their membership functions, *low*, *adequate*, and *high*, as well as short-term fuzzy temporal windows (FTWs). The linguistic terms are computed in real time within the wrist-worn device in order to advise patients

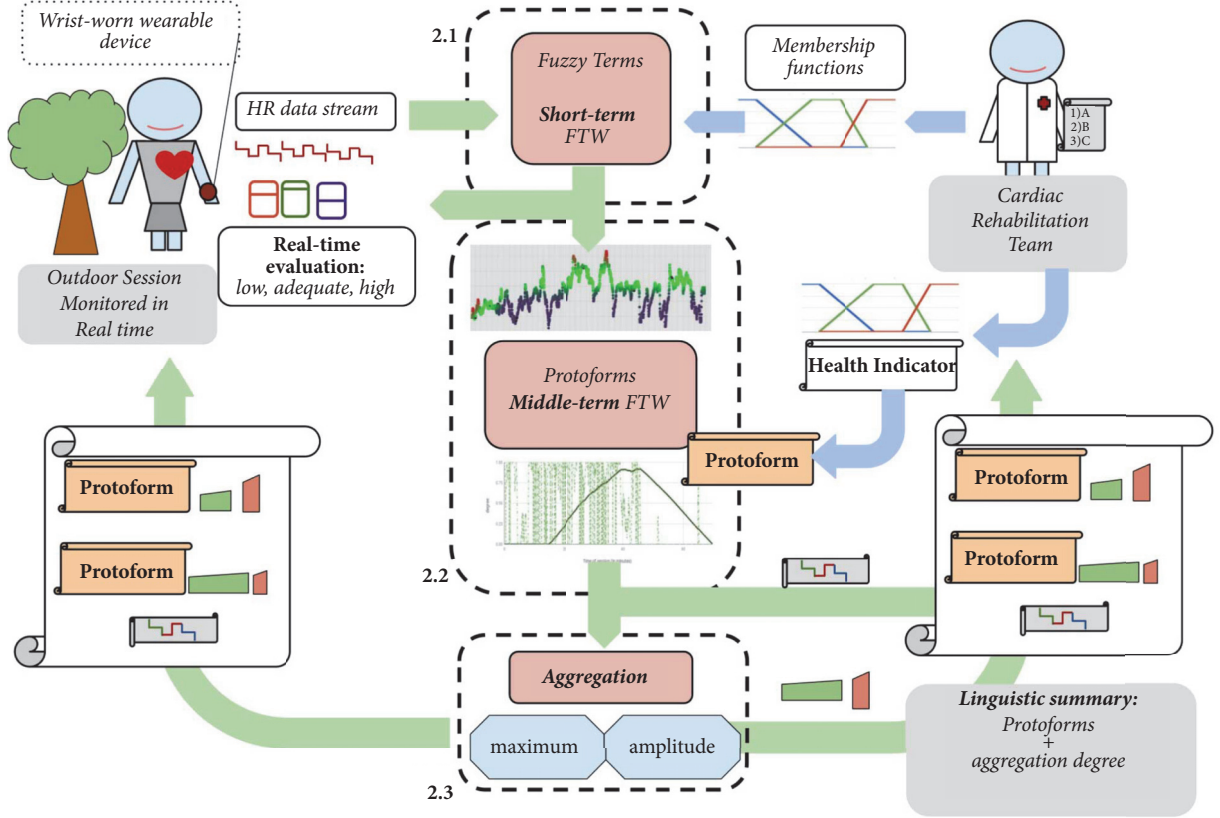


FIGURE 1: Architecture of the proposed methodology. First (2.1), the heart rate streams generated by wearable devices are computed to determine if the adherence to rehabilitation programs is adequate in real time. Second (2.2), linguistic summaries identify health indicators defined by experts using protoforms. Third (2.3), a final aggregation degree from the protoforms describes the rehabilitation sessions of the patients.

while they are undergoing the rehabilitation session. The main key points are briefly described in Section 2.1.

In this work, we present a methodology to compute linguistic summaries identifying key health indicators using expert knowledge from the linguistic terms computed in the first stage. Unlike the first stage, which is computed in real time while the patient is doing the exercise and without the complete information of the session, the summaries are computed for the cardiac rehabilitation team in a centralized way to evaluate a huge number of patients and sessions. For this purpose, we present a flexible model which selects and recovers the sessions according to expert criteria linguistically and intuitively.

So, in the second stage, we integrate an interpretable approach for the cardiac rehabilitation team that models knowledge linguistically. To do so, we use ad hoc protoforms. Protoforms are a general form of linguistic data summary [8].

The protoforms proposed in this methodology define linguistic summaries from the HRS of the sessions by means of long-term fuzzy temporal windows and fuzzy quantifiers, which provide rich expressiveness in the model.

In the third stage, we present the computing of a unique aggregation degree for the protoforms which describes the rehabilitation sessions summarizing the relevance and impact

of the protoforms, during the complete rehabilitation sessions. To do so, two semantics of aggregation operators are described: *maximum* and *amplitude*.

In Figure 1, we describe the three stages of the proposed methodology to summarize the rehabilitation sessions of patients by means of protoforms using a linguistic approach.

**2.1. Real-Time Monitoring of Heart Rate Streams.** In work [22], real-time monitoring of heart rate streams was proposed. First, three intuitive terms *low*, *adequate*, and *high* defined with three membership functions by means of fuzzy sets were proposed in order to describe the variable *heart rate* (HR).

The HR was measured by a pair value  $\bar{s}_i = \{s_i, t_i\}$ , where  $s_i$  represents a given value in the HRS and  $t_i$  is its time stamp. Hence, the HRS of a session is composed of a set of measured values  $S_{HRS} = \{\bar{s}_0, \dots, \bar{s}_i, \dots, \bar{s}_n\}$ , which are collected by the heart rate sensor of the wearable device.

Second, the fuzzification of HR,  $\overline{hr}_i$ , is determined by (i) Optimal Heart Rate Training Zones (OHRTZ) where the patient must develop the sessions, which is represented as a discrete HR range within  $[r_+^*, r_-^*]$ , and (ii) the Ventilatory Thresholds  $[VT_1, VT_2]$ , which represent the aerobic-anaerobic thresholds for the performance of an

efficient and safe physical activity. So, the terms *low*, *adequate*, and *high* of the variable HR  $s_i$  are described by a fuzzy set characterized by a membership function whose shape corresponds to the trapezoidal functions (TS, TR, and TL are described in Abbreviations) of

$$\begin{aligned}\mu_{adequate}(S_{HRS}) &= TS(S_{HRS})[VT_1, r_-^*, r_+^*, VT_2], \\ VT_1 &< r_-^* < r_+^* < VT_2 \\ \mu_{high}(S_{HRS}) &= TR(S_{HRS})[r_+^*, VT_2], \quad VT_2 > r_+^* \\ \mu_{low}(S_{HRS}) &= TL(S_{HRS})[VT_1, r_-^*], \quad VT_1 < r_-^*.\end{aligned}\quad (1)$$

In Figure 2, we show the representation of the fuzzy sets (membership functions) of HR for the three linguistic terms: *low*, *adequate*, and *high*. As noted in work [22], the thresholds of TS  $r_+^*(t_i)$ ,  $r_-^*(t_i)$ ,  $VT_1(t_i)$ ,  $VT_2(t_i)$  could progressively increase from basal state-defining time-dependent terms, but for the sake of simplicity here we write  $r_+^*$ ,  $r_-^*$ ,  $VT_1$ ,  $VT_2$ .

Third, a fuzzy temporal window [28, 29] to model the HRS was proposed in order to weight fuzzy linguistic terms based on fuzzy temporal linguistic terms and provide flexibility in the presence of eventual signal loss or variance in the sample rate. The FTWs are described straightforwardly according to the distance of the current time  $t_0$  to a given time stamp  $t_i$  as  $\Delta t_i = t_i - t_0$ . In this work, the use of FTWs is introduced also to describe temporal evaluation of long and middle terms in  $S_{HRS}$ .

Fourth, the degrees of the fuzzy linguistic terms  $V = \{V_{low}, V_{adequate}, V_{high}\}$  are weighted by the degree of their time stamps evaluated by the FTW  $T_k$ :

$$\begin{aligned}V_r \cap T_k(\bar{s}_i) &= V_r(s_i) \cap T_k(\Delta t_i) \in [0, 1] \\ V_r \cup T_k(S_{HRS}) &= \bigcup_{\bar{s}_i \in S_{HRS}} V_r \cap T_k(\bar{s}_i) \in [0, 1].\end{aligned}\quad (2)$$

A Fuzzy Weighted Average (FWA) [30] was proposed as an operation to model the t-norm and conorm:

$$\begin{aligned}V_r \cup T_k(S_{HRS}) &= \frac{1}{\sum_{t_i}^{S_{HRS}} T_k(\Delta t_i)} \sum_{t_i}^{S_{HRS}} T_k(\Delta t_i) V_r(s_i) \\ &\times T_k(\Delta t_i), \in [0, 1].\end{aligned}\quad (3)$$

Under evaluation, the experts defined and selected the most adequate size for the FTWs  $T_{low}$ ,  $T_{adequate}$ , and  $T_{high}$  in order to weight the terms  $V_{low}$ ,  $V_{adequate}$ , and  $V_{high}$  respectively. An embedded application in the wrist-worn device computes the degree of the terms *low*, *adequate*, and *high* in real time for each  $\bar{s}_i$  using

$$\begin{aligned}low(\bar{s}_i) &= V_{low} \cup T_{low}(\bar{s}_i) \\ adequate(\bar{s}_i) &= V_{adequate} \cup T_{adequate}(\bar{s}_i) \\ high(\bar{s}_i) &= V_{high} \cup T_{high}(\bar{s}_i).\end{aligned}\quad (4)$$

Finally, the degrees of the fuzzy linguistic terms  $low(\bar{s}_i)$ ,  $adequate(\bar{s}_i)$ , and  $high(\bar{s}_i)$  are computed within the wrist-band device: (i) showing a visually interpretable colored circle

which represents whether the current HR  $\bar{s}_i$  is *low*, *adequate*, or *high* during its FTWs and (ii) alerting the patient through sensor vibration in the wrist-band when  $\bar{s}_i$  is computed as  $high(adequate(\bar{s}_i) < high(\bar{s}_i))$  in its FTWs.

**2.2. Fuzzy Linguistic Summaries of Cardiac Rehabilitation Sessions.** In the previous section, we described the real-time evaluation of HR in a wrist-band application using a clinical-based protocol for monitoring and advising  $S_{HRS}$ . Here, we detail a methodology to generate linguistic summaries and identify key interest indicators using expert knowledge from the fuzzy linguistic terms computed in the wrist-worn devices, which describe the real-time adherence and performance of the patient in his/her HRS.

For this purpose, we start from the degree of the terms *adequate*, *low*, and *high* described in (4) for each  $\bar{s}_i$  within the data stream  $S_{HRS}$ . An example is shown in Figure 3, where a timeline with a real HRS is plotted using gradual colors *blue*, *green*, and *red* based on the degree of the fuzzy linguistic terms  $low(\bar{s}_i)$ ,  $adequate(\bar{s}_i)$ , and  $high(\bar{s}_i)$ , respectively.

**2.2.1. Protoforms for Describing Heart Rate Streams.** The aim of the proposed methodology is to generate linguistic summaries from the rehabilitation sessions of patients. For this purpose, in this second stage we process the fuzzy linguistic terms  $low(\bar{s}_i)$ ,  $adequate(\bar{s}_i)$ , and  $high(\bar{s}_i)$  from the data stream  $S_{HRS}$ , which are described in the previous section.

First, in order to integrate an interpretable and rich-expressive approach to model the expert knowledge linguistically, we introduce an ad hoc *protoform*  $P_o$  in the form of

$$P_o(\bar{s}_i) : (Q_k L_i T_j), \quad (5)$$

where

- (i)  $L_i$  defines a fuzzy linguistic term to evaluate the data stream. Here,  $L_i$  is straightforwardly related to fuzzy terms  $low(\bar{s}_i)$ ,  $adequate(\bar{s}_i)$ , and  $high(\bar{s}_i)$
- (ii)  $T_j$  defines a fuzzy temporal term where the term  $L_i$  is aggregated. The use of FTWs, which were introduced in the previous section, is extended to generate linguistic summaries of middle-long temporal terms from  $S_{HRS}$ . The aggregation of  $L_i$  over  $T_j$  for a given  $\bar{s}_i$  is computed by (4) as  $L_i \cup T_j(\bar{s}_i)$
- (iii)  $Q_k$  defines a fuzzy quantifier (FQ) to evaluate the impact and fulfillment of the linguistic term  $L_i$  within the temporal window  $T_j$  [28]. A FQ applies a transformation  $\mu_{Q_k} : [0, 1] \rightarrow [0, 1]$  to the aggregated temporal degree of  $\mu_{Q_k}(V_r \cup T_k(\bar{s}_i))$

The aim of modeling knowledge through protoforms is allowing the rehabilitation team to define key interest indicators using expert intuitive representations of temporal and quantification terms linguistically. An example of protoform is *most of the time* ( $Q_k$ ) *HR is adequate* ( $L_i$ ) *for around 40-60 minutes* ( $T_j$ ). We note that the protoforms are suitable to linguistically describe the impact and fulfillment of a fuzzy linguistic term in a fuzzy temporal window in more detail,



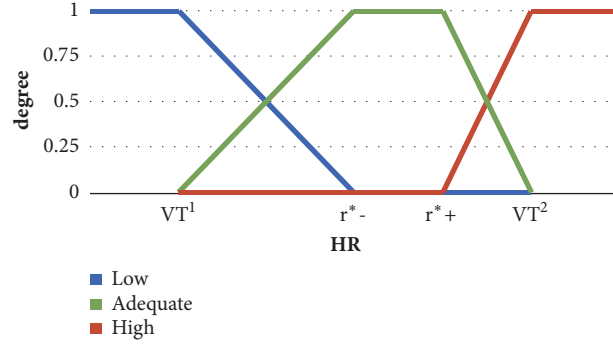


FIGURE 2: Representation with trapezoidal membership functions of the linguistic terms *low*, *adequate*, and *high* of the HR variable by means of Optimal Heart Rate Training Zones and Ventilatory Thresholds.

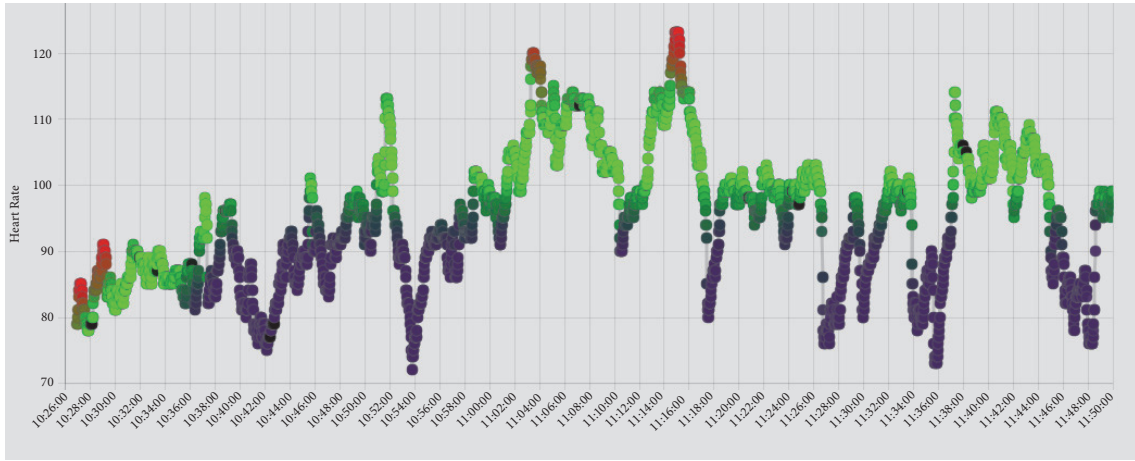


FIGURE 3: A timeline with a real HRS. The time and value configure the points in the chart  $\bar{s}_i = \{s_i, t_i\}$ , which are plotted using gradual colors *blue*, *green*, and *red* based on the degree of the terms  $low(\bar{s}_i)$ ,  $adequate(\bar{s}_i)$ , and  $high(\bar{s}_i)$ , respectively.

but subsequently they can be renamed to a shorter linguistic description, such as *adequate HR in session*.

In Figure 4, the protoform *most of the time HR is adequate for around 30-50 minutes* is shown, whose degree is represented with a real  $S_{HRS}$ . We also plot the degree of the protoform  $P_o(\bar{s}_i)$  in relation to the term  $adequate(\bar{s}_i)$  to describe the importance of the middle-long FTW and the FQ which enable the linguistic interpretability of the sentence *adequate heart rate in session* in the sensor stream. For the sake of simplicity, the shapes of the fuzzy membership functions of the example are detailed in Section 3.

Second, protoforms  $P_o(\bar{s}_i)$  can be combined using fuzzy logical operators to increase the linguistic capabilities of the model. So, we briefly introduce the following basic operations, which could be straightforwardly increased with advanced fuzzy operations in other contexts:

- (i) Fuzzy negation operator, which is represented as the complement  $\neg$  by the fuzzy function  $\neg P_o(\bar{s}_i) = 1 - P_o(\bar{s}_i)$
- (ii) Fuzzy union operator, which is represented by the t-norm  $P_o \wedge P_q(\bar{s}_i) = P_o(\bar{s}_i) \wedge P_q(\bar{s}_i)$ . The semantic function proposed for the fuzzy union operator is min:  $P_o \wedge P_q(\bar{s}_i) = \min\{P_o(\bar{s}_i), P_q(\bar{s}_i)\}$

- (iii) Fuzzy intersection operator, which is represented by the conorm  $P_o \vee P_q(\bar{s}_i) = P_o(\bar{s}_i) \vee P_q(\bar{s}_i)$ . The semantic function proposed for the fuzzy intersection operator is max:  $P_o \vee P_q(\bar{s}_i) = \max\{P_o(\bar{s}_i), P_q(\bar{s}_i)\}$

**2.2.2. Aggregation Operation for Protoforms.** As we detailed in Section 2.2.1, protoforms are defined to represent the health indicators from the rehabilitation sessions linguistically by means of expert knowledge. Although the evaluation of protoforms is properly computed  $P_o(\bar{s}_i)$  throughout the data stream  $S_{HRS}$ , an aggregation degree is proposed here in order to summarize the relevance of a protoform.

First, in (6) we describe the aggregation operation  $\cup(P_o)$  of the protoform  $P_o$ , which computes a single degree from the degree of protoform  $P_o(\bar{s}_i)$  over  $S_{HRS}$ , as

$$\cup(P_o) = \bigcup_{\bar{s}_i}^{S_{HRS}} P_o(\bar{s}_i) \in [0, 1]. \quad (6)$$

In order for the aggregated degree of the aggregation operator  $\cup(P_o)$  to keep its semantic integrity with the degree of the protoforms, we define two properties which the aggregation operation should assess:



**P\_o: Most of the time HR is adequate while around 40 and 60 minutes**

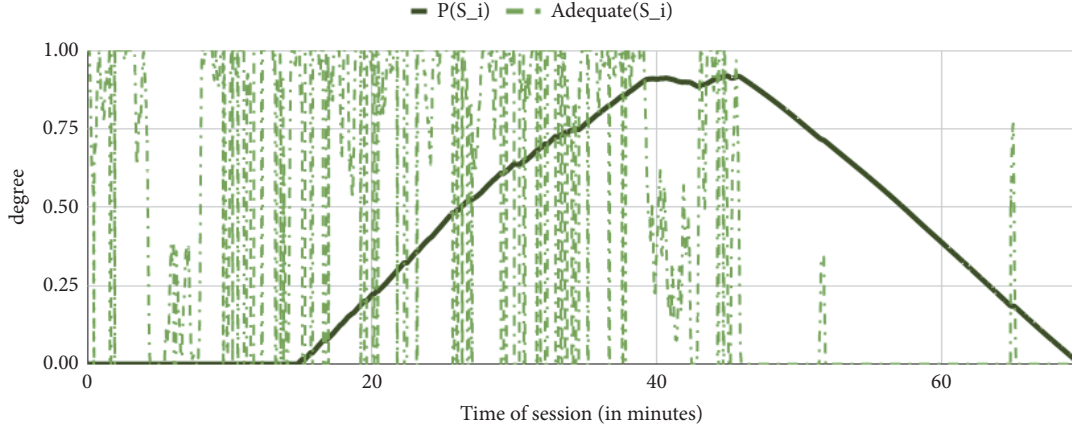


FIGURE 4: A timeline with the degrees of the term  $adequate(\bar{s}_i)$  (dotted line) and protoform *most of the time HR is adequate for around 30-50 minutes*  $P_o(\bar{s}_i)$  (thick line) on a real HRS. Based on its semantics, the degree of the protoform increases when the patient has maintained an adequate HR for 30 minutes.

- (i)  $\cup_0$ : zero-aggregation from zero stream. If the degree of the protoform is zero in the stream, the aggregation is zero:  $P_o(\bar{s}_i) = 0, \forall \bar{s}_i \in S_{HRS} \rightarrow \cup(P_o) = 0$ . It is a necessary condition of boundary property [31]
- (ii)  $\cup_+$ : positive-aggregation from nonzero stream. If there is a nonzero degree of the protoform in the stream, the aggregation is nonzero:  $\exists P_o(\bar{s}_i) > 0, \bar{s}_i \in S_{HRS} \rightarrow \cup(P_o) \neq 0 \rightarrow \cup(P_o) > 0$

We note the two properties have been determined based on the semantics of the protoforms to describe the HRS, which guarantee that the number of instances which match to a given protoform  $\cup(P_o) > 0$  is equal although the semantics of the aggregation varies. Specifically, the aggregation degree is zero if and only if the protoform is not representative in the session  $\cup(P_o) = 0 \iff P_o(\bar{s}_i) = 0, \forall \bar{s}_i \in S_{HRS}$ ; in another case  $\cup(P_o) > 0$ . We note that, in different fields and other contexts, further properties in aggregation operators can be selected [32].

Second, we propose two semantics to aggregate the degree of the protoform  $P_o(\bar{s}_i)$ : *maximum* and *amplitude*.

- (i) On the first hand, the *maximum* aggregation operator  $\max(P_o)$  computes the maximal value of degree of the protoform, which is shown in

$$\max(P_o) = \max\{P_o(\bar{s}_0), \dots, P_o(\bar{s}_i)\} \in [0, 1]. \quad (7)$$

The semantics of  $\max(P_o)$  describes the maximal truth degree of the protoform providing an intuitive representation as aggregation, which has been widely used in fuzzy logic as aggregation of rules within Mamdani-type inference models [33]. It fulfills

TABLE 1: Membership functions for the terms adequate, low, and high  $\mu_V$  and their respective FTWs  $\mu_T$ .

Term	$\mu_V$	$\mu_T$
adequate	$TS(s_i)[VT_1, r_-, r_+, VT_2]$	$TL(\Delta t_i)[3s, 5s]$
low	$TL(s_i)[VT_1, r_-]$	$TL(\Delta t_i)[3s, 5s]$
high	$TR(s_i)[r_+, VT_2]$	$TL(\Delta t_i)[0s, 1s]$

the properties of zero-aggregation  $\cup_0$  and positive-aggregation  $\cup_+$ .

$$\cup_0 : P_o(\bar{s}_i) = 0,$$

$$\forall \bar{s}_i \in S_{HRS} \equiv \cup(P_o) = \max\{0, \dots, 0\} = 0$$

$$\cup_+ : \exists P_o(\bar{s}_i)^+ > 0,$$

$$\bar{s}_i \in S_{HRS} \equiv \cup(P_o) = \max\{0, P_o(\bar{s}_i)^+, \dots, 0\} > 0$$





- (ii) On the other hand, the *amplitude* aggregation operator  $|P_o|$  describes the persistence and presence of the protoform degree  $P_o(\bar{s}_i)$  throughout the rehabilitation session. For this purpose, the fuzzy quantification of the weight of the protoform  $W(P_o)$  degree within the HRS is proposed in

$$W(P_o) = \frac{\sum_{\bar{s}_i \in S_{HRS}} P_o(\bar{s}_i)}{|S_{HRS}|} \quad (9)$$

$$|P_o| = Q(W(P_o)) \in [0, 1].$$

First, the weight of the protoform degree  $W(P_o)$ , which represents a suitable measure as fuzzy aggregation [34], is computed as the relation between the

TABLE 2: Textual description in natural language: short linguistic description and related protoforms.

Icon	Id	short linguistic descriptions	Protoforms: $Q_k L_i T_j$
	$P_1$	Adequate HR in session	At least half of the time the HR is adequate for around 25-50 minutes
	$P_2$	High HR in session is worrying	Most of the time the HR is high for around 1-3 minutes
	$P_3$	Low HR intensity in session	Most of the time the HR is low for around 15-25 minutes
	$P_4$	Unstable HR progression in session	While a part of the time the HR is high in the last 2 minutes $\cap$ While a part of the time the HR was low 1-3 minutes ago

sum of the degrees  $\sum_{\bar{s}_i}^{S_{HRS}} \bar{s}_i$  regarding the norm (or size) of the complete HRS  $|S_{HRS}|$ . Second, a fuzzy quantification is provided by a FQ which transforms  $W(P_o)$  into  $|(P_o)| = Q(W(P_o))$  through the fuzzy membership function  $\mu_Q : [0, 1] \rightarrow [0, 1]$ . For the sake of simplicity, we refer to  $\mu_Q$  as  $Q$ .

To guarantee that  $|(P_o)|$  fulfills the properties of zero-aggregation and positive-aggregation, the membership function  $Q$  of the FQ should assess the following properties:  $Q$  is a monotone function  $x \leq y \rightarrow Q(x) \leq Q(y)$ ,  $Q(0) = 0$ , and  $\lim_{x \rightarrow 0^+} Q(x) > 0$ :

$$\begin{aligned}
\cup_0 : \quad & Q(0) = 0, \\
& P_o(\bar{s}_i) = 0, \\
& \forall \bar{s}_i \in S_{HRS} \equiv W(P_o) = 0 \rightarrow \\
& Q(W(P_o)) = 0 \\
\cup_+ : \quad & \lim_{x \rightarrow 0^+} Q(x) > 0, \\
& x \leq y \rightarrow \\
& Q(x) \leq Q(y), \\
& \exists P_o(\bar{s}_i) > 0, \\
& \bar{s}_i \in S_{HRS} \equiv \sum_{\bar{s}_i}^{S_{HRS}} \bar{s}_i > 0 \rightarrow \\
& W(P_o) > 0 \rightarrow \\
& Q(W(P_o)) > 0
\end{aligned} \tag{10}$$

### 3. Case Study

In this section, we present a case study which illustrates the proposed methodology. The data of the rehabilitation sessions correspond to three patients who freely participated in the project *Monitoring of Patients with Ischemic Heart Disease within Outdoor Cardiac Rehabilitation Programs* of the Council of Health for the Andalusian Health Service in the Region of Jaén (Spain). They had a wrist-worn heart rate device

(Polar M600) (<https://www.polar.com/es/productos/sport/M600-GPS-smartwatch> (accessed on 10/14/2018)), which collected the heart rate data during the rehabilitation sessions using a wearable application.

141 rehabilitation sessions were collected (48, 55, and 38, respectively, for the three patients) from April to August 2018. The duration of the sessions was defined and adapted to patient evolution by the cardiac rehabilitation team, varying from 30 minutes to 80 minutes. A total of 639.709 heart rate samples were collected.

**3.1. Real-Time Monitoring of Heart Rate Streams.** The application embedded in the wrist-worn heart rate device collected the heart rate of patients and advised patients during their outdoor rehabilitation sessions. For this purpose, the application computes fuzzy linguistic terms *adequate*, *high*, and *low* using a short FTW. Their membership functions were obtained from the previous work [22] and are described in Table 1.

The values of OHRTZ  $[r_+^*, r_-^*]$  and Ventilatory Thresholds  $[VT_1, VT_2]$  were adapted for each patient based on an initial controlled stress test at the Health Center.

**3.2. Protoforms for Describing Heart Rate Streams.** The protoforms enable the rehabilitation team to define key health indicators linguistically using expert knowledge. In Table 2, some examples of the protoforms defined by the cardiac rehabilitation team are described.

Next, the membership functions of the FTWs and FQs were straightforwardly defined by both the computer science team and cardiac rehabilitation team of the project. They have been defined by different shapes of trapezoidal membership functions, whose values are shown in Table 3.

In Figure 5, we show the computing of the protoform  $P_o$  in real rehabilitation sessions, including the degree of the term  $L_i$ . We note that the protoforms are also useful to determine the region of interests over  $S_{HRS}$  indicating the ranges where the truth degree is activated  $P_o(\bar{s}_i) > 0$ .

**3.3. Aggregation Operation for Protoforms.** In this section, we describe the results of the aggregation of the protoform in the HRS of patients, which determine a single and descriptive

TABLE 3: Trapezoidal membership functions for FTWs and FQ of protoforms.

Textual description in natural language	Type	$\mu_T / \mu_Q$
For around 25-50 minutes	$T_j$	$TL(\Delta t_i)[25m, 50m]$
For around 15-25 minutes	$T_j$	$TL(\Delta t_i)[15m, 25m]$
In the last 2 minutes	$T_j$	$TL(\Delta t_i)[1m, 2m]$
Around 1-3 minutes ago	$T_j$	$TS(\Delta t_i)[0m, 1m, 2m, 3m]$
At least half of the time	$Q_k$	$TR((V_r \cup T_k(\bar{s}_i))[0.25, 0.75])$
While a part of the time	$Q_k$	$TR((V_r \cup T_k(\bar{s}_i))[0.25, 0.5])$
Most of the time	$Q_k$	$TR((V_r \cup T_k(\bar{s}_i))[0.5, 1])$

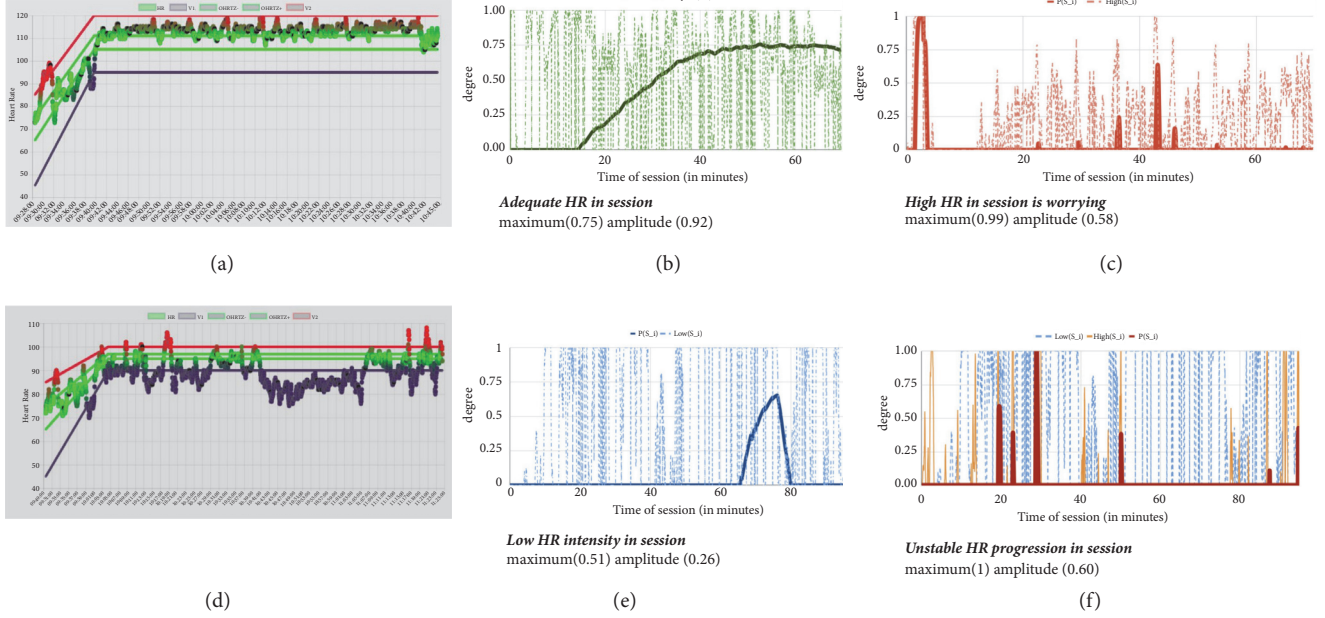


FIGURE 5: Example of the protoforms in real rehabilitation sessions. (a) A session with predominance of adequate adherence and some high rates. (b) The degree of the term *adequate* is represented by the dotted line; the degree of the protoform *at least half of the time the HR is adequate for around 25-50 minutes* is represented by the thick line. The aggregated degrees *maximum* and *amplitude* are shown in the bottom. (c) The degree of the term *high* is represented by the dotted line; the degree of the protoform *most of the time the HR is high for around 1-3 minutes* is represented by the thick line. The aggregated degrees *maximum* and *amplitude* are shown in the bottom. (d) A session with predominance of low adherence and some high rates. (e) The degree of the term *low* is represented by the dotted line; the degree of the protoform *most of the time the HR is low for around 15-25 minutes* is represented by the thick line. The aggregated degrees *maximum* and *amplitude* are shown in the bottom. (f) The degrees of the terms *high* and *low* are represented by the dotted line in yellow and blue, respectively; the degree of the protoform *part of the time the HR is high in the last 2 minutes*  $\cap$  *part of the time the HR was low around 1-3 minutes ago* is represented by the thick line. The aggregated degrees *maximum* and *amplitude* are shown in the bottom.

degree from the protoform degree  $P_o(\bar{s}_i)$  over the heart rate stream  $S_{HRS}$ .

In this work, two semantics to aggregate the degree of the protoform are proposed: *maximum*  $\max(P_o)$  and *amplitude*  $|P_o|$ , which represent the maximal truth degree of the protoform and the presence of the protoform throughout the rehabilitation session, respectively.

The *maximum* aggregation operator is not parametric. Conversely, the *amplitude* aggregation operator was modeled using expert knowledge of the computer science team as well as the cardiac rehabilitation team, who determined the membership function of the fuzzy quantifiers,  $\mu_Q$ , for each protoform  $|P_o|$ , which are described in Table 4.

TABLE 4: Membership functions for fuzzy quantifiers of *amplitude* aggregation operator.

Textual description in natural language	$\mu_Q$
<i>adequate heart rate in session</i>	$TR(W(P_o))[0, 0.5]$
<i>the high rate is worrying</i>	$TR(W(P_o))[0, 0.05]$
<i>the session presents low intensity</i>	$TR(W(P_o))[0, 0.25]$
<i>the session has unstable rates</i>	$TR(W(P_o))[0, 0.01]$

We note that (i) the membership functions are in the shape of  $TR(x)[0, \alpha]$  to fulfill the properties of zero-aggregation and positive-aggregation described in

TABLE 5: Metrics in the aggregation of protoforms from rehabilitation sessions by patient (number of sessions  $N$  and percentage % from the RS  $|RS|$  total in parentheses).









					
	Total	$P_1$	$P_2$	$P_3$	$P_4$
Patient	$ RS $	$N(\%)$	$N(\%)$	$N(\%)$	$N(\%)$
1	49	2(4%)	24(48%)	49(100%)	9(18%)
2	56	36(64%)	44(78%)	48(85%)	26(46%)
3	39	36(92%)	35(90%)	4(10%)	7(18%)

TABLE 6: Metrics in the aggregation of protoforms from rehabilitation sessions (number of sessions and percentage from the total of RS in parentheses).

 $P_1$		 $P_2$		 $P_3$		 $P_4$		
$\alpha - cut$	max	amp	max	amp	max	amp	max	amp
$\alpha = 0$	74(51%)	74(51%)	103(72%)	103(72%)	101(70%)	101(70%)	42(29%)	42(29%)
$\alpha = 0.5$	46(32%)	51(35%)	51(35%)	81(56%)	72(48%)	87(51%)	15(10%)	29(20%)
$\alpha = 0.9$	32(22%)	32(23%)	40(27%)	65(45%)	63(43%)	60(42%)	13(9%)	20(14%)

Section 2.2.2 and (ii) the quantification membership functions of protoforms where a high rate is involved relate short weights to relevant amplitudes, due to the fact that high rates are very significant and worrying in  $S_{HRS}$ . An example of real sessions and the aggregation operators *maximum* and *amplitude* for the proposed protoforms is presented in Figure 5.

In Table 5, we summarize the number and percentage of RS which have been recovered for each protoform  $\cup(P_o) > 0$  and user. We note the descriptive summary which represents the aggregation in determining and differentiating the performance of the patients within the rehabilitation program.

We note, based on the properties described for the aggregation operators, that both *maximum*  $\max(P_o)$  and *amplitude*  $|P_o|$  recover same HRs.

Finally, as each aggregation operator determines a degree  $\cup(P_o) \in [0, 1]$  which can be filtered by a threshold  $\alpha$  using a straightforward  $\alpha - cut$  in order to recover more descriptive and relevant sessions which match the protoform  $P_o$ , this intuitive value  $\alpha$  can be also modified by experts when analyzing the summaries of the patients to filter and select the RSs. In Table 6, we present the number of RSs recovered by  $\alpha - cut$  for each protoform in function of the values of  $\alpha = \{0, 0.5, 0.9\}$ .

#### 4. Conclusions and Future Works

Complex IoT systems for e-health solutions allow us to reach a greater number of patients. However, this positive fact generates a vast amount of information that must be analyzed by the health team. This paper has been focused on the real-time monitoring of an outdoor cardiac rehabilitation program for patients with ischemic heart disease, which was designed and supervised remotely by the cardiac rehabilitation team. In this program, wearable wrist-worn devices

with heart rate sensors integrating a high-quality protocol based on clinical guidelines were used to monitor the heart rate of patients in a personalized way. A wearable mobile-cloud platform was defined for collecting and synchronizing data between patients and the cardiac rehabilitation team to provide feedback.

The main motivation behind this work has been to provide the cardiology rehabilitation team with linguistic summaries of the rehabilitation sessions based on the heart rate streams of patients. In order to address this challenge, a methodology has been proposed in this paper, which is based on the use of the linguistic descriptions of data with protoforms and fuzzy logic of the heart rate streams of patients in order to provide linguistic summaries with rich expressiveness of interest indicators. So, the proposed methodology models short descriptions such as *the high rate is worrying in the session* or *the session presents low intensity*.

The proposed methodology enables (i) a fast analysis process to monitor a higher number of benefited patients and (ii) identification of sessions with interest indicators for the cardiac rehabilitation team to provide feedback. To do so, on the one hand, linguistic temporal terms and linguistic quantifiers have been used on linguistic aggregation operators in the heart rate streams of patients. On the other hand, flexible linguistic modeling has been defined in the proposed methodology where the cardiac rehabilitation team intuitively defines the key interest indicators using protoforms by means of expert knowledge in order to recover and dynamically select the rehabilitation sessions which suit and match the expert criteria.

In future works, the methodology will be extended to automatically generate linguistic recommendations for the patients in further sessions by means of machine learning techniques and based on the knowledge of the cardiac rehabilitation team.



## Abbreviations

HR:	Heart rate
FQ:	Fuzzy quantifier
FTW:	Fuzzy temporal window
FWA:	Fuzzy Weighted Average
HRS:	Heart rate stream
OHRTZ:	Optimal Heart Rate Training Zones
RS:	Rehabilitation session
TS:	$TS(x)[l_1, l_2, l_3, l_4] = \{0, x \leq 0;$ $(x - l_1)/(l_2 - l_1), l_1 \leq x \leq l_2; 1, l_2 \leq x \leq l_3;$ $(l_4 - x)/(l_4 - l_3), l_3 \leq x \leq l_4; 0, l_4 \leq x\}$
TR:	$TR(x)[l_1, l_2] = \{1, x \leq l_1; (l_2 - x)/(l_2 - l_1),$ $l_1 \leq x \leq l_2; 0, l_2 \leq x\}$
TL:	$TL(x)[l_1, l_2] = \{0, x \leq l_1; (x - l_1)/(l_2 - l_1),$ $l_1 \leq x \leq l_2; 1, l_2 \leq x\}$ .

## Data Availability

The HR data from the rehabilitation sessions and the results presented in this work are available through the URL <http://serezade.ujaen.es:8054/redcore-summary-data/> (accessed on 10/25/2018).

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

María Dolores Peláez-Aguilera developed the methodology and contributed materials and analysis tools; Javier Medina designed the methodology and wrote the paper; Macarena Espinilla analyzed and improved the methodology and wrote the paper; María Rosa Fernández Olmo defined the rehabilitation indicators.

## Acknowledgments

This contribution has been supported by the Council of Health of the Andalusian Health Service, Spain, through project PI-0203-2016 together with the Spanish government through research project TIN2015-66524-P.

## References

- [1] E. Borgia, "The internet of things vision: key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014.
- [2] D. Gil, A. Ferrández, H. Mora-Mora, and J. Peral, "Internet of things: a review of surveys based on context aware intelligent services," *Sensors*, vol. 16, no. 7, article 1069, 2016.
- [3] I. Kholod, I. Petuhov, and M. Efimova, "Data Mining for the Internet of Things with Fog Nodes," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, vol. 9870 of *Lecture Notes in Computer Science*, pp. 25–36, Springer International Publishing, Cham, 2016.
- [4] J. Kacprzyk and S. Zadrozny, "Comprehensiveness of linguistic data summaries: A crucial role of protoforms," *Studies in Computational Intelligence*, vol. 445, pp. 207–221, 2013.
- [5] L. A. Zadeh, "Generalized theory of uncertainty (GTU)—principal concepts and ideas," *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 15–46, 2006.
- [6] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning I," *Information Sciences*, vol. 8, pp. 199–249, 1975.
- [7] L. A. Zadeh, "A prototype-centered approach to adding deduction capability to search engines - The concept of protoform," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS-FLINT 2002*, pp. 523–525, USA, June 2002.
- [8] J. Kacprzyk and S. Zadrozny, "Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools," *Information Sciences*, vol. 173, no. 4, pp. 281–304, 2005.
- [9] R. R. Yager, "On linguistic summaries of data," *On linguistic summaries of data*, pp. 347–363, 1991.
- [10] E. Kim, S. Helal, C. Nugent, and M. Beattie, "Analyzing activity recognition uncertainties in smart home environments," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 4, 2015.
- [11] A. Ramos-Soto, A. J. Bugarín, S. Barro, and J. Taboada, "Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 44–57, 2015.
- [12] J. Medina Quero, M. A. Lopez Medina, A. Salguero Hidalgo, and M. Espinilla, "Predicting the Urgency Demand of COPD Patients From Environmental Sensors Within Smart Cities With High-Environmental Sensitivity," *IEEE Access*, vol. 6, pp. 25081–25089, 2018.
- [13] J. Medina, M. Espinilla, Á. L. García-Fernández, and L. Martínez, "Intelligent multi-dose medication controller for fever: From wearable devices to remote dispensers," *Computers Electrical Engineering*, vol. 65, pp. 400–412, 2018.
- [14] A. Gatt, N. Marín, F. Portet, and D. Sánchez, "The role of graduality for referring expression generation in visual scenes," *Communications in Computer and Information Science*, vol. 610, pp. 191–203, 2016.
- [15] M. Espinilla, J. Medina, A.-L. García-Fernández, S. Campaña, and J. Londoño, "Fuzzy Intelligent System for Patients with Preeclampsia in Wearable Devices," *Mobile Information Systems*, vol. 2017, Article ID 7838464, 10 pages, 2017.
- [16] Y. Sun, M. Hrušovský, C. Zhang, and M. Lang, "A Time-dependent fuzzy programming approach for the green multi-modal routing problem with rail service capacity uncertainty and road traffic congestion," *Complexity*, vol. 2018, Article ID 8645793, 22 pages, 2018.
- [17] M. Espinilla, J. Medina, A. Calzada, J. Liu, L. Martínez, and C. Nugent, "Optimizing the configuration of a heterogeneous architecture of sensors for activity recognition, using the extended belief rule-based inference methodology," *Microprocessors and Microsystems*, vol. 52, pp. 381–390, 2017.
- [18] Y. Fan, K. Xing, and X. Jiang, "Fuzzy Adaptation Algorithms' Control for Robot Manipulators with Uncertainty Modelling Errors," *Complexity*, vol. 2018, Article ID 5468090, 8 pages, 2018.
- [19] F. Lilik, S. Nagy, and L. T. Kóczy, "Improved Method for Predicting the Performance of the Physical Links in Telecommunications Access Networks," *Complexity*, vol. 2018, Article ID 3685927, 14 pages, 2018.
- [20] World Health Organization, *Needs and Action Priorities in Cardiac Rehabilitation and Secondary Prevention in Patients*



with Coronary Heart Disease, WHO Regional Office for Europe, 1993.

- [21] C. C. Oliveira, R. Dias, and J. M. da Silva, "A Fuzzy Logic Approach for a Wearable Cardiovascular and Aortic Monitoring System. In," in *ICT Innovations 2015*, pp. 265–274, Springer, Cham, 2016.
- [22] J. Medina Quero, M. R. Fernández Olmo, M. D. Peláez Aguilera, and M. Espinilla Estévez, "Real-time monitoring in home-based cardiac rehabilitation using wrist-worn heart rate devices," *Sensors*, vol. 17, no. 12, p. 2892, 2017.
- [23] B. S. Heran, J. M. Chen, S. Ebrahim et al., "Exercise-based cardiac rehabilitation for coronary heart disease," *Cochrane Database of Systematic Reviews*, vol. 7, Article ID CD001800, 2011.
- [24] C. S. S. Yue, "Barriers to participation in a phase II cardiac rehabilitation programme," *Hong Kong Medical Journal*, vol. 11, no. 6, pp. 472–475, 2005.
- [25] G. J. Balady, P. A. Ades, V. A. Bittner et al., "Referral, enrollment, and delivery of cardiac rehabilitation/secondary prevention programs at clinical centers and beyond: A presidential advisory from the american heart association," *Circulation*, vol. 124, no. 25, pp. 2951–2960, 2011.
- [26] J. Daly, A. P. Sindone, D. R. Thompson, K. Hancock, E. Chang, and P. Davidson, "Barriers to participation in and adherence to cardiac rehabilitation programs: a critical literature review," *Progress in Cardiovascular Nursing*, vol. 17, no. 1, pp. 8–17, 2002.
- [27] J. F. Horton, P. Stergiou, T. S. Fung, and L. Katz, "Comparison of Polar M600 Optical Heart Rate and ECG Heart Rate during Exercise," *Medicine & Science in Sports & Exercise*, vol. 49, no. 12, pp. 2600–2607, 2017.
- [28] J. Medina, M. Espinilla, and C. Nugent, "Real-Time Fuzzy Linguistic Analysis of Anomalies from Medical Monitoring Devices on Data Streams," in *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pp. 300–303, Cancun, Mexico, May 2016.
- [29] J. Medina, L. Martínez, and M. Espinilla, "Subscribing to fuzzy temporal aggregation of heterogeneous sensor streams in real-time distributed environments," *International Journal of Communication Systems*, vol. 30, no. 5, p. e3238, 2017.
- [30] W. M. Dong and F. S. Wong, "Fuzzy weighted averages and implementation of the extension principle," *Fuzzy Sets and Systems*, vol. 21, no. 2, pp. 183–199, 1987.
- [31] R. Mesiar and M. Komornikova, "Aggregation operators," in *Proceeding of the XI. Conference on applied Mathematics PRIM*, D. Herceg and K. Surla, Eds., pp. 193–211, Univ. Novi Sad, Novi Sad, 1997.
- [32] M. Detyniecki, *Fundamentals on aggregation operators. PhD Thesis [Ph.D. thesis]*, University of California, Berkeley, 2001.
- [33] I. Iancu, *A Mamdani type fuzzy logic controller*, InTech, Theories and Applications, 2012.
- [34] S. Martinez-Municio, L. Rodriguez-Benítez, E. Castillo-Herrera, J. Giral-Muñoz, Jiménez-Linares., and L. Rodriguez-Benítez, *Modelado lingüístico y síntesis de series temporales heterogéneas de consumo energético*, Springer, Granada, 2018.

## Research Article

# Fully Flexible Parallel Merge Sort for Multicore Architectures

Zbigniew Marszałek, Marcin Woźniak , and Dawid Połap

*Institute of Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland*

Correspondence should be addressed to Marcin Woźniak; [marcin.wozniak@polsl.pl](mailto:marcin.wozniak@polsl.pl)

Received 10 June 2018; Revised 30 August 2018; Accepted 23 September 2018; Published 2 December 2018

Guest Editor: Julian Szymanski

Copyright © 2018 Zbigniew Marszałek et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The development in multicore architectures gives a new line of processors that can flexibly distribute tasks between their logical cores. These need flexible models of efficient algorithms, both fast and stable. A new line of efficient sorting algorithms can support these systems to efficiently use all available resources. Processes and calculations shall be flexibly distributed between cores to make the performance as high as possible. In this article we present a fully flexible sorting method designed for parallel processing. The idea we describe in this article is based on modified merge sort, which in parallel form is designed for multicore architectures. The novelty of this idea is in particular way of processing. We have developed a fully flexible method that can be implemented for a number of processors. The tasks are flexibly distributed between logical cores to increase the efficiency of sorting. The method preserves separation of concerns; therefore, each of the processors works separately without any cross actions and interruptions. The proposed method was described in theoretical way, examined in tests, and compared to other methods. The results confirm high efficiency and show that with each newly added processor sorting becomes faster and more efficient.

## 1. Introduction

Multicore architectures support a number of coworking logical processors that receive tasks distributed for parallel processing. Modern computing needs procedures that use advanced programming techniques oriented on performance, which can be achieved by parallelization of algorithms. The algorithms must be implemented in a way that preserves an appropriate separation of concerns which helps to avoid cross actions and interferences between processors. These aspects are important for the efficiency of data base systems and information processing, where a flexible usage of several processors improves performance. Bochenina et al. [1] presented a flexible method designed for parallelization of processing in simulation of stochastic Kronecker graphs. Czarnul et al. [2] proposed special environments for testing parallel applications on large distributed systems. A framework for distributed systems in health care monitoring was proposed by Mora et al. [3]. Esmaeili et al. [4] designed multiagent based algorithm and Stanescu et al. [5] described data base management for distributed systems. Sorting algorithms are used in all types of systems; therefore, improvements in these methods will benefit in many aspects. Research

on sorting methods covers many problems from storage to sorting itself. Depending on the idea we can find theoretical or practical advances that directly improve data management, speed of sorting, communication over hardware, storage, etc. Janetschek et al. [6] described how multicore architectures improve runtime environments. The results show that devoted algorithms when run in parallel may significantly improve efficiency. Parallelization of processes and devoted methods are very important for data management, especially for systems where we do not use frames that keep the order of information. De Farias et al. [7] presented devoted method for NoSQL systems based on regression. González-Aparicio et al. [8] described tests on NoSQL key-value data bases. We can see that methods, especially sorting, when parallelized in efficient way are very beneficial for data systems.

Classic versions of various sorting algorithms were presented by Aho and Hopcroft [9] and Knuth [10] among which three have main impact on the development in information processing: quick sort, heap sort, and merge sort. These methods are constantly improved to operate on modern architectures in the most efficient way.

Quick sort was improved by prevention of deadlocks and construction of the new pivot method. In the papers

of Bing-Chao and Knuth [11] faster exchange mechanism with the new pivot was presented. Francis and Pannan [12] discussed how to change partitioning of sorted strings by dynamic assignments, while Rauh and Arce [13] presented an improvement by application of median value for partitioning. Tsigas and Zhang [14] proposed an implementation oriented on efficiency for Sun Microsystems, Inc. The research on improvements for pivot procedure results in the new mechanisms of changing elements in the output string. In the article of Daoud et al. [15] an interesting mechanism for nonquadratic method was proposed, and in the work of Edmondson [16] the research on various pivot possibilities was discussed, while in the paper by Kushagra et al. [17] a multipivot procedure was proposed.

Heap sort benefited from new propositions of multilevel structures to store the data and improvements to the algorithms implemented for changing elements in this structure. Ben-Or [18] presented mathematical assumptions for modeling relations between elements in levels of the heap. Efficiency of reorganizing this structure was presented by Doberkat [19] and Wegner and Teuhola [20], while additional possibilities to boost the method by improved swap methods were proposed by Sumathi et al. [21]. Heap sort was also examined on various computing architectures (Roura [22]) with some possibilities for parallelization (Abrahamson et al. [23]).

Merge sort was improved by new ideas of sublinear methods and various approaches to parallelization. Carlsson et al. [24] discussed a sublinear procedure that was used for composition of sorted substrings. Theoretical background for the first approach to parallel version was presented by Cole [25]. An idea to compose devoted mechanism for partially sorted strings was discussed by Gediga and Düntsch [26]. Results from tests for new implementations were presented by Harris [27] and memory usage on various architectures was proposed by Salzberg [28] and Huang and Langston [29]. Zheng and Larson [30] discussed how to improve input-output operations for faster merging. In the paper by Zhang and Larson [31] dynamic memory assignments were proposed for various capacity of computing architectures. Buffering and reading from input strings were presented by Zhang and Larson [32]. Merge sort was also evaluated in various tests and benchmarks by Vignesh and Pradhan [33], Cheema et al. [34], and Paira et al. [35]. These research results show that merge sort has a very high potential for new improvements.

Quick sort, heap sort, and merge sort were also mixed and derived to present new methods of sorting. Speed of sorting that increased by virtual memory assignments was discussed by Alanko et al. [36] and Larson and Graefe [37]. Cash usage was examined by LaMarca and Ladner [38]. Skewed strings and other input types were examined by Crescenzi et al. [39], while derivatives composed to adapt to the input were presented by Estivill-Castro and Wood [40]. Self-sorting by adaptation of Markov idea for chain rules was proposed by Axtmann et al. [41], while Abdel-Hafeez and Gordon-Ross [42] proposed a free sorting approach. Mohammed et al. [43] proposed an insertion of the elements into the output string by application of bidirectional method of sorting.

*1.1. Related Works.* During research on improved and new methods of sorting we have proposed improvements to classic versions. Dynamic division of length for quick sort was proposed by Woźniak et al. [44]; this gave an improvement in sorting, prevented deadlocks, and sped up the quick sort of about 10%. Heap structure for faster processing of large data sets was discussed by Woźniak et al. [45]. We have described how to compose and search the structure of the heap in a way that speeds up sorting of about 5% to 10%. The research on improvements for merge sort gave new, faster processing of input string but also improved management of the data during iterations in the algorithm. In [46] we have shown that dynamic rule of merging speeds up the process of about 10%. In [47] we have proved that nonrecursive sorting makes the merge method flexible to various architectures, and in [48] our ideas were reported for Hadoop systems. A theoretical introduction to parallelization of sorting was presented by Cole [25]. The results given there have shown the way to divide the tasks between processors using the idea of binary trees. Uyar [49] presented an approach to first parallelization of merge sort, after which a new method was proposed by Marszałek [50] and Marszałek [51].

The algorithm we would like to discuss here assumes a new concept of sorting by the use of independent merge. Presented in this article is a method that makes the algorithm sorting in a fully flexible way, which means that with each new processor the algorithm gains additional capacity of sorting. The new method we present here is implemented in the way that preserves separate concerns executed on each logical processor. This makes the novelty of one of the most important aspects for this algorithm. The model for this method is using concept of allocation of strings in memory block for the currently merging processor. The idea is based on the PRAM model. The model of PRAM machine is a theoretical assumption of parallel processing; therefore, it does not consider hardware aspects like delay in communication and information exchange between registers in the system. The proposed approach is moving away from the dynamic splitter strings. For the independence of the input and flexible execution of efficient sort, a certain presorted string approach is proposed. This approach allows us to estimate the running time of the algorithm regardless of the input data with a certain processing model. The sorting tasks are divided between processors so that each of them is sorting without any interruptions or cross actions. In benchmark tests the algorithm was about 15% more efficient in comparison to other sorting methods. The proposed parallel sorting has theoretical time complexity  $O((\log_2 n)^2)$ .

The proposed model of independent and fully flexible sorting was implemented in C# MS Visual 2015 on MS Windows Server 2012. For the research we have used Opteron AMD Processor 8356 8p. The algorithm has been described in theoretical analysis and examined in practical benchmark tests. The results we present in this article show that this algorithm is fast and gives very good improvement when run on multicore architectures.

The concept of faster sorting is one of the main topics for recent advances in architectures and big data processing for complex Internet of things systems. Modern computing

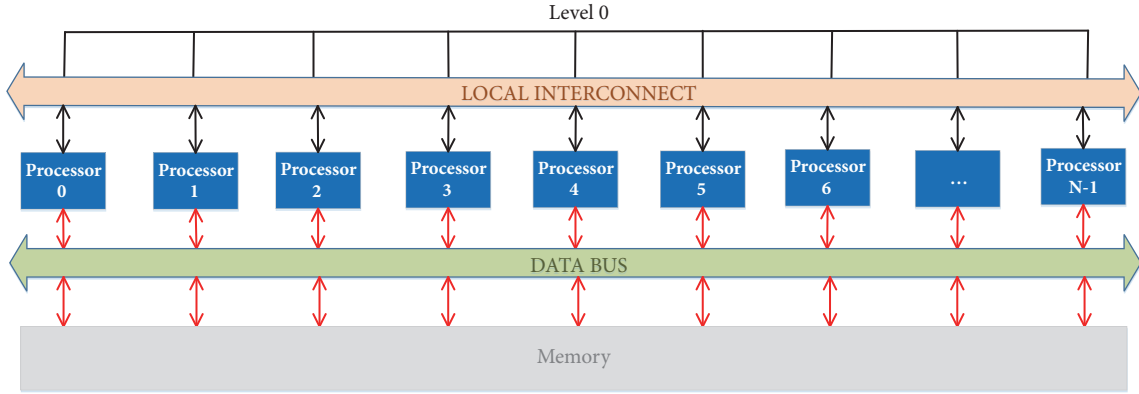


FIGURE 1: Parallel Random Access Machine.

requires more and more information; therefore, systems devoted to faster processing will give an impact on the final efficiency in data mining easing work for the users. Our solution fits this concept both for theoretical and practical aspects.

## 2. Data Processing in NoSQL Database and Parallel Sorting Algorithms

A significant problem in theory of computational complexity is the acceleration time of sorting algorithms. Mainly it is improved by dividing the input into parts and executing all tasks concurrently by multiple processors. The new approach we discuss here runs into the limits to show a fully flexible method of this concept. What to do when we can no longer divide inputs into smaller portions, and we would like to make the sorting faster by using multiple processors? A solution to this example can be an independent choice of the operating processor. In this article we present how to implement parallel sorting method with time complexity  $O((\log_2 n)^2)$ .

The architecture of modern processors allows performing in a parallel way multiple processes. For the analysis of algorithms run on modern computers we use the theoretical multiprocessor machine model PRAM (Parallel Random Access Machine) presented in Figure 1. The efficiency of the algorithm depends on its time complexity and memory usage. The abstract model of the PRAM machine is used to analyze parallel algorithms in terms of complexity. PRAM is defined as a system  $\langle P, I, M \rangle$  composed of processors  $P_i$  for  $i = 0, \dots, N-1$ , instructions  $I$  for each of them and memory  $M$  accessed during these operations. Due to the access of the processors to read from and write into memory we can talk about four PRAM machines:

- (i) Concurrent Read Concurrent Write (CRCW)
- (ii) Exclusive Read Concurrent Write (ERCW)
- (iii) Concurrent Read Exclusive Write (CREW)
- (iv) Exclusive Read Exclusive Write (EREW)

The first two types of machine PRAM are typically theoretical models of computing. The third type allows to access memory

using any processor but only one processor can read from the memory and write into the memory cell at the same time. If two processors are simultaneously writing to the same memory cell, the result of the operation is undefined. For the evaluation of our idea we used a type that allows to read and to write on a single processor. Presented in this work was an algorithm developed and implemented using PRAM model to show its theoretical aspects, but the efficiency was verified in practical benchmark tests and compared to other methods.

*Definition 1.* The time complexity of the algorithm run on the PRAM machine can be defined as the time (number of instructions) of the longest procuring processor during the execution of the algorithm.

*Definition 2.* The memory complexity is the number of cells used by the PRAM machine to perform the algorithm.

*Definition 3.* The processor complexity is the maximum number of active processors during computations.

For computational analysis of sorting algorithms, which use comparisons of sorted sequence elements, a simplified calculation model is appropriate. We define the number of comparisons made by the algorithm during sorting and on this basis we determine the complexity of time, processor, and memory for sorting methods on the PRAM machine. The presented calculation model adequately determines the operation time of the sorting algorithm on the computer which performs the entire sorting in the operating memory. For the analysis of external sorting algorithms, other calculation models are used, e.g., No-Remote Memory Access (NoRMA) or Uniform Memory Access (UMA).

The Fully Flexible Parallel Merge Sort method describes how to split the sorting processes between independently working processors. A general scheme of task division between processors is presented in Algorithm 1 and in a flow chart shown in Figure 2. Each merged pair of strings is allocated in memory block in which the currently merging processor can read from and write to memory locations. The algorithm constructed in a classic way has the time complexity  $O(n)$ . We propose a new way to merge pairs of



```

1: for  $t = 0$  to  $\lceil \log_4 n \rceil$  do
2:   For current  $t$  processor, determine the starting number of the string  $a_i$  to locate element,
3:   Find the boundary indexes of the string to insert  $a_i$ ,
4:   Find the index of the next element before which the  $a_i$  element is inserted,
5:   For the computed index insert  $a_i$  element into array  $b$ ,
6:   Wait for all processors,
7:   For current  $t$  processor, determine the starting number of the string  $b_i$  to locate element,
8:   Find the boundary indexes of the string to insert  $b_i$ ,
9:   Find the index of the next element before which the  $b_i$  element is inserted,
10:  For the computed index insert  $b_i$  element into array  $a$ ,
11:  Wait for all processors,
12: end for

```

ALGORITHM 1: The algorithm to divide tasks between processors  $P_i$   $i = 0, \dots, n$ .

sorted strings of  $n/2$  elements by  $n$  independently working processors. The novelty of our method is in flexibility of implementation. We present a solution using the principle that each processors can read from the memory cells but only one can write to a cell that no other processor is using. Therefore the proposed method has a very high efficiency without cross actions, what results in higher efficiency with each additional processor. The proposed algorithm of combining two sequences using  $n$  independently working processors has time complexity  $O((\log_2 n)^2)$ . This result was proved in theory and examined in benchmark tests, what we discuss in the following sections of the article.

### 3. Efficient Parallel Sorting

New computers make it possible to implement methods that parallelize processing. Main requirements for these methods are low computational complexity and flexibility to adjust to various multicore architectures. Among possible applications of these methods we can define data base systems and information management systems. This is due to the fact that new machines with a number of processors can divide the tasks between logical processors and therefore the requests are answered in a shorter time. The efficiency of this stage depends on implemented method, which must be flexible for any number of processors and preserve the sorting algorithm without any unnecessary cross actions or interruptions between processors.

**3.1. Fully Flexible Parallel Merge Sort Algorithm.** A well-known classic merge sort algorithm to merge two sorted strings of  $n$  elements allows to merge them into a single sorted sequence by doing so at maximum  $2n-1$  comparisons and not less than  $n$  comparisons. This algorithm is difficult to perform independently on several processors. We can of course try to parallelize merge by the use of two independently working processors; however, this idea is not efficient. Here we present a novel approach to efficiently merge in parallel way so that the processors do not interfere with each other and the method is flexible for various numbers of processors. We use the possibilities of modern computers and the fact

that all processors can simultaneously read from memory cell, but at the same time only one can write into the memory.

Now suppose that we have two sorted strings to merge  $n/2$  elements in array  $a_0 \leq a_1 \leq \dots \leq a_{n-1}$ . The first half of the original string is stored in  $a_0 \leq a_1 \leq \dots \leq a_{n/2-1}$  and the second half of the original string is stored in  $a_{n/2} \leq \dots \leq a_{n-1}$ . The algorithm inserts an element of the array  $a$  into the array  $b = [b_0, b_1, \dots, b_{n-1}]$  using processors with indexes  $i, 0 \leq i < n$  in simple steps. Each of processors in the loop has a unique index and transmits the information about the dimension of the merged string.

Processor number  $i, 0 \leq i < n/2$  computes the index of the element  $a_i$  on the second half of the original string before which the element should be inserted to have  $a_i, a_{t-1} < a_i \leq a_t$ . In case where the insertion must be done after the last element of the array  $a$ , the value of the index is  $n$ . Processor  $i$  inserts the value of the element  $a_i$  in the array  $b$  under the index  $i + t - n/2$ . Imagine, for instance, a way to merge two strings stored in the array  $a = [2, 5, 7, 9, 0, 2, 4, 8]$  using processors  $P_0, P_1, P_2, P_3$  which insert elements into the array  $b$ . The situation is shown in Figure 3.

Each of the processors  $P_i, i = 0, 1, 2, 3$  operates independently and determines the index  $t_i$  of the element in the second half of the original string, before which the element should be inserted. For example, the processor  $P_0$  inserts the element  $a_0 = 2$  prior to  $a_5 = 2$ . Hence, the index is calculated and inserted element 2 into the array  $b$  is equal to the sum of the indexes of elements  $a_1 = 2$  and  $a_5 = 2$  minus 4 and is 1, see Figure 3. Processor number  $i, n/2 \leq i < n$  computes the index of the element  $a_i$  on the first half of the original string before which the element should be inserted  $a_i, a_{t-1} \leq a_i < a_t$ . In case where the insertion must be done after the last element of the array  $y$ , the value of the index is  $n/2$ . Processor  $i$  performs insertion of the value of the element  $a_i$  into the output string  $b$  under the index  $i + t - n/2$ . Imagine, for instance, a way to merge two strings recorded in array  $a = [2, 5, 7, 9, 0, 2, 4, 8]$  using processors  $P_4, P_5, P_6, P_7$  which insert elements into the array  $b$ . The situation is shown in Figure 4.

Each of the processors  $P_i, i = 4, 5, 6, 7$  operates independently and determines the index  $t_i$  of the element in the first half of the array, before which the element should be



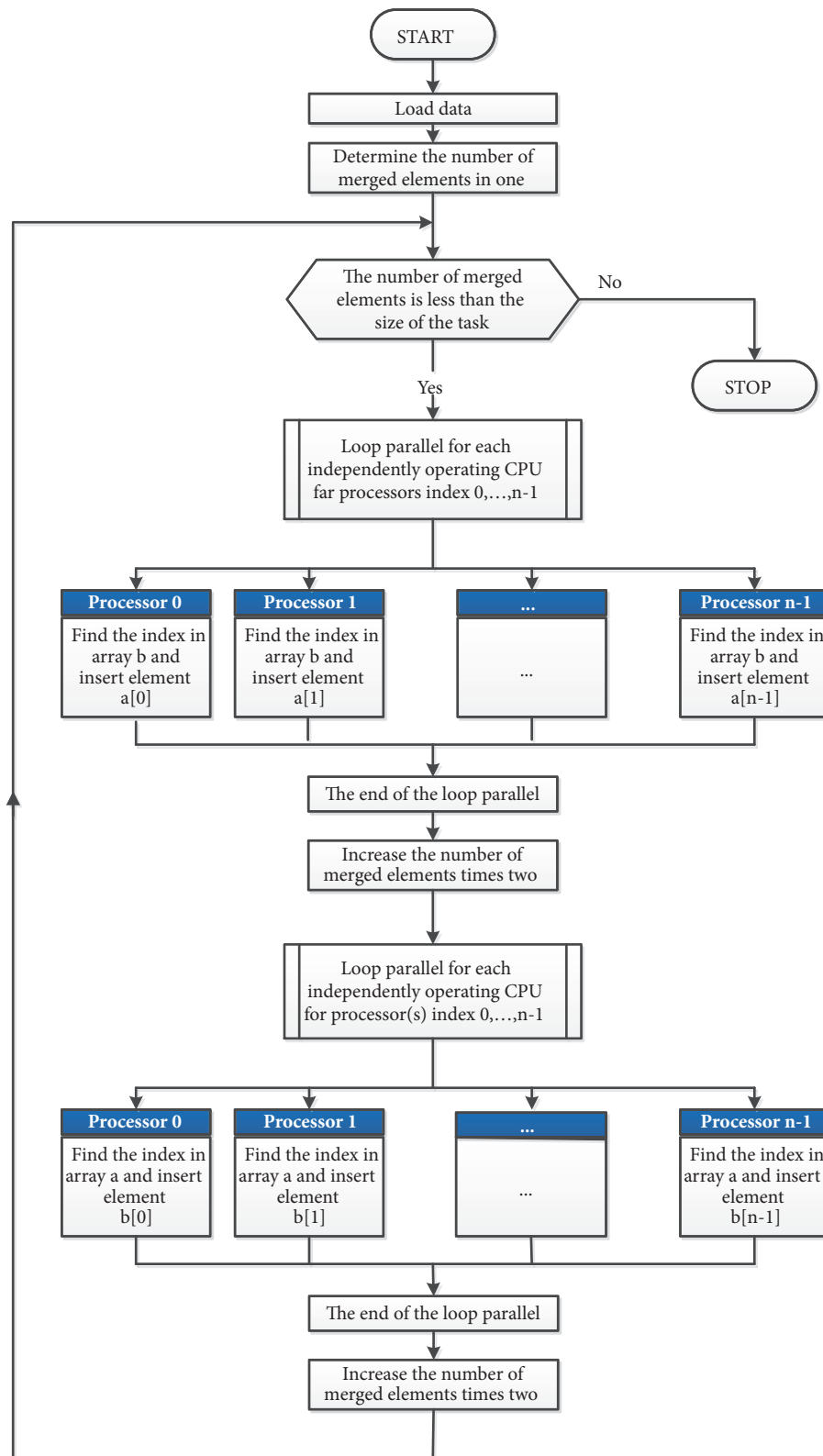
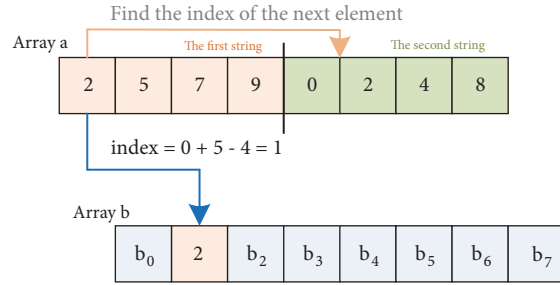
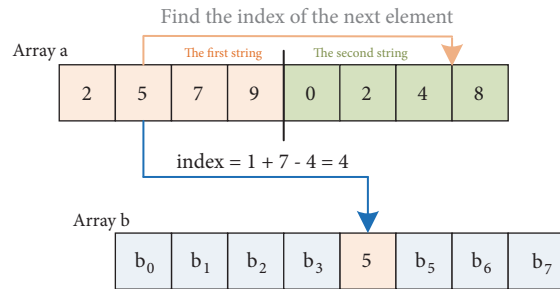


FIGURE 2: The block diagram of iterative task assignment to the processors while implementing proposed flexible algorithm that adapts to the number of used processors.

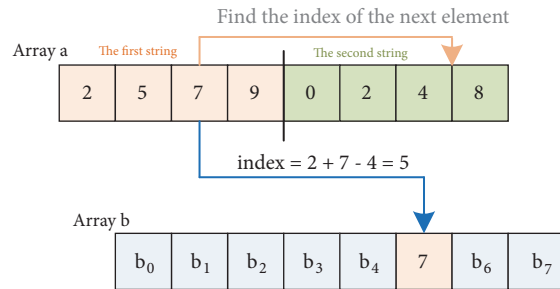
Processor 0



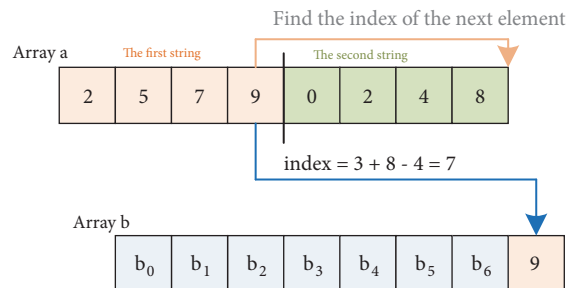
Processor 1



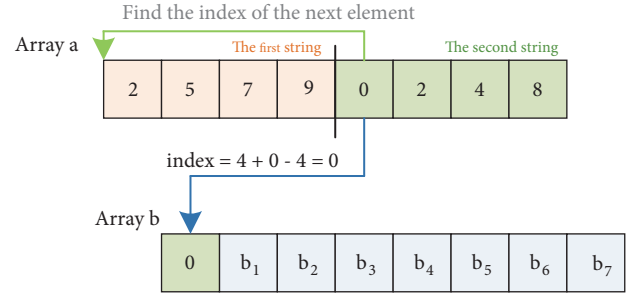
Processor 2



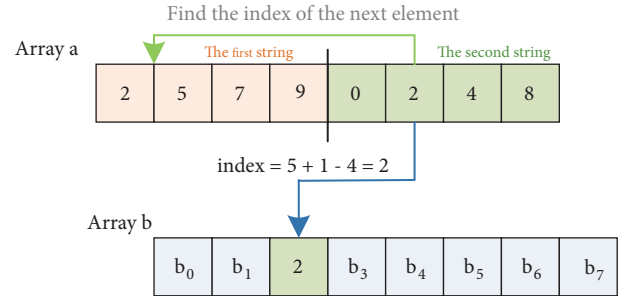
Processor 3



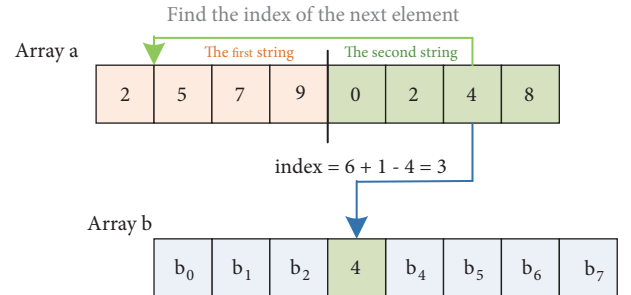
Processor 4



Processor 5



Processor 6



Processor 7

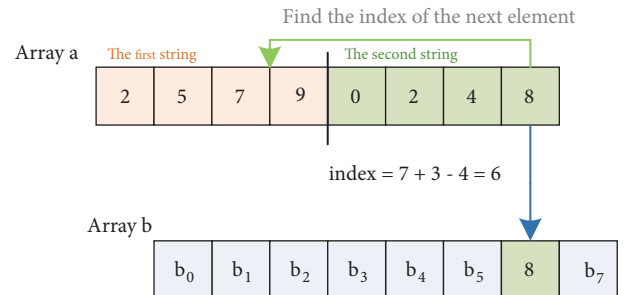


FIGURE 3: Model of the applied flexible sorting for the number of processors for the insertion of elements into the array  $b$ : processor  $P_0$  sorts element  $a_0$  into the array  $b$ , processor  $P_1$  sorts element  $a_1$  into the array  $b$ , processor  $P_2$  sorts element  $a_2$  into the array  $b$ , etc.

FIGURE 4: Model of the applied flexible sorting for the number of processors for the insertion of elements into the array  $b$ : processor  $P_4$  sorts element  $a_4$  into the array  $b$ , processor  $P_5$  sorts element  $a_5$  into the array  $b$ , processor  $P_6$  sorts element  $a_6$  into the array  $b$ , etc.

```

1: The dimension of the array  $a$  is  $n$ ,
2: Create an array  $b$  of dimension  $n$ ,
3: Set options for parallelism to use all processors of the system,
4:  $t \leftarrow 1$ ,
5: while  $t < n$  do
6:    $m2 \leftarrow 2 * t$ ,
7:    $m4 \leftarrow 4 * t$ ,
8:   parallel for  $i1 \leftarrow 0$  to  $n-1$  do
9:      $j \leftarrow i1/m2$ ,
10:     $i \leftarrow m2 * j$ ,
11:     $iw \leftarrow i1 \% m2$ ,
12:     $p1 \leftarrow i + t$ ,
13:    if  $p1 > n$  then
14:       $p1 \leftarrow n$ ,
15:    end if
16:     $p2 \leftarrow i + m2$ ,
17:    if  $p2 > n$  then
18:       $p2 \leftarrow n$ ,
19:    end if
20:    if  $i1 < p1$  then
21:       $iz \leftarrow \text{IndexRight}(a, p1, p2, a[i1])$ ,
22:       $b[iz + i + iw] \leftarrow a[i1]$ ,
23:    else
24:       $iz \leftarrow \text{IndexLeft}(a, i, p1, a[i1])$ ,
25:       $b[iz + i1 - t] \leftarrow a[i1]$ ,
26:    end if
27:  end parallel for
28:  parallel for  $i1 \leftarrow 0$  to  $n-1$  do
29:     $j \leftarrow i1/m4$ ,
30:     $i \leftarrow m4 * j$ ,
31:     $iw \leftarrow i1 \% m4$ ,
32:     $p2 \leftarrow i + m2$ ,
33:    if  $p2 > n$  then
34:       $p2 \leftarrow n$ ,
35:    end if
36:     $p3 \leftarrow i + m4$ ,
37:    if  $p3 > n$  then
38:       $p3 \leftarrow n$ ,
39:    end if
40:    if  $i1 < p2$  then
41:       $iz \leftarrow \text{IndexRight}(b, p2, p3, b[i1])$ ,
42:       $a[iz + i + iw] \leftarrow b[i1]$ ,
43:    else
44:       $iz \leftarrow \text{IndexLeft}(b, i, p2, b[i1])$ ,
45:       $a[iz + i1 - m2] \leftarrow b[i1]$ ,
46:    end if
47:  end parallel for
48:   $t \leftarrow 4 * t$ ,
49: end while

```

ALGORITHM 2: Proposed flexible parallel merge.

inserted. For example, processor  $P_4$  inserts the element  $a_4 = 0$  prior to  $a_0 = 2$ . Hence, the index is calculated and inserted element 0 into the array  $b$  is equal to the sum of the indexes of elements  $a_4 = 0$  and  $a_0 = 2$  minus 4 and is 0, see Figure 4. A sorting algorithm which can flexibly involve each additional processor is presented in Algorithm 2. Algorithm for finding the index of the element before inserting the new element has time complexity  $O(\log_2 n)$ .

Because in our method the processors operate independently, we use the machine CREW PRAM model with  $n$  processors. The most commonly used model for analysis of parallel algorithms is machine PRAM. In practice, we distinguish between three variants of this model. The first model the Exclusive Read Exclusive Write PRAM gives you the ability to read from and write into memory by only one processor. The second model the Concurrent Read

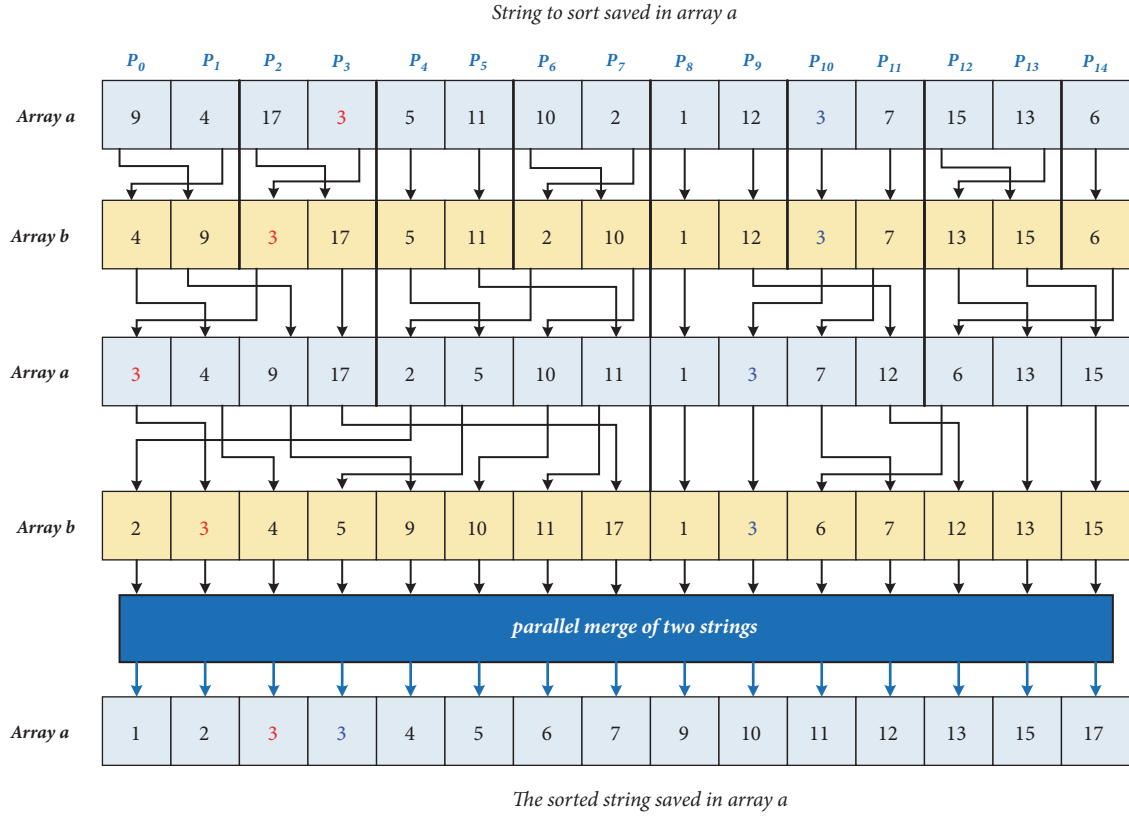


FIGURE 5: Fully Flexible Parallel Merge Sort Algorithm sorting the input array. The method is a stable method in the sense of keeping the order of arranging elements of the same value in a sorted sequence. In the drawing, this is shown on the element with value 3, which has been marked with different colors. In the sorted sequence, the elements remain in the stacking order, such as that they had in the input of not sorted sequence.

Exclusive Write gives the right to read from memory by any processor and the right to write to at the same time by only one processor. The third model of the Concurrent Read Concurrent Write PRAM enables simultaneous access of all processors to memory. Therefore in the description and experimental research we use CREW PRAM model as it is the most suitably fitting the operations in modern computer architectures. Proposing fully flexible algorithm on CREW PRAM model for merging two strings will perform in time  $O((\log_2 n)^2)$ .

In each iteration, the algorithm presented in Figure 5 merges four successive ordered sequences into one ordered numerical sequence. Each iteration is divided into two steps. In the first step, the next two strings from the array *a* are merged and the result of merging is stored in the table *b*. In the second step, the algorithm merges the next two ordered sequences from the array *b* and merged result is written into the array *a*. So each iteration increases the size of the ordered sequences four times. Because the algorithm uses a parallel version to merge two numeric strings, so in each iteration it can use  $n$  processors at every step which makes it flexible to the number of processors. For example, in the first iteration in the first step, two one elemental strings are merged by  $n$  processors; therefore,  $n/2$  times the algorithm of parallel merging of two numerical sequences is run on independently working processors.

**3.2. Theoretical Aspects of Computational Complexity.** The Algorithm 2 is a flexible parallel merging of numeric strings without cross actions or interruptions. The presented method is flexible and can be adjusted to a large number of logical processors; therefore, we call it Fully Flexible Parallel Merge Sort (FFPMS). Each step of the merge was divided into two stages. First, the method merges each pair of strings into the temporary array. If at the end of the input array is one element string, it will be rewritten into the temporary array. Then the method merges each pair of sorted strings from the temporary array into the output array. Similarly as in the first stage, if there is one element at the end of the string, it will be rewritten from the temporary array into the output array. The result in the output array is increased four times. To perform the merge was developed the new parallel algorithm that determines the index of each element  $a_i$ ,  $i = 0, \dots, n-1$ . For each of the processors  $P_i$ ,  $i = 0, \dots, n-1$  we call developed search algorithm to select the index of the element before which the new element  $a_i$  should be inserted. Because all processors can simultaneously read the memory and each processor executes the insertion of an element in another cell, the whole process can be run simultaneously without cross actions and interruptions. Figure 6 shows the first step of the first iteration of merging one element strings with the sorted strings of two elements stored in the temporary array. Way to parallelize it further in the process of merging  $n$  strings



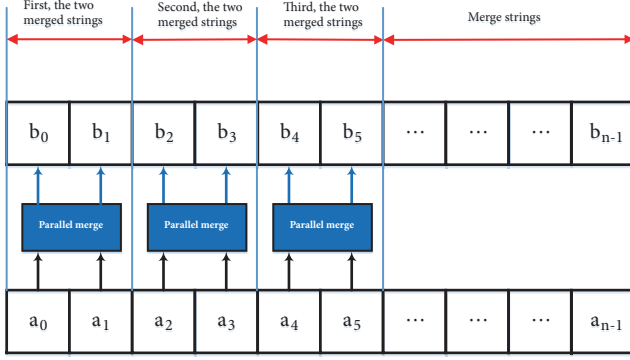


FIGURE 6: The first step of the proposed flexible parallel merge sort, when the algorithm merges elements of the input strings in pairs.

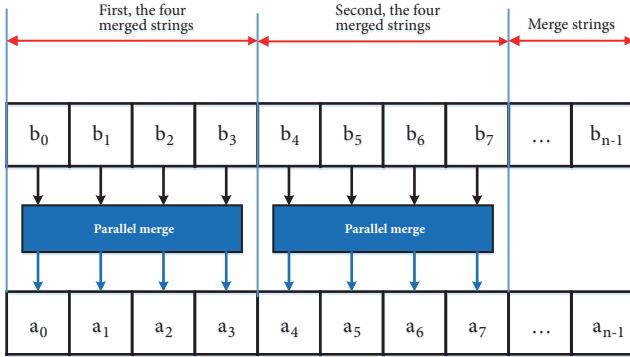


FIGURE 7: The following operations in the proposed flexible parallel merge of the temporary array into the input array.

is shown in Figure 7. In all the steps of the algorithm we merge the data in the same way, strings are enlarged four times and the odd indexed element is rewritten until the final merge, see Figure 8. The method is stable in the sense of keeping the order of arranging elements of the same value in a sorted numerical sequence. In Figure 5 we can see sample sorting, where stability of the order is shown on the elements with value 3, which have been marked with different colors. In the sorted sequence, the elements keep the order of the arrangement, such as they had in the input sequence.

**Theorem I.** *Proposed Fully Flexible Parallel Merge Sort for  $n$  processors has time complexity  $O((\log_2 n)^2)$ .*

*Proof.* We are limiting deliberations to  $n = 4^k$ , where  $k = 1, 2, \dots$

Let us first notice that sequences  $a_0 \leq a_1 \leq \dots \leq a_{n/2-1}$  and  $a_{n/2} \leq \dots \leq a_{n-1}$  of  $n/2$  elements can be merged into one sequence  $b_0 \leq \dots \leq b_{n-1}$  using  $n$  processors. Moreover parallel merging of these two sequences makes no more than  $\lceil \log_2 n \rceil + C$  comparisons by each of the processors. Thus, time complexity of the parallel algorithm to merge two strings on a CREW PRAM machine model is  $O(\log_2 n)$ .

At each step  $t = 1, \dots, k = \log_4 n$ , in the beginning the algorithm saves into the temporary array two merged halves of the original string. Next, we merge them from the

temporary array and save the result in the array of the sorted elements. Because all processors work independently, thread synchronization happens after each stage. The maximum operating time of each step of the merger of four strings can be written in the form

$$\begin{aligned} T_{\max}(t) &= \lceil \log_2 2^{t-1} \rceil + \lceil \log_2 (2 \cdot 2^{t-1}) \rceil + C_1 \\ &= 2 \lceil \log_2 2^{t-1} \rceil + C_2 \end{aligned} \quad (1)$$

The first component  $\lceil \log_2 2^{t-1} \rceil$  is the maximum time for all  $2^{t-1}$  components to merge. The second component  $\lceil \log_2 (2 \cdot 2^{t-1}) \rceil$  is time to merge the double-expanded strings after merging strings in the first stage and writing them in the temporary table. The  $C_1$  constant consists of a fixed number of operations performed both when merging items into temporary tables and a fixed number of operations when merging strings from a temporary table into an array of ordered items. It is independent of step  $t$ . By transforming

$$\begin{aligned} \lceil \log_2 (2 \cdot 2^{t-1}) \rceil &= \lceil \log_2 2 \rceil + \lceil \log_2 2^{t-1} \rceil \\ &= 1 + \lceil \log_2 2^{t-1} \rceil \end{aligned} \quad (2)$$

and inserting into (1) it yields that  $2 \lceil \log_2 2^{t-1} \rceil + C_2$  equals the maximum time that all processors execute sorting in each step  $t$ . Then we aggregate all the sorting times  $\log_4 n$  performed by the processors in each step  $t = 1, \dots, k$  and write down the sum of the times as

$$T_{\max} = \sum_{t=1}^k T_{\max}(k) = 2 \sum_{t=1}^k \lceil \log_2 2^{t-1} \rceil + C_2 \log_4 n \quad (3)$$

when calculating

$$\begin{aligned} \sum_{t=1}^k \lceil \log_2 2^{t-1} \rceil &= \lceil \log_2 2 \rceil + 2 \lceil \log_2 2 \rceil + \dots \\ &\quad + (k-1) \lceil \log_2 2 \rceil \\ &= \lceil \log_2 2 \rceil [1 + 2 + \dots + (k-1)] \\ &= \frac{k(k-1)}{2} \lceil \log_2 2 \rceil \\ &= \frac{\log_4 n (\log_4 n - 1)}{2} \lceil \log_2 2 \rceil \end{aligned} \quad (4)$$

it is assumed that  $n = 4^k$  and the result is consistent with our expectation, since the sum of the arithmetical progress of natural numbers is equal  $k(k-1)/2$ , where  $k = \log_4 n$ . Intuitively, it can be said that in the subsequent stages of the merge process each of the independent processors searches for the index of the item being rendered by means of a binary search of time complexity  $O(t)$ ,  $t = 1, \dots, \log_2 n$ . Since the number of algorithm steps is  $\log_2 n$ , we obtain the method of time complexity  $O[(\log_2 n)^2]$ . Therefore by substituting and

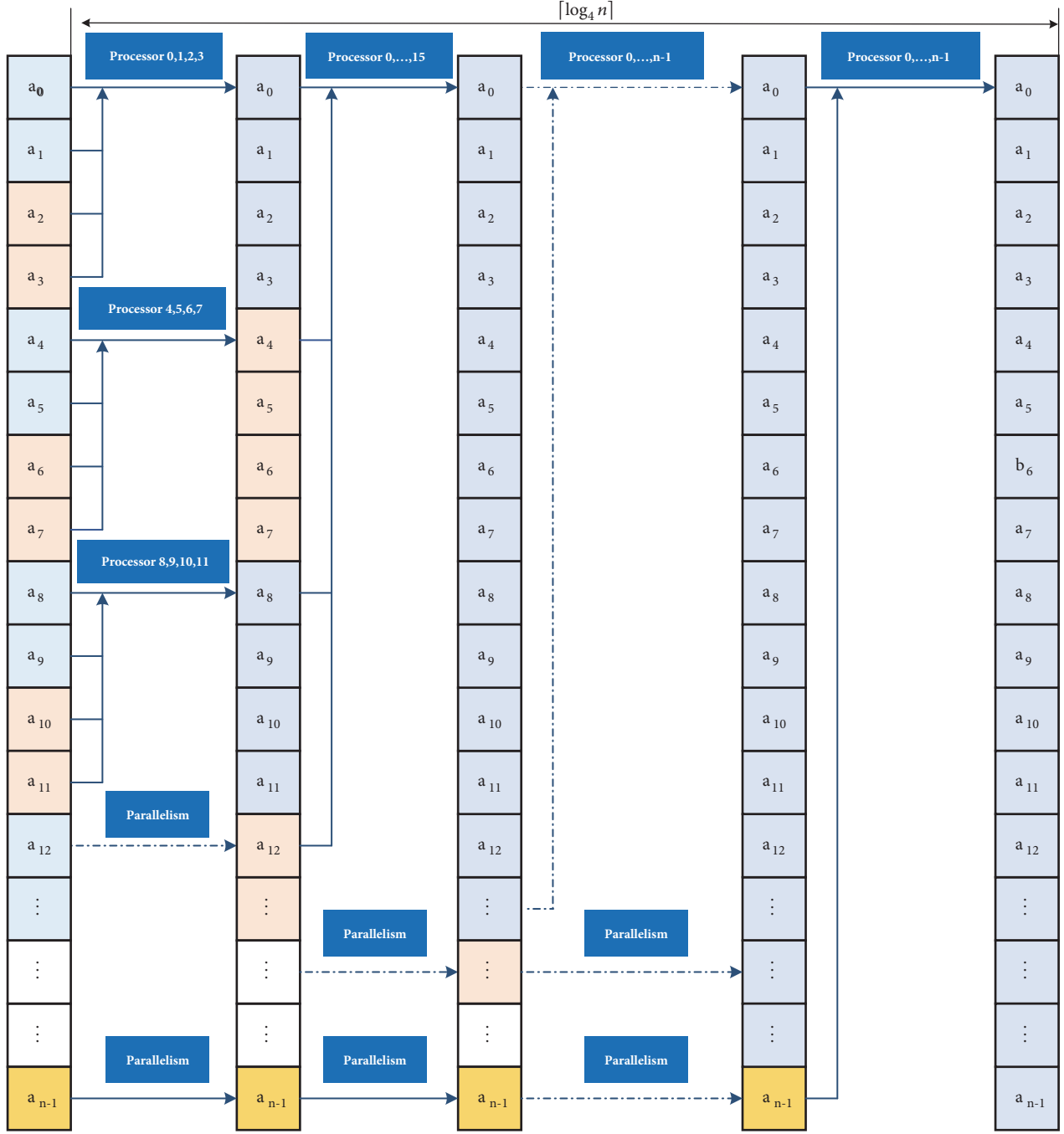


FIGURE 8: Fully Flexible Parallel Merge Sort Algorithm.

taking into account  $\lceil \log_2 2 \rceil = 1$  and  $\log_4 n = \log_2 n / \log_2 4 = \log_2 n / 2$ , we get

$$\begin{aligned}
 T_{\max} &= (\log_4 n)^2 + (C_2 - 1) \log_4 n \\
 &= \frac{(\log_2 n)^2}{4} + \frac{(C_2 - 1) \log_2 n}{2}
 \end{aligned} \tag{5}$$

which was proved.  $\square$

The square logarithm of the algorithms time complexity is the result of separating the work of processors in each iteration of the merge strings. Each processor operates independently, but it performs more string comparisons to determine the index of the inserted element. This enables each iteration to use the same number of processors working independently.

**Theorem II.** By using  $k$  processors for the parallel sorting method on CREW PRAM, we can lower the time complexity to  $O(n(\log_2 n)^2/k)$ .

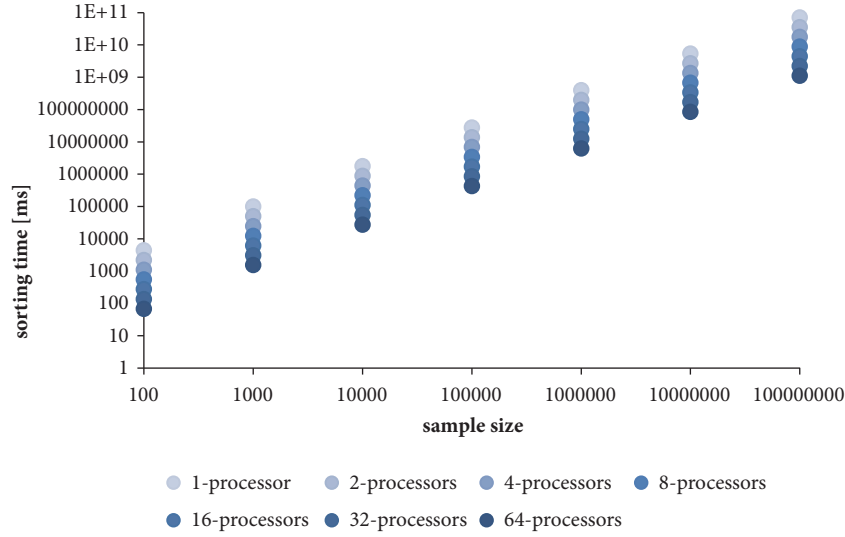


FIGURE 9: Chart of the theoretical complexity for the proposed method showing the potential growth in efficiency of the proposed approach for multicore architectures up to 64 cores.

*Proof.* The proof comes as natural derivation from the proof of Theorem I.  $\square$

If we assume that the time increment for an algorithm with the increase in the task dimension is a unit of time, then for parallel sorting method by merging we obtain the following graph of the algorithm's theoretical run time, see Figure 9. This theoretical result of  $O(n(\log_2 n)^2/k)$ , where  $k$  is the number of processors, is an asymptotic result which does not take into account the delays that may occur when data is transmitted on the data bus and other operations by the system as a result of the implementation. Hence, the difference in the speed of the algorithm is only visible for a sufficiently large task dimension. Analyzing Figure 9 we see that with each new added processor the method should gain additional advantage in sorting big data sets. The chart shows theoretical results for architectures of up to 64 logical processors. We can conclude that in theoretical way from 1 to 64 logical processors the method should gain about 10% to 15% on efficiency which is very important for big data sets. The proposed algorithm is fully scalable and can be executed for a veritable dimension of the sorted sequence of numbers using the specified number of processors without any problems.

For benchmark tests the method was implemented in C# Visual Studio Enterprise 2015. To assign tasks and to facilitate processors loop *Parallel For* available in C# language has been used, which enables the usage of the maximum number of processors available in the system. In the method we have two additional functions targeted at the delivery of the index of the element before inserting the new ones. The first function returns an index to the next element in the string on the right side, see Figure 10. The second function returns an index of the next element in the string on the left side, see Figure 11.

#### 4. The Study of the Algorithm

Performance analysis is based on benchmark tests for the algorithms implemented in C# in Visual Studio 2015

Enterprise on MS Windows Server 2012, namely: quick sort, heap sort, and classic merge, and presented here is flexible parallel merge. For testing were used 100 samples generated at random for the task size from 100 to 100 000 000 elements, increasing the size of sorted array ten times in the following experiments. In addition, for each set of 100 samples generated randomly for a given dimension size were added samples consisting numbers ascending and descending, as well as samples containing numbers which compose a critical situation for sorting algorithms (Woźniak et al. [44]). The number of samples was chosen as 100 since it is a standard statistical number to examine proposed methods in benchmark tests.

**4.1. Benchmark Tests.** Let us now show practical verification of the computational complexity and comparisons to other methods. We are interested in presenting how the method works, but we do not assume delays from hardware, bus of the motherboard, hard disk connectors, etc. The tests we present here are focused on measuring efficiency of applied processors. We assume that tests are free of other delays and compare the results to evaluate how the sorting methods are run on processors. In comparisons we have used statistical measures. Each of the experiments was verified for sorting time measures in [ms] and Central Processing Unit (CPU) operations measured in [ti]. The arithmetic mean is equal to the mean value of the measurements from the experiments

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (6)$$

The standard deviation from the expected value is computed as

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (7)$$

where  $n$  represents the number of measurements  $x_1, x_2, \dots, x_n$ , and  $\bar{x}$  is the arithmetic mean (6). The

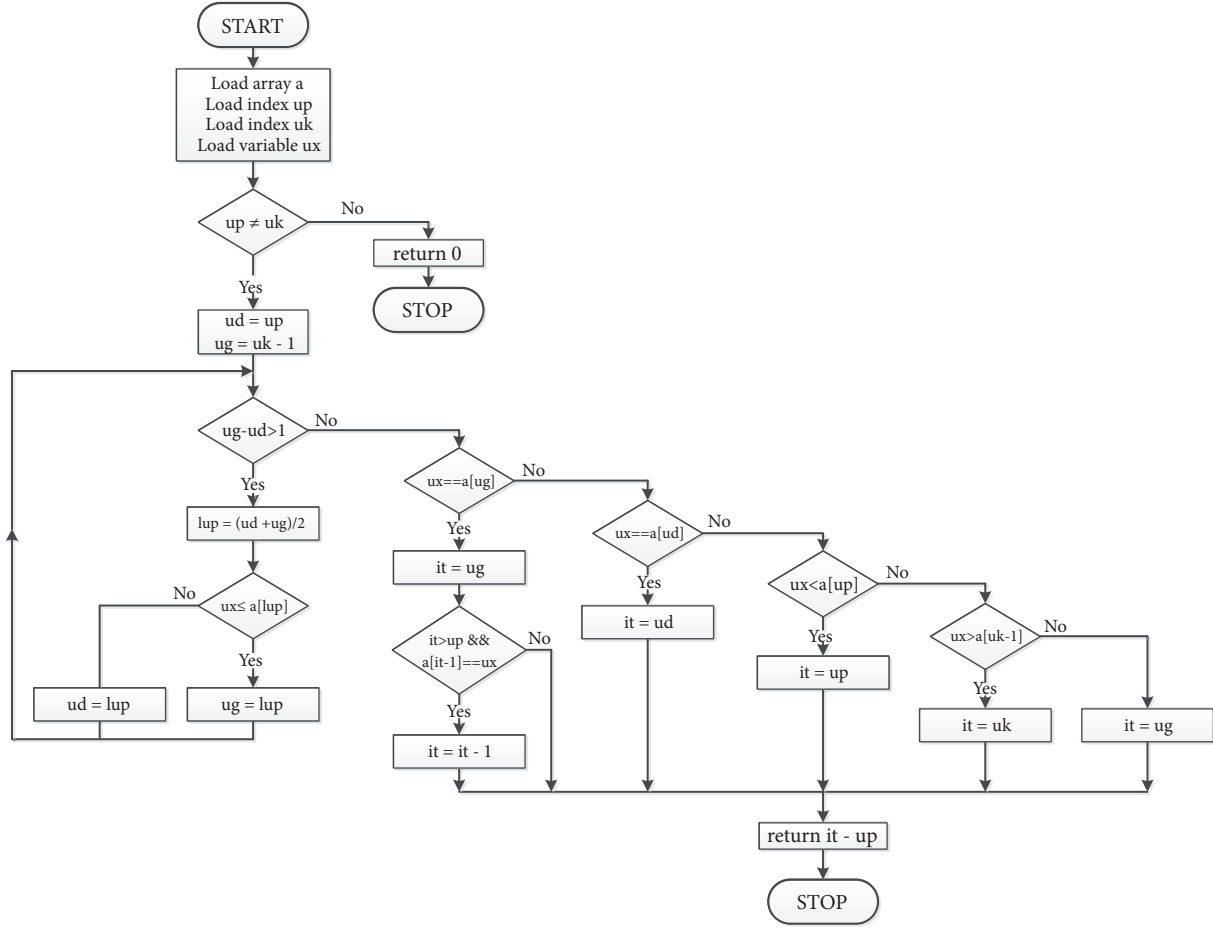


FIGURE 10: The block diagram of the function which returns index located in the right string.

coefficient of variation presents possible diversity between the results and is computed as

$$V = \frac{\sigma}{\bar{x}} \quad (8)$$

where the symbols are the arithmetic mean (6) and the standard deviation 7. Each sorting operation was measured in time [ms] and CPU (Central Processing Unit) usage represented in tics of CPU clock [ti]. Tests were carried out on quad core amd opteron processor 8356 8p. These results are averaged for 100 experiments to show comparison presented in Figures 12 and 13. The results of the newly proposed method are presented in Table 1 for time and Table 2 for computing operations. Comparison of coefficient of variation is presented in Tables 3 and 4.

Analyzing Tables 1 and 2 we see that with each new processing core the efficiency is extended and the sorting is done faster. While from Tables 3 and 4 we see that the algorithm for any number of CPUs used in the research has almost the same stability for large data sets. Some variations in stability of the algorithm for small inputs are due to the fact that the system exceeds the sorting capability automatically, what has an influence on the performance.

**4.2. Analysis and Comparison.** Let us compare the algorithms assuming that duration of the method using only one processor is a base line and let us examine if the duration is shorter when using multiple processors. The results are shown in Figures 14 and 15. Comparing the results in Figures 14 and 15 we can see that each new processor gives additional boost to sorting by decreasing sorting time and necessary operations and therefore makes the method more efficient. The most spectacular difference is between one and two processors. Figure 14 shows that time of processing can be really shorter for large collections. For collections above 10 000 elements we can see the boost in sorting time and the differences by the use of additional processors become visible. For collections above 1 000 000 elements the sorting time becomes shorter with each newly added processor of about 10%-15%. The dashed lines representing trends confirm this and show that additional processors seriously boost the proposed method. Figure 15 shows that the application of further processors improves the method of about 40% if we switch from 1 processor to 2 processors. Additionally for adding the next 2 processors we can get another 20% of efficiency. Next processors give about 10% to 15% rise in the efficiency with each new logical core used in the method. The



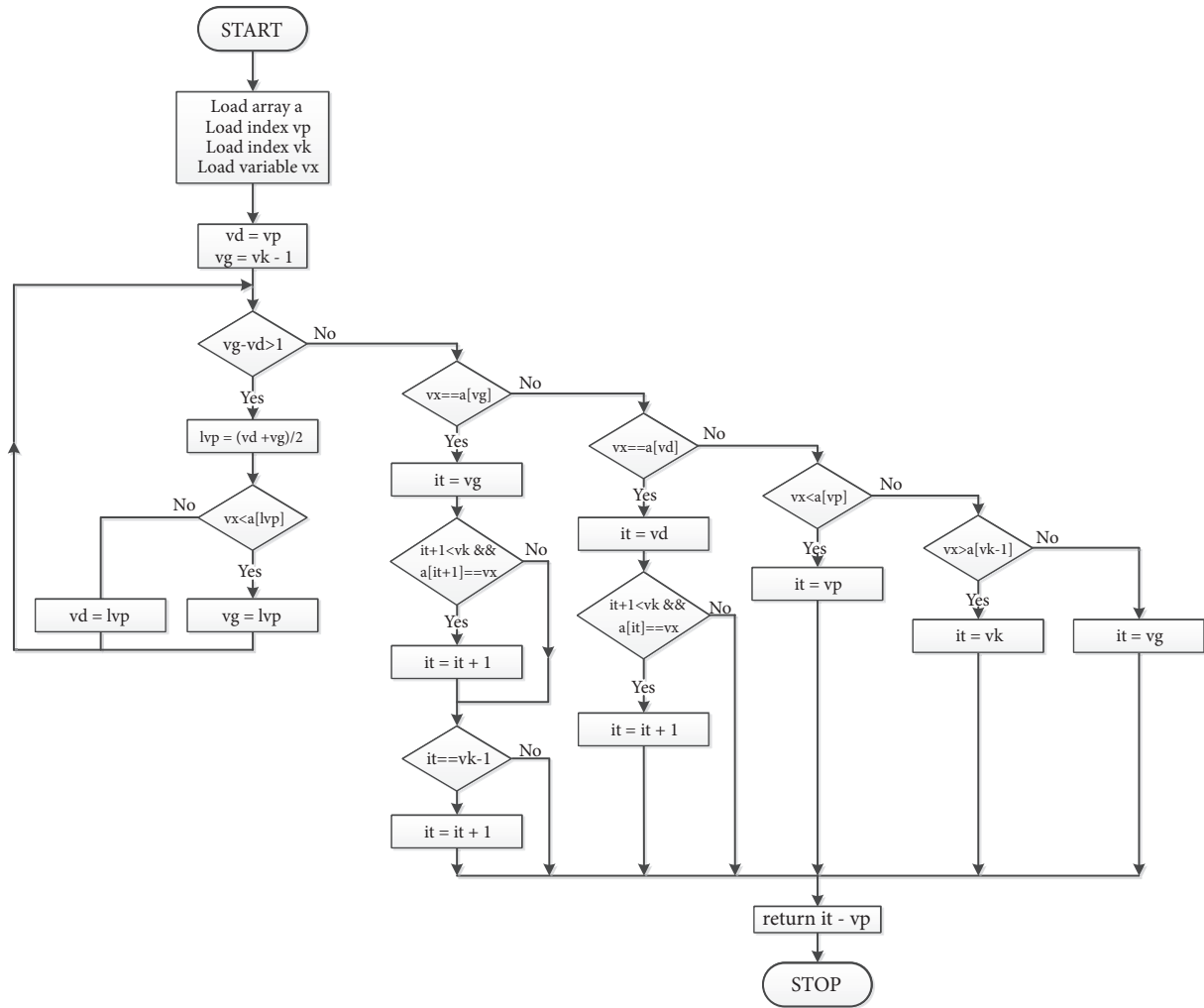


FIGURE 11: The block diagram of the function which returns index located in the left string.

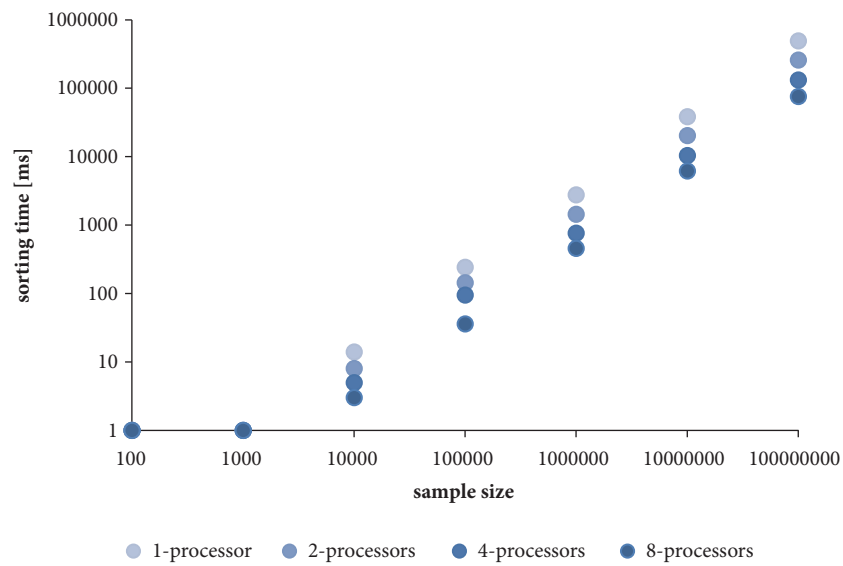


FIGURE 12: Comparison of benchmark time [ms].

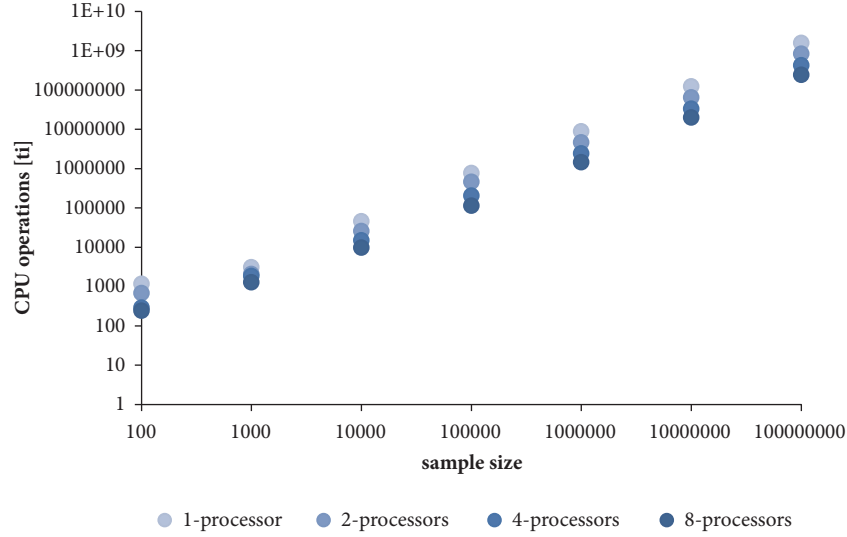


FIGURE 13: Comparison of benchmark operations [ti].

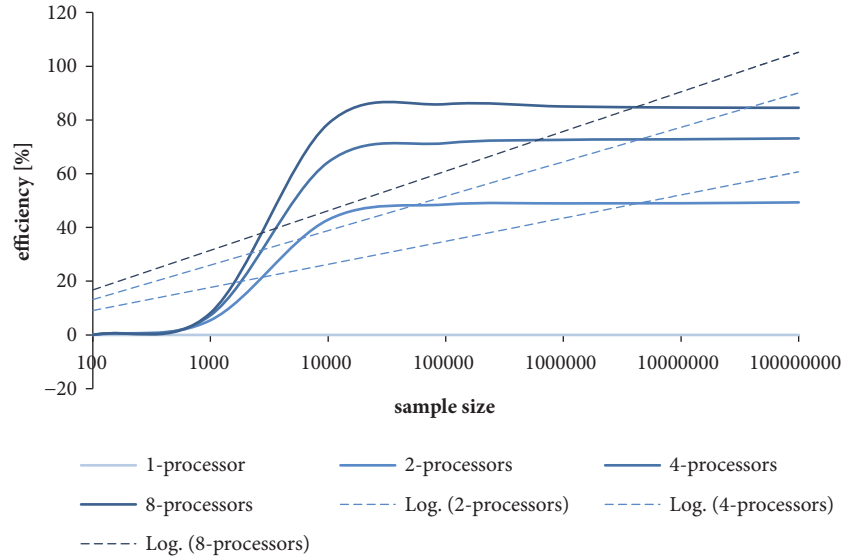


FIGURE 14: Comparison of the benchmark efficiency using multiple processors in terms of operation time [ms].

dashed trend lines confirm the stability of usage of computing powers by the proposed method. That shows possibility to improve sorting on multicore architectures. Due to proposed implementation the method is fully flexible; therefore, we can add a large number of processors to speed up sorting. This is very important for new computer architectures. In Figure 16 we can see a comparison of the proposed method to other sorting algorithms. As a baseline for comparisons was selected heap sort algorithm. We can see that FFPMS is performing much faster in comparison to other methods. The usage of additional processors makes improvements for large data sets. We can see a difference between using 8 and 12 processors in FFPMS, which is visible for sets above 1 000 000 elements. Other methods are much less efficient and might have deadlocks. For small sets quick sort (Woźniak et al. [44]) and classic merge (Woźniak et al. [46]) work very

similar. Proposed method is flexible, what means we can use a various number of processors. However it does not mean that the method is boosted the same with each new processor. There are some limitations visible also in our research. There exists significant improvement from 4 processors to 8 processors; however, from 8 processors to 12 processors this improvement is lower and in the above comparisons is visible for collections above 10 000 000 elements. Dashed lines for trends confirm numerical results. We can see that trend lines for FFPMS are growing which means that the proposed method shall be more efficient for big data sets. Similar grow in trend line is visible for classic merge; however, this method is about 15% less efficient. Trend line for quick sort is decreasing what shows that quick sort will be losing efficiency for big data sets. These benchmark tests confirm high potential of the proposed flexible parallel method.

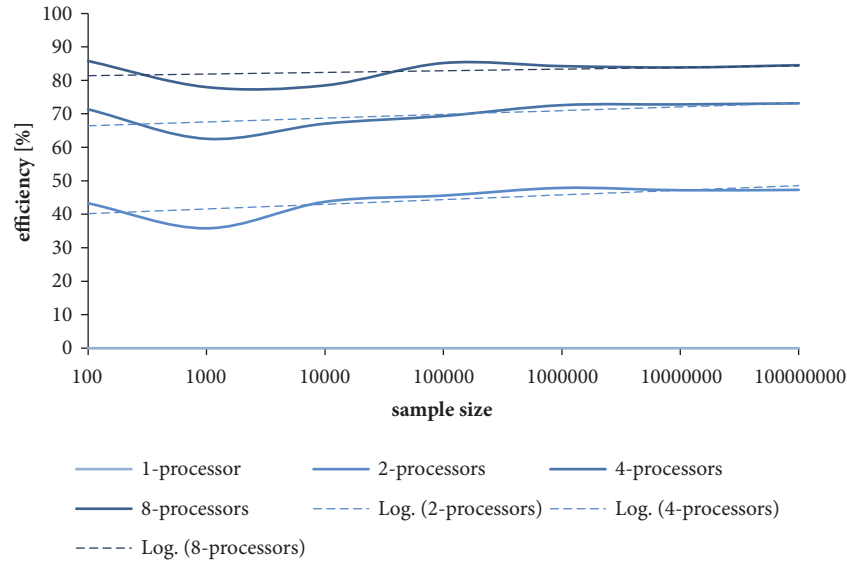


FIGURE 15: Comparison of the benchmark efficiency using multiple processors in terms of operations [ti].

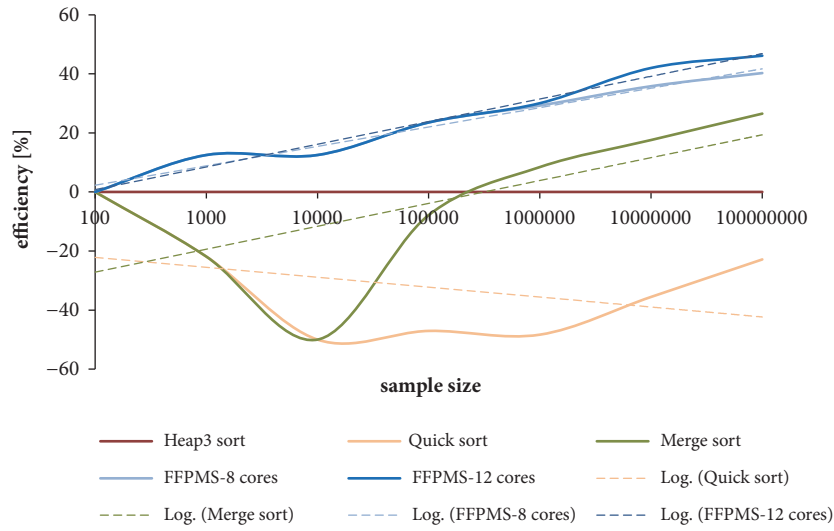


FIGURE 16: Benchmark comparison of the proposed Fully Flexible Parallel Merge Sort on 8 (light blue) and 12 (blue) cores to other sorting algorithms: heap sort with 3 divisions (brown), quick sort (cream), and classic merge sort (green) presented with logarithmic trend lines (dashed).

TABLE 1: Average benchmark sorting time for 100 samples in [ms].

Elements	1 - processor	2 - processors	4 - processors	8 - processors
100	1	1	1	1
1 000	1	1	1	1
10 000	14	8	5	3
100 000	241	124	69	36
1 000 000	2770	1414	759	415
10 000 000	38341	19552	10423	5892
100 000 000	491229	248920	131960	75985

TABLE 2: Average benchmark sorting operations for 100 samples in [ti].

Elements	1 - processor	2 - processors	4 - processors	8 - processors
100	1379	782	395	196
1 000	3097	1988	1160	682
10 000	45845	25795	15099	9863
100 000	775806	422189	237789	115017
1 000 000	8923418	4650899	2445417	1405403
10 000 000	123523898	65244951	33581034	19949470
100 000 000	1582604411	834171066	425138053	244802699

TABLE 3: Coefficient of variation for sorting time [ms].

Elements	1 - processor	2 - processors	4 - processors	8 - processors
100	0.3413121	0.2931151	0.3012821	0.3043211
1 000	0.2257182	0.1152201	0.1923175	0.1831273
10 000	0.1549049	0.1118317	0.2267786	0.1781741
100 000	0.1092983	0.1260794	0.2274779	0.0921129
1000 000	0.0809068	0.0836661	0.0826251	0.0717478
10000 000	0.0707954	0.0789099	0.0739682	0.0664342
100000 000	0.0687242	0.0737327	0.0721888	0.0632200

TABLE 4: Coefficient of variation for sorting operations [ti].

Elements	1 - processor	2 - processors	4 - processors	8 - processors
100	0.2490578	0.4343373	0.3066603	0.2368111
1 000	0.1389146	0.0654149	0.1422623	0.1645332
10 000	0.0856474	0.0501013	0.0694751	0.0769935
100 000	0.0903991	0.0959194	0.0853802	0.0600897
1000 000	0.0808740	0.0837383	0.0827343	0.0717861
10000 000	0.0707980	0.0789032	0.0739785	0.0664350
100000 000	0.0687246	0.0737334	0.0721900	0.0632179

**4.3. Conclusions.** The study shows that FFPMS operates in shorter time measuring tasks above 1 000 000. The proposed method is effective when using a large number of processors available in modern chip-sets. Its theoretical complexity is  $O(n(\log_2 n)^2/k)$ , where  $k$  is the number of logical processors that are used in calculations. Tests conducted on a limited number of threads fully confirm the theoretical results of the research, showing a clear improvement with each new additional processor.

Reduced sorting time is a big advantage for large data sets. Moreover the method does not have deadlocks. Proposed implementation gives a separation of concerns which makes it possible to freely enlarge the number of processors used for sorting. Other methods are less efficient. During tests merge sort, quick sort and heap sort gave results about 10% to 20% worse. From the research results we can conclude that merge sort is the only algorithm that can compete with FFPMS, but still these results are about 10% worse. At every step of the algorithm, the transformation of the arrays is mutually univocal and unique. Therefore, it will never happen that an element is inserted by the processor outside the last element of the array. In addition, each processor inserts exactly to one

memory cell and no other processor inserts into the same memory cell.

Due to the fact that the FFPMS algorithm is particularly effective in the case of merging large numerical sequences using a large number of processors, it seems to be purposeful to additionally combine it with the algorithm described in Marszałek [51]. In a hybrid sorting method designed in this way, the initial merging steps would be performed using Marszałek [51], and the final steps of merging long numerical sequences would be performed using the algorithm presented in this paper. The method constructed in such a way would enable the use of a large number of processors in entire sorting process in even more efficient manner. This will be one of the directions in our future research.

The complexity of the square from the logarithm of the dimension of the sorting task is asymptotically smaller than the element from the task's dimension. The comparison is presented in Table 5. That is why, even on one processor, we obtain sorting results for the 100 000 000 dimension in a very short time, and you cannot receive such results in real time for other sorting methods. In Table 5 we have comparison to other algorithms. We can see that Grover algorithm

TABLE 5: Comparison of complexity for sorting operations.

Elements	$\lceil \sqrt{n} \rceil$	$\lceil \log_2 n \rceil$	$\lceil \log_2^2 n \rceil$
100	10	7	49
1 000	32	10	100
10 000	100	14	196
100 000	317	17	289
1000 000	1 000	20	400
10000 000	3163	24	576
100000 000	10 000	27	729

on quantum computer (represented in  $\lceil \sqrt{n} \rceil$  complexity) is much slower, and similarly we see comparison to binary search algorithm complexity (represented in  $\lceil \log_2 n \rceil$  complexity). Our method (represented in  $\lceil \log_2^2 n \rceil$  complexity) is very efficient, and let us sort inputs even faster than on quantum computer. Comparing to some other classic methods like bubble sorting method (complexity  $n^2$ ) we see that the proposed algorithm shows very high boost.

In the implementation of the presented sorting algorithm was used the C# class *System.Threading.Tasks*. This class has a parallel for loop, which is not an iterative in classic understanding, but an iteration assigns tasks to the processes which are computed for each of the processors. Subsequent processes receive the identifier of consecutive natural numbers starting with zero and ending with one less than the number of processes. According to the object-oriented programming principle, variables declared inside this loop are only available for a given thread, and arrays and variables declared outside the parallel loop are available for all processes. Synchronization of all processes occurs after the parallel loop ends. A simple rule applies here: all processes can read the arrays and global variables, but only one of them can write to the same memory location. While transferring data between memory and processes the cache compartment is done automatically and the programmer is not affected. Therefore proposed implementation is very complex and makes the program oriented on maximum efficiency for the parallelization of sorting processes.

The novelty of the presented algorithm is the way of dividing tasks so that the processes do not interfere with each other. Each of the processes inserts the specified item into the merged string only on the basis of the information, and there are no parallel loop insertions in the same memory cell. This division of work in processes increases the computational complexity of the algorithm with a small number of processors. At the same time it increases the sorting performance of servers with a large number of processors and large operating memory (such as Oracle SPARC M8-8). The SPARC M8-8 is equipped with 256 cores with 8 interleaves, a total of 2048 logical processors and 76TB of memory. For this type of computer the proposed method will be a perfect solution, which can independently operate on each of the processors without cross actions. Therefore the efficiency of our method will be much higher. However our method also fits Microsoft software processors manufacturer like Intel and AMD, which were used in presented tests. The method strives

to match any number of processors and can be implemented actually in any programming language. In our tests, according to the available licenses for the software and hardware testing, the algorithm was performed in C# language under Visual Studio. There should be no problem to rewrite the implementation of the proposed algorithm for Oracle SPARC M8 server in Java. Oracles Java is an object-oriented language which has rich library for handling multithreaded algorithms and applications. The results of statistical studies presented in this paper guarantee the reproducibility of the obtained results on other servers.

From the research have arisen some interesting open questions. In the tests we tried to present the method working on various numbers of processors. We assumed that there were no delays from hardware, bus of the motherboard, hard disk connectors, etc. However these aspects shall be verified in our future research. We also plan to concentrate on some more sophisticated data structures which can additionally support data management in the system. It will be also interesting to develop devoted versions of the presented algorithm for specialized management systems used in medicine, geoscience, financial systems, etc.

The algorithm presented in the paper has been designed for a typical architecture that corresponds to the modern computers equipped with multithreaded cores with a fast cooperative memory. This corresponds to the Uniform Memory Access (UMA) model, and in principle there is no difference in performance relative to the CREW PRAM model. The computer network on the PRAM model is similar in actions to the data bus through which the processors have access to the entire memory. Of course, it is possible to distribute calculations on many computers over the Internet which is done in the UMA model. However, this would require additional research on the possibility of applying the presented idea to distributed systems. The No-Remote Memory Access (NoRMA) architecture model is very complex and does not correspond to the processing of data on modern computers, so it was not considered in this work. The question of the possibility of using the proposed method for GPUs remains open. General-Purpose computing on Graphics Processing Unit (GPGPU) shows the computing power used by graphic cards where the processor cores work together. However, it is important to note that the processors used there are low in computing power and that the graphic card's performance is decisive in principle. An additional aspect is the operating memory that the graphic chip-set has. For modern architectures we can talk about very efficient memory in terms of data capacity and access speed. In the case of graphic cards, the processors and memory used in them are much less efficient. Thus, for large data sets, there may be situations in which such systems will not sufficiently cope with processing input data, and thus the sorting methods may lose a significant portion of performance. Therefore, in the assumption of the developed method we use the concept of the main processor from motherboard as appropriate for the presented method of computing.

An interesting research conclusion would be also in relation to quantum computing. The current state of knowledge



indicates that the best results for quantum computing are achieved only theoretically and we still have no real quantum machines. The reason is unstable energy allocation in quantum machines. According to our knowledge there is no such algorithm for current computers, which can show comparable results to quantum ones. However, in this work we have shown that our algorithm is able to match quantum efficiency (which exists only in theory at the present time), but as shown here it works in reality on real multicore architectures. This is very important feature of our method.

## 5. Final Remarks

The article presents a Fully Flexible Parallel Merge Sort Algorithm which is composed to preserve high performance in sorting at the lowest possible computational complexity.

Presented in this article is a method for an effective way to organize large amounts of data using a number of processors. The method was developed using separation of concerns; therefore, there are no cross actions or interferences between processors. The proposed model is fully flexible for various number of processors. The tests confirmed the theoretical computational complexity and the stability of the algorithm. Moreover, the achievement of an operational time equals to  $O(\sqrt{N})$ , which is an interesting innovation considering that no better result was yet achieved by any sort algorithm on classic computers as well as in quantum theory. An example is the Grover algorithm, which is considered to be the best one because it can execute  $O(\sqrt{N})$  queries, while our proposition has similar result. It is worth noting that the current hardware state does not give the possibility to implement the Grover algorithm on the dedicated machine, which makes it still only theoretical proposition. Unlike the method presented in this work, it is implemented and flexible to a growing number of operational cores.

## Data Availability

The article presents a method which can be used to manage data in computer systems so we do not have any special data set to report.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

Authors acknowledge contribution to this project from the Rector of the Silesian University of Technology under grant for perspective professors no. 09/010/RGH18/0035.

## References

- [1] K. Bochenina, S. Kesarev, and A. Boukhanovsky, "Scalable parallel simulation of dynamical processes on large stochastic Kronecker graphs," *Future Generation Computer Systems*, vol. 78, pp. 502–515, 2018.
- [2] P. Czarnul, J. Kuchta, M. Matuszek et al., "MERPSYS: An environment for simulation of parallel application execution on large scale HPC systems," *Simulation Modelling Practice and Theory*, vol. 77, pp. 124–140, 2017.
- [3] H. Mora, D. Gil, R. M. Terol, J. Azorín, and J. Szymanski, "An IoT-Based Computational Framework for Healthcare Monitoring in Mobile Environments," *Sensors*, vol. 17, no. 10, p. 2302, 2017.
- [4] A. Esmaeili, N. Mozayani, M. R. Jahed Motlagh, and E. T. Matson, "A socially-based distributed self-organizing algorithm for holonic multi-agent systems: Case study in a task environment," *Cognitive Systems Research*, vol. 43, pp. 21–44, 2017.
- [5] L. Stanescu, M. Brezovan, and D. D. Burdescu, "Automatic mapping of MySQL databases to NoSQL MongoDB," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016*, pp. 837–840, Poland, September 2016.
- [6] M. Janetschek, R. Prodan, and S. Benedict, "A workflow runtime environment for manycore parallel architectures," *Future Generation Computer Systems*, vol. 75, pp. 330–347, 2017.
- [7] V. A. E. Farias, F. R. C. Sousa, J. G. R. Maia, J. P. P. Gomes, and J. C. Machado, "Regression based performance modeling and provisioning for NoSQL cloud databases," *Future Generation Computer Systems*, vol. 79, pp. 72–81, 2018.
- [8] M. T. González-Aparicio, M. Younas, J. Tuya, and R. Casado, "Testing of transactional services in NoSQL key-value databases," *Future Generation Computer Systems*, vol. 80, pp. 384–399, 2018.
- [9] A. V. Aho and J. E. Hopcroft, *The design and analysis of computer algorithms*, Pearson Education India, 1974.
- [10] D. E. Knuth, *The art of computer programming. Vol. 3*, Addison-Wesley, Reading, MA, 1998.
- [11] H. Bing-Chao and D. E. Knuth, "A one-way, stackless quicksort algorithm," *BIT. Nordisk Tidskrift for Informationsbehandling (BIT)*, vol. 26, no. 1, pp. 127–130, 1986.
- [12] R. S. Francis and L. J. H. Pannan, "A parallel partition for enhanced parallel QuickSort," *Parallel Computing*, vol. 18, no. 5, pp. 543–550, 1992.
- [13] A. Rauh and G. R. Arce, "A fast weighted median algorithm based on Quickselect," in *Proceedings of the 2010 17th IEEE International Conference on Image Processing, ICIP 2010*, pp. 105–108, Hong Kong, September 2010.
- [14] P. Tsigas and Y. Zhang, "A simple, fast parallel implementation of Quicksort and its performance evaluation on SUN Enterprise 10000," in *Proceedings of the 11th Euromicro Conference on Parallel, Distributed and Network-Based Processing, Euro-PDP 2003*, pp. 372–381, Italy, February 2003.
- [15] A. M. Daoud, H. Abdel-Jaber, and J. Ababneh, "Efficient non-quadratic quick sort (NQQuickSort)," *Communications in Computer and Information Science*, vol. 194, pp. 667–675, 2011.
- [16] J. Edmondson, "M pivot sort—replacing quick sort presenter," in *Proceedings of the James edmondson conference: Amcs05–2005 world congress in applied computing*, 2005.
- [17] S. Kushagra, A. López-Ortiz, J. I. Munro, and A. Qiao, "Multi-pivot quicksort: Theory and experiments in," in *Proceedings of the Meeting on Algorithm Engineering & Experiments*, pp. 47–60, 2014.
- [18] M. Ben-Or, "lower bounds for algebraic computation trees," in *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pp. 80–86.

- [19] E.-E. Doberkat, "Inserting a new element into a heap," *BIT: Nordisk Tidsskrift for Informationsbehandling*, vol. 21, no. 3, pp. 255–269, 1981.
- [20] L. M. Wegner and J. I. Teuhola, "The External Heapsort," *IEEE Transactions on Software Engineering*, vol. 15, no. 7, pp. 917–925, 1989.
- [21] S. Sumathi, A. M. Prasad, and V. Suma, "Optimized Heap Sort Technique (OHS) to Enhance the Performance of the Heap Sort by Using Two-Swap Method," in *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, vol. 327 of *Advances in Intelligent Systems and Computing*, pp. 693–700, Springer International Publishing, Cham, 2015.
- [22] S. Roura, "Digital access to comparison-based tree data structures and algorithms," *Journal of Algorithms in Cognition, Informatics and Logic*, vol. 40, no. 1, pp. 1–23, 2001.
- [23] K. Abrahamson, N. Dadoun, D. G. Kirkpatrick, and T. Przytycka, "A simple parallel tree contraction algorithm," *Journal of Algorithms in Cognition, Informatics and Logic*, vol. 10, no. 2, pp. 287–302, 1989.
- [24] S. Carlsson, C. Levcopoulos, and O. Petersson, "Sublinear merging and natural mergesort," *Algorithmica. An International Journal in Computer Science*, vol. 9, no. 6, pp. 629–648, 1993.
- [25] R. Cole, "Parallel merge sort," *SIAM Journal on Computing*, vol. 17, no. 4, pp. 770–785, 1988.
- [26] G. Gediga and I. Düntsch, "Approximation quality for sorting rules," *Computational Statistics & Data Analysis*, vol. 40, no. 3, pp. 499–526, 2002.
- [27] J. D. Harris, "Sorting unsorted and partially sorted lists using the natural merge sort," *Software: Practice and Experience*, vol. 11, no. 12, pp. 1339–1340, 1981.
- [28] B. Salzberg, "Merging sorted runs using large main memory," *Acta Informatica*, vol. 27, no. 3, pp. 195–215, 1989.
- [29] B.-C. Huang and M. A. Langston, "Practical In-Place Merging," *Communications of the ACM*, vol. 31, no. 3, pp. 348–352, 1988.
- [30] L. Zheng and P.-Å. Larson, "Speeding up external mergesort," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 2, pp. 322–332, 1996.
- [31] W. Zhang and P. A. Larson, "Dynamic memory adjustment for external mergesort," in *VLDB, Citeseer*, pp. 25–29, 1997.
- [32] W. Zhang and P. A. Larson, "Buffering and read-ahead strategies for external mergesort," in *The VLDB Journal*, pp. 523–533, 1998.
- [33] R. Vignesh and T. Pradhan, "Merge sort enhanced in place sorting algorithm," in *Proceedings of the 2016 International Conference on Advanced Communication Control and Computing Technologies, ICACCT 2016*, pp. 698–704, India, May 2016.
- [34] S. M. Cheema, N. Sarwar, and F. Yousaf, "Contrastive analysis of bubble & merge sort proposing hybrid approach," in *Proceedings of the 6th International Conference on Innovative Computing Technology, INTECH 2016*, pp. 371–375, Ireland, August 2016.
- [35] S. Paira, S. Chandra, and S. K. S. Alam, "Enhanced Merge Sort-A New Approach to the Merging Process," in *Proceedings of the 6th International Conference On Advances In Computing and Communications, ICACC 2016*, pp. 982–987, India, September 2016.
- [36] T. O. Alanko, H. H. A. Erkio, and I. J. Haikala, "Virtual Memory Behavior of Some Sorting Algorithms," *IEEE Transactions on Software Engineering*, vol. SE-10, no. 4, pp. 422–431, 1984.
- [37] P.-Å. Larson and G. Graefe, "Memory management during run generation in external sorting," *SIGMOD Record*, vol. 27, no. 2, pp. 472–483, 1998.
- [38] A. LaMarca and R. E. Ladner, "The influence of caches on the performance of sorting," *Journal of Algorithms in Cognition, Informatics and Logic*, vol. 31, no. 1, pp. 66–104, 1999.
- [39] P. Crescenzi, R. Grossi, and G. F. Italiano, "Search data structures for skewed strings," in *Experimental and efficient algorithms*, vol. 2647 of *Lecture Notes in Comput. Sci.*, pp. 81–96, Springer, Berlin, 2003.
- [40] V. Estivill-Castro and D. Wood, "A survey of adaptive sorting algorithms," *ACM Computing Surveys*, vol. 24, no. 4, pp. 441–476, 1992.
- [41] M. Axtmann, T. Bingmann, P. Sanders, and C. Schulz, "Practical massively parallel sorting," in *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2015*, pp. 13–23, USA, June 2015.
- [42] S. Abdel-Hafeez and A. Gordon-Ross, "An efficient  $O(N)$  comparison-free sorting algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 6, pp. 1930–1942, 2017.
- [43] A. S. Mohammed, Ş. E. Amrahov, and F. V. Çelebi, "Bidirectional Conditional Insertion Sort algorithm; An efficient progress on the classical insertion sort," *Future Generation Computer Systems*, vol. 71, pp. 102–112, 2017.
- [44] M. Woźniak, Z. Marszałek, M. Gabryel, and R. K. Nowicki, "Preprocessing Large Data Sets by the Use of Quick Sort Algorithm," in *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*, vol. 364 of *Advances in Intelligent Systems and Computing*, pp. 111–121, Springer International Publishing, Cham, 2016.
- [45] M. Woźniak, Z. Marszałek, M. Gabryel, and R. Nowicki, "Triple heap sort algorithm for large data sets," *Looking into the Future of Creativity and Decision Support Systems*, pp. 657–665, 2013.
- [46] M. Woźniak, Z. Marszałek, M. Gabryel, and R. K. Nowicki, "Modified Merge Sort Algorithm for Large Scale Data Sets," in *Artificial Intelligence and Soft Computing*, vol. 7895 of *Lecture Notes in Computer Science*, pp. 612–622, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [47] Z. Marszałek, M. Woźniak, G. Borowik et al., "Benchmark Tests on Improved Merge for Big Data Processing," in *Proceedings of the Asia-Pacific Conference on Computer-Aided System Engineering, APCASE 2015*, pp. 96–101, Ecuador, July 2015.
- [48] D. Czerwinski, "Digital Filter Implementation in Hadoop Data Mining System," in *Computer Networks*, vol. 522 of *Communications in Computer and Information Science*, pp. 410–420, Springer International Publishing, Cham, 2015.
- [49] A. Uyar, "Parallel merge sort with double merging," in *Proceedings of the 8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014*, Kazakhstan, October 2014.
- [50] Z. Marszałek, "Novel Recursive Fast Sort Algorithm," in *Information and Software Technologies*, vol. 639 of *Communications in Computer and Information Science*, pp. 344–355, Springer International Publishing, Cham, 2016.
- [51] Z. Marszałek, "Parallelization of modified merge sort algorithm," *Symmetry*, vol. 9, no. 9, 2017.

## Research Article

# Application of the Variable Precision Rough Sets Model to Estimate the Outlier Probability of Each Element

Francisco Maciá Pérez<sup>1</sup>,<sup>1</sup> Jose Vicente Berna Martienz<sup>1</sup>,<sup>1</sup> Alberto Fernández Oliva,<sup>2</sup>  
and Miguel Abreu Ortega<sup>3</sup>

<sup>1</sup>Computer Science & Technology Department, University of Alicante, Spain

<sup>2</sup>Department of Computer Science, Faculty of Mathematics and Computer Science, University of Havana, Cuba

<sup>3</sup>MSc Student at Georgia Institute of Technology, USA

Correspondence should be addressed to Jose Vicente Berna Martienz; [jvberna@ua.es](mailto:jvberna@ua.es)

Received 26 April 2018; Accepted 2 September 2018; Published 8 October 2018

Guest Editor: Magnus Johnsson

Copyright © 2018 Francisco Maciá Pérez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In a data mining process, outlier detection aims to use the high marginality of these elements to identify them by measuring their degree of deviation from representative patterns, thereby yielding relevant knowledge. Whereas rough sets (RS) theory has been applied to the field of knowledge discovery in databases (KDD) since its formulation in the 1980s; in recent years, outlier detection has been increasingly regarded as a KDD process with its own usefulness. The application of RS theory as a basis to characterise and detect outliers is a novel approach with great theoretical relevance and practical applicability. However, algorithms whose spatial and temporal complexity allows their application to realistic scenarios involving vast amounts of data and requiring very fast responses are difficult to develop. This study presents a theoretical framework based on a generalisation of RS theory, termed the variable precision rough sets model (VPRS), which allows the establishment of a stochastic approach to solving the problem of assessing whether a given element is an outlier within a specific universe of data. An algorithm derived from quasi-linearisation is developed based on this theoretical framework, thus enabling its application to large volumes of data. The experiments conducted demonstrate the feasibility of the proposed algorithm, whose usefulness is contextualised by comparison to different algorithms analysed in the literature.

## 1. Introduction

Outlier detection is an area of increasing relevance within the more general data mining process. Outliers may highlight extremely important findings in a wide range of applications: fraud detection, detection of illegal access to corporate networks, and detection of errors in input data, among others.

The rough sets basic model created by Pawlak [1] is a model with a simple and solid mathematical basis: the equivalence relation theory, which enables the description of partitions consisting of classes of indiscernible objects. The rough sets (RS) rationale consists of approximating a set using a pair of sets, termed lower and upper approximations. In general, the RS approach is based on the ability to classify data collected through various means. In recent years, this model has been successfully applied in various contexts

[2–4]. Therefore, its study has attracted the attention of the international scientific community, especially regarding solving problems that involve establishing relationships between data.

An outlier detection method is proposed in [5], which is the first Pawlak rough sets application to this problem. However, its computational implementation is complicated by its exponential order. An extension of the theoretical framework of the previous proposition is presented in [6], in which an outlier detection algorithm is implemented based on Pawlak rough sets—the Pawlak rough sets algorithm—with a nonexponential order of temporal and spatial complexity. In [6], a method for the detection of outliers has been proposed with a simple and rigorous theoretical setup, starting from a definition of outliers that is simple, intuitive, and computationally viable for large

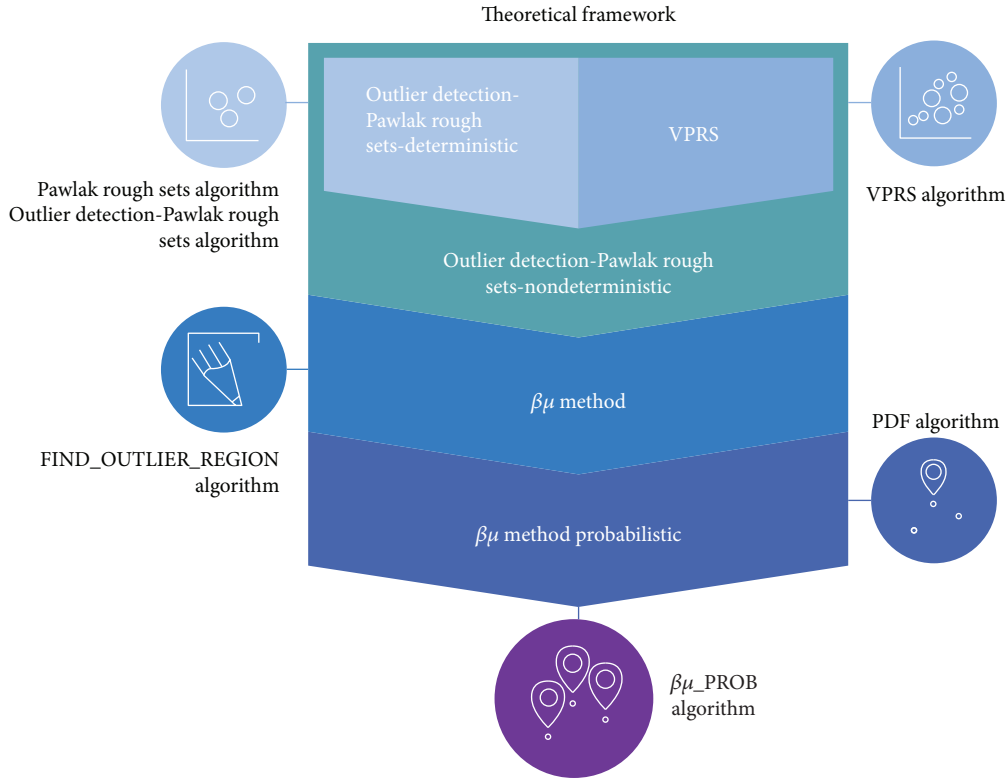


FIGURE 1: Global view of the theoretical framework.

datasets. From this method, an efficient algorithm for outlier mining has been developed, conceptually based on a novel and original approach using rough set theory, which has not been applied in any previous category of classification for the methods of rough set detection. The proposed algorithm is linear with respect to the cardinality of the data universe over which it is applied, and it is quadratic with respect to the number of equivalence relations used to describe the universe. However, this number of relations merely represents a constant, as it is usually significantly smaller than the cardinality of the universe in question. In contrast to many other methods that present difficulties in their application depending on the nature of the data to be analyzed, our proposal is applicable to both continuous and discrete data. The possibility that the datasets may contain a mix of attribute types (e.g., a mix of continuous and categorical attributes) does not present a limitation for the applicability of the proposed algorithm. Nevertheless, this result has the drawback for our purposes of inheriting the deterministic nature of the Pawlak rough sets regarding the classification.

The variable precision rough sets model (VPRS) [7] is a generalisation of the Pawlak rough sets that rectifies its deterministic nature through a new concept of inclusion of standard sets: the inclusion of majority sets [8, 9], which makes it possible to incorporate user-defined thresholds. A computationally viable algorithm for the nondeterministic detection of outliers, termed the VPRS algorithm, based on the VPRS, which was in turn based on the theoretical framework provided by Pawlak rough sets and VPRS,

termed *nondeterministic outlier detection-Pawlak rough sets* (Figure 1), is presented in [10]. Figure 1 shows a global view of the theoretical framework for the formalisation of a computationally viable algorithm for unsupervised probabilistic estimation of the outlier condition of each element of a given universe of data used in this paper.

The Pawlak rough sets and VPRS algorithms solve the following problem: “to determine the set of outliers of a given universe of data from a preset exceptionality threshold ( $\mu$ ) defined in [6] at a given allowed classification error ( $\beta$ ) defined by [7].”

In this paper, a new approach to the problem of outlier detection that solves the limitations of the aforementioned results is proposed: to preset the thresholds and to develop scalable algorithms independent of the context and nature of the problem. Therefore, the aim of this research may be summarised as follows: “to create a computationally viable method that calculates the outlier probability of each element from a given universe of data without the need to establish preconditions—that is, the determination of the thresholds ( $\mu, \beta$ ) of the analysis—that depend on each specific context to which the algorithm is applied.”

The starting hypothesis is summarised as follows: “a new theory may be developed by extending the basic concepts and the formal tools provided by RS theory [1, 11] and VPRS [7], applied to the outlier detection problem, which allows the unsupervised determination, for each element of a universe of data, of the region of threshold values ( $\mu, \beta$ ) in which such element is an outlier.” Based on this approach, which was termed the  $\beta\mu$  Method (see Figure 1), “the outlier probability



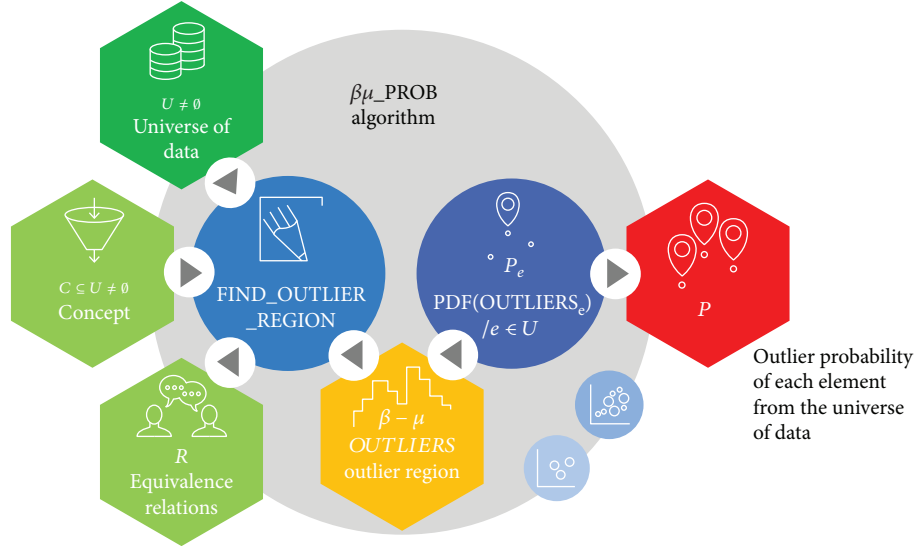


FIGURE 2: General outline of the proposed solution.

of each element from the universe of data can be determined.” This new method is termed the *Probabilistic  $\beta\mu$  Method* (see Figure 1).

To develop the method proposed in the research objective as a solution (see Figure 2), the theoretical framework developed in [6, 10] is expanded based on conceptual elements of the Pawlak rough sets and VPRS and on the theoretical proposition of [5]. Combined, they make it possible to formally demonstrate the theoretical elements proposed in the new concept of the method and serve as a reference framework to design and implement a computationally viable algorithm that validates the starting hypothesis. This algorithm has been termed the  $\beta\mu\_PROB$  algorithm, as can be seen in Figure 2. This figure shows a general outline of the proposed solution, specified in the implementation of a computationally viable algorithm ( $\beta\mu\_PROB$  Algorithm) for the unsupervised probabilistic estimation of the outlier condition of each element from a universe of data, entirely based on the development of the theoretical framework created in this research study.

Based on the above, the text below is divided into four sections. In Section 2, a theoretical framework termed  $\beta\mu$  Method (Figure 1) is proposed alongside an algorithm that determines the outlier region of each element from the universe of data, termed the *FIND\_OUTLIER\_REGION* Algorithm (Figure 2). In Section 3, new theoretical elements collected using a method termed *Probabilistic  $\beta\mu$  Method* (Figure 1) are proposed, and statistical techniques that make it possible to solve the problem posed are applied by proposing the  $\beta\mu\_PROB$  algorithm (Figure 2), which determines the outlier probability of each element from the universe of data within such universe. In Section 4, the experiments that validate the proposed solution are designed, the findings are analysed, and the algorithms based on RS and the classical algorithms, in addition to the different RS algorithms that have been developed to achieve the final solution, are compared. In Section 5, the conclusions from this research study

are presented, and some perspectives and future studies continuing this research are considered.

## 2. Outlier Region

In essence, the entire proposal in this article is summarized in the following two phases:

- (i) In the first, it is determined for each element  $e$  of the finite universe  $U$ , under what conditions (threshold of exceptionality  $\mu$  and classification error allowed  $\beta$ ) that element behaves as an exceptional element (outlier). These conditions ( $\mu$  and  $\beta$ ) establish an  $R$  region within which the element is considered outlier
- (ii) In the second phase, taking into account the determined  $R$  region, for each element of the finite universe  $U$ , the probability of each of them being an outlier in  $U$  is calculated using statistical techniques

To solve the problem, first, we expanded the theoretical framework defined in [6, 10] (Section 2.1). This framework is based on a method that we have termed the  $\beta\mu$  Method. The method provides the formal tools that, second, make it possible to develop a computationally efficient algorithm to solve the problem, which we have termed the *FIND\_OUTLIER\_REGION* algorithm (Section 2.2).

**2.1. Theoretical Framework:  $\beta\mu$  Method.** The  $\beta\mu$  Method consists of three main tasks that can be easily differentiated: (a) to determine the outlier region in relation to threshold  $\beta$ , which makes it possible to calculate the allowable classification error, (b) to determine the outlier region in relation to threshold  $\mu$ , that is, to calculate the preset outlier threshold, and (c) to integrate both specific solutions to determine the outlier region ( $\beta, \mu$ ) of each element from the universe of data. Below, we detail each of these tasks.



**2.1.1. Outlier Region in Relation to  $\beta$ .** To determine the outlier region in relation to the set of values of  $\beta$  (referred to as the allowable  $\beta$ -error in the classification), three specific subproblems are solved.

**Subproblem 1:** to determine the range of  $\beta$  values for which  $B_i \subseteq B_j$ ,  $i \neq j$ ,  $1 \leq i, j \leq m$ .  $B_i, B_j$ : internal borders with respect to equivalence relations  $i$  and  $j$ , where  $m$  is the total number of equivalence relations taken into account in the analysis. Based on the theoretical framework described in [6], it is known that if no internal border  $B_i$  is a subset of another internal border  $B_j$ , then all  $B_j$  elements are candidates for outliers in the dataset or universe of data,  $U$ . Therefore, the problem is restated as follows: to determine the set of  $\beta$  values for which an internal border  $B_i$ ,  $i \neq j$ , is a subset of the internal border  $B_j$ , that is,  $B_i \subseteq B_j$ . After calculating this set,  $\forall i \neq j$ ,  $1 \leq i \leq m$ , then the complement of the union of all ranges of  $\beta$  values calculated will be the set of values, in relation to such threshold, for which all  $B_j$  elements are candidates for outliers.

**Subproblem 2:** to determine the range of  $\beta$  values for which a given internal border is null. Similarly, in the theoretical framework on which the detection method is based, it is assumed that the internal borders considered in the analysis are not null. Accordingly, the  $\beta$  values for which this condition is met are determined. The analysis is performed for any internal border  $B_i$ , and subsequently, this result is generalised to any other internal border through a similar analysis.

**Subproblem 3:** to determine the set of  $\beta$  values for which  $B_i = B_j$ ,  $i \neq j$ ,  $1 \leq i, j \leq m$ . In the theoretical framework on which the detection method is based, the existence of two equal internal borders is not considered either, thereby requiring determining the set of  $\beta$  values for which this condition is met. In this case, the problem consists of determining the set of  $\beta$  values for which  $B_i = B_j$ ,  $i \neq j$ ,  $1 \leq i, j \leq m$ , which is easily deduced through the following sequence of equivalences:  $B_i = B_j \Leftrightarrow B_i \subseteq B_j \wedge B_j \subseteq B_i \Leftrightarrow \beta \in I_{ij} \wedge \beta \in I_{ji} \Leftrightarrow \beta \in I_{ij} \cap I_{ji}$ . From these, we can conclude that the set of  $\beta$  values for which  $B_i = B_j$ ,  $i \neq j$ ,  $1 \leq i, j \leq m$ , is  $EQ_{ij} = \{\beta : \beta \in I_{ij} \cap I_{ji}\}$ , in which  $I_{ij}$  is the set of  $\beta$  values for which  $B_i \subseteq B_j$ ,  $i \neq j$ ,  $1 \leq i, j \leq m$ .

After concluding the analysis of the three proposed subproblems, from the sequence of sets, a general criterion can be established defining when an internal border is a subset of another.

$A$ : set of  $\beta$  values for which a nonempty internal border exists, which is a specific subset of the internal border  $j$ .  $(I_{1j} - EQ_{1j} - N_1) \cup (I_{2j} - EQ_{2j} - N_2) \cup \dots \cup (I_{mj} - EQ_{mj} - N_m) = A$ , where  $N_i$ : set of  $\beta$  values for which  $B_i = \phi$ ,  $1 \leq i \leq m$ .

$A^c$ : set of  $\beta$  values for which no nonempty internal border is a specific subset of the internal border  $j$ .

$S_j$ : set of  $\beta$  values for which no nonempty internal border is a specific subset of the internal border  $j$  excluding the values for which such border is empty.  $S_j = A^c - N_j$ .

Considering that for all  $B_j$  elements to be outliers, the condition that no other internal border is a subset of this

border must be met; the previous results suggest that this only occurs when  $\beta \in S_j$ . Therefore,  $S_j$  is the range of  $\beta$  values for which an element  $e$  from the universe of data  $U$ ,  $e \in B_j$ , belongs to some nonredundant outlier set, and thus  $e$  is a possible outlier.

**2.1.2. Outlier Region in Relation to  $\mu$ .** The next step is to perform a similar analysis to determine the set of outlier threshold values  $\mu$  for which each element from the universe of data may be considered an outlier. The problem is now the following: given an element  $e \in U$ , to determine the range of values of the threshold  $\mu$  for which the outlier degree of  $e$  is higher than that of  $\mu$ . The theoretical elements necessary to solve this problem are presented below according to the following logical sequence:

- (i) To define the set of values of  $\beta$  for which  $\forall e : e \in U$  belongs to internal border  $B_i$ ,  $1 \leq i \leq m$
- (ii) To establish a new definition of outlier degree  $\forall e : e \in U$ , in a new interpretation of the values of  $\beta$ :  $\text{ExcepDegree}(e, \beta)$
- (iii) To determine  $\forall e \in U$  the range of values of  $\mu$  for which  $\text{ExcepDegree}(e, \beta) \geq \mu$  for a given  $\beta$  value

Following this sequence, first, the set of  $\beta$  values for which  $e \in U$  belongs to the internal border  $B_i$ ,  $1 \leq i \leq m$ , is defined.

**Definition 1.** Let  $U$  be a universe of data,  $X$  the subset of values of  $U$  that meet a specific concept,  $\forall e \in U$ ,  $1 \leq i \leq m$ , and  $EC$  an equivalence class of the partition induced by the equivalence relation  $r_i$  in  $U$  such that  $e \in EC$ . The set of values of  $\beta$  for which  $e$  belongs to the internal border  $B_i$  is defined as follows:

$$M_i(e) = \begin{cases} \beta : \beta < c(EC, X) < 1 - \beta, & \text{if } e \in X, \\ \emptyset, & \text{if } e \notin X, \end{cases} \quad (1)$$

wherein  $c(A, B)$  is the measure of the degree of declassification of set  $A$  in relation to set  $B$ , that is, the relative error of classification of a set of objects, defined in the VPRS [7] as follows:

$$c(A, B) = \begin{cases} 1 - \frac{|A \cap B|}{|A|}, & \text{if } |A| \neq 0, \\ 0, & \text{if } |A| = 0. \end{cases} \quad (2)$$

As established by  $M_i(e)$ , the values of parameter  $\beta$  must meet the following restrictions to ensure that  $e$  belongs to the internal border  $B_i$ :  $\beta < c(EC, X) < 1 - \beta \Rightarrow [\beta < 1 - c(EC, X)] \wedge [\beta < c(EC, X)]$ . Therefore, the following range of  $\beta$  values within which  $e \in B_i : \forall \beta : \beta \in [0, \min(-c(EC, X), 1 - c(EC, X))]$  can be established from  $M_i(e)$ . This result satisfies the criterion required to state that an

element  $e \in U$  may be an outlier candidate. In this case, this means that it belongs to some internal border. Accordingly, below, a new definition of outlier degree of an element  $e \in U$  is established, with a new interpretation: its dependence on the values of  $\beta$ . Preliminarily, a new definition and a new proposition must be established based on that dependence.

**Definition 2.**  $\forall e \in U, 1 \leq i \leq m$

$$\lambda_i(e) = \begin{cases} \text{Sup}(M_i(e)), & \text{if } M_i(e) \neq \emptyset, \\ 0, & \text{another case,} \end{cases} \quad (3)$$

wherein  $\text{Sup}(M_i(e))$  is the lowest value of  $\beta$  that is higher than all values of the  $M_i(e)$  range. For all  $\beta < \lambda_i(e)$ , the element  $e$  belongs to the internal border  $B_i$ . Thus,

**Proposition 1.**  $\forall e \in U, 1 \leq i \neq j \leq m$ , if  $\lambda_i(e) \leq \lambda_j(e) \Rightarrow \forall \beta : \beta < \lambda_i(e), e \in B_i \wedge e \in B_j$ . Based on the analysis performed, a specific sequence of the supremum  $\lambda_i(e), 1 \leq i \leq m$  can be obtained for each element  $e \in U$  associated with each internal border  $B_i, Z_i(e)$ . Being  $Z_1(e), \dots, Z_m(e)$ , such that  $\lambda_{Z_1(e)}(e) \leq \dots \leq \lambda_{Z_m(e)}(e)$  a permutation of indices that order the  $\lambda_i(e)$ .

**Definition 3.** With  $e \in U, \beta \in [0; 0,5)$  and  $m$  the number of internal borders considered in the analysis, the Total number of internal borders to which element  $e$  belongs at a given  $\beta$  value is defined as follows:

$$\text{Total}(e, \beta) = \begin{cases} m, & \text{if } \beta < \lambda_{Z_1(e)}(e), \\ 0, & \text{if } \beta \geq \lambda_{Z_m(e)}(e), \\ m - \max_k (\beta \geq \lambda_{Z_k(e)}(e)), & \text{in another case.} \end{cases} \quad (4)$$

The first two parts of Definition 3 are established to ensure that when the max function is evaluated, a defined result is always established (especially when the condition established in the predicate  $\lambda_{Z_k(e)}(e)$  is not satisfied). The graphical interpretation of the  $\text{Total}(e, \beta)$  function is illustrated in Figure 3. In this figure,  $v = \max_k (\beta \geq \lambda_{Z_k(e)}(e))$ . This value is the highest value of  $k$  such that  $(\beta \geq \lambda_{Z_k(e)}(e))$ , that is, is exactly the number of internal borders to which  $e$  does not belong. Furthermore, from  $k' = k + 1, \beta < \lambda_{Z_{k'}}(e)$  will be fulfilled and therefore  $e$  belongs to the internal borders  $B_{Z_{k'}}(e), \dots, B_{Z_m(e)}$ , by Proposition 1 and does not belong to the internal borders  $B_{Z_1(e)}, \dots, B_{Z_k(e)}$ .

As a function of Definitions 2 and 3 and Proposition 1, the concept of the outlier degree of an element  $e \in U$  is defined as a function of the  $\beta$  values.

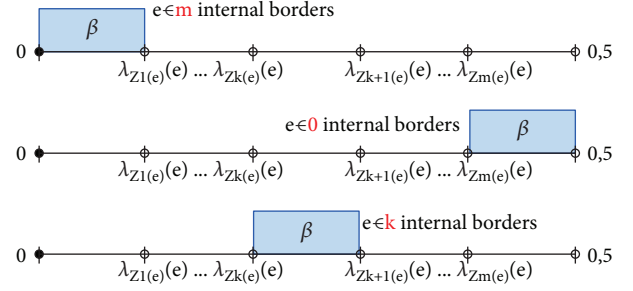


FIGURE 3: Graphic view of the  $\text{Total}(e, \beta)$  function.

**Definition 4.** With  $e \in U$  a value  $\beta \in [0, 0.5]$  and  $m$  the number of internal borders considered in the analysis, the outlier degree of element  $e$  at a given  $\beta$  value is defined as follows:  $\text{ExcepDegree}(e, \beta) = \text{Total}(e, \beta)/m$ .

This definition does not contradict the proposition presented in [6]. Based on this proposition,  $\forall e \in U$ , the outlier degree of such element can be assessed for any  $\beta$  value and therefore the  $\mu$  values for which  $\text{ExcepDegree}(e, \beta) \geq \mu$ .

**2.1.3. Integrating Regions.** The definitions above enable us to establish the following general method for determining the values of  $\beta$  and  $\mu$  for which the element  $e \in U$  is an outlier in  $U$ .

- (1) To determine  $M_i(e)$ :  $\beta$  values for which the element  $e \in B_i$
- (2) To determine  $S_i$ :  $\beta$  values for which there is no internal border that is a subset of the internal border  $B_i$
- (3) To determine  $D_i(e) = M_i(e) \cap S_i$ :  $\beta$  values for which the element  $e$  belongs to  $B_i$  and there is no internal border that is a subset of the internal border  $B_i$

For values of  $\beta \in D_i(e)$ , the element  $e$  belongs to some nonredundant outlier set and is the only representative of the internal border  $B_i$  in such set, that is, for  $\beta$  values in  $D_i(e), e \in E_i$

- (4)  $\forall \beta_o, \mu_o: \beta_o \in \cup_{k=1}^m D_k(e) \wedge \mu_o \leq \text{ExcepDegree}(e, \beta_o)$ , then:  $e$  is an outlier in  $U$ . A  $\beta_o \in \cup_{k=1}^m D_k(e)$  represents a value for which the element  $e$  belongs to some internal border of which no other internal border is a subset, and in such a case,  $\mu_o$  must be lower than or equal to  $\text{ExcepDegree}(e, \beta_o)$

Figure 4 shows the range of  $\beta$ - $\mu$  values for which any element  $e$  of the universe is an outlier in  $U$ . In this case, the following was assumed:

$$\text{range}(1) \cup \text{range}(2) = \cup_{k=1}^m D_k(e). \quad (5)$$

**2.2. Computational Implementation: FIND\_OUTLIER\_REGION Algorithm.** In this section, the FIND\_OUTLIER\_REGION algorithm is developed. This algorithm enables the unsupervised calculation of the range of values of the thresholds  $\beta$ - $\mu$  in which each element of the universe is an outlier.

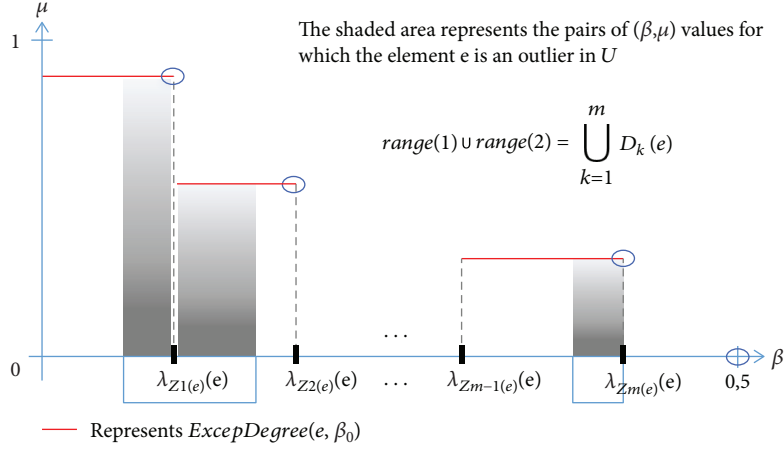


FIGURE 4: Range of  $\beta$ - $\mu$  values for which any element of the universe is an outlier in  $U$ .

This algorithm validates the  $\beta$ - $\mu$  method defined in the previous section and proceeds in three key steps.

- Calculation of the dependences between internal borders, or calculation of the inclusion relationship between them: `BUILD_ $\beta$ _OUTLIER_REGION` algorithm (see Algorithm 1)
- Calculation of the outlier region in relation to the threshold  $\mu$ : `BUILD_ $\mu$ _OUTLIER_REGION` algorithm (see Algorithm 2)
- Integration of both regions to obtain, for each element of the universe, the regions of  $\beta$ - $\mu$  values in which the element would be an outlier: `OUTLIERS` set and `FIND_OUTLIER_REGIONS` algorithm (see Algorithm 3)

All these algorithms contain the inputs *universe*  $U$  (dataset),  $|U| = n$ , and *concept*  $X \subseteq U \neq \emptyset$  and the equivalence relationships  $R = \{r_1, r_2, \dots, r_n\}$ .

The output of the `BUILD_ $\beta$ _OUTLIER_REGION` algorithm (Algorithm 1) is set  $S$  with the dependences between internal borders or the inclusion relationship between them. The output of the `BUILD_ $\mu$ _OUTLIER_REGION` algorithm (Algorithm 2) consists of a tuple with two values: the outlier region `ExcepDegree` in relation to the outlier threshold  $\mu$  and the set of classification errors  $\beta$  for which each element belongs to each equivalence relation  $r_i \in R$ .

Finally, the output of the `FIND_OUTLIER_REGION` algorithm (Algorithm 3) is the set of `OUTLIERS` with the regions of the  $\beta$ - $\mu$  values in which every element would be an outlier.

**2.3. Analysis of the Complexity of the Method and the Algorithm.** The temporal complexity of the algorithms depends on the number of ranges in the sets of specific ranges. Table 1 outlines the costs of each structure calculated for each algorithm. Based on these calculations, the temporal complexity of the `FIND_OUTLIER_REGION` algorithm is then determined, which, in the worst case,

will be equal to the maximum of each of its three main tasks:  $\mathcal{O}(n^2 \times m^2 \times \log(m))$ .

The most original aspect of the `FIND_OUTLIER_REGION` algorithm is that it enables the unsupervised calculation of the range of threshold values (parameters  $\beta$  and  $\mu$ ) in which each element of the universe will be considered an outlier. However, the temporal and spatial complexity of the algorithm is of a higher order than that of the algorithms Pawlak rough sets and VPRS [1, 7] because the result from the `FIND_OUTLIER_REGION` algorithm is more general.

When executing the algorithm once for a given data universe, the specific outputs of the previous algorithms can be obtained for any value of  $(\beta, \mu)$ . Determining, for each element of the universe, the total region of values of such thresholds in which such element is an outlier ensures that the entire universe can be subsequently searched for specific pairs of values of the thresholds  $(\beta, \mu)$  belonging to the outlier region of any element. Thus, the usefulness of the `FIND_OUTLIER_REGION` algorithm becomes clear when seeking to assess the outlier condition of the elements of the universe for a given set of threshold values.

In summary, the result from the execution of the algorithm contains any particular result that could be obtained from the execution of the algorithms Pawlak rough sets and VPRS. This is the main advantage of the algorithm, compared with the expected advantage from increasing its temporal and spatial complexity when used only to calculate the regions of a single element of the universe.

Nevertheless, despite the high order of temporal complexity identified in the *worst case*, the algorithm can reach an order of temporal complexity similar to that of the algorithms Pawlak rough sets and VPRS, almost linear for the *best case*  $\Omega(n \times m^2 \times c)$ .

The `OUTLIERS` region obtained allows a stochastic approximation to the solution of the problem of determining whether a given element is an outlier within a given universe of data (to establish a probabilistic criterion on such condition).

<b>BUILD_<math>\beta</math>_OUTLIER_REGION (U, X, R): S</b>	
<i>Pseudo-code</i>	<i>Comments</i>
1 <b>for each</b> $r \in R$	
2 <b>for each</b> $q \in R - \{r\}$	
3 $S1[r][q] = \{[0, 0.5]\}$	Start solving Sub-problem No. 1
4 $S3[r][q] = \{[0, 0.5]\}$	Start solving Sub-problem No. 3
5 $S2[r] = \{[0, 0.5]\}$	Start solving Sub-problem No. 2
6 <b>for each</b> $r \in R$	
7 $P_r = \text{CLASSIFY-ELEMENTS}(U, r)$	Partition induced by the equiv. relation $r$
8 $\text{class-max} = 0$	starting the null minimum value $[r]$
9 <b>for each</b> $\text{class} \in P_r$	
10 $\text{case1}[r][\text{class}] = \{[\min(c(\text{class}, X), 1 - c(\text{class}, X)), 0.5]\}$	Obtain the solution for the equivalence class for Case1
11 $\text{class-max} = \max(\text{class-max}, c(\text{class}, X), 1 - c(\text{class}, X))$	Update the null minimum value $[r]$
12 <b>for each</b> $q \in R - \{r\}$	Searching the solution for the equiv. class of case2
13 $q\text{-min} = \min(c(\text{class}, X), 1 - c(\text{class}, X))$	Minimum error of the equiv. classes according to $q$ with elements of the equiv. class according to $i$
14 <b>for each</b> $e \in \text{class}$	For each class element
15 $q\text{-class} = \text{CLASSIFY-ELEMENT}(U, q, e)$	Obtain equiv. class to which it belongs according to $q$
16 $q\text{-min} = \min(q\text{-min}, c(q\text{-class}, X), 1 - c(q\text{-class}, X))$	Update the minimum value
17 $\text{case2}[r][q][\text{class}] = [0, q\text{-min}]$	Obtain the solution of the equiv. class for Case 2
18 $S1[r][q] = S1[r][q] \cap (\text{case1}[r][\text{class}] \cup \text{case2}[r][q][\text{class}])$	Update $S1$ with new ranges of the equiv. class
19 $S2[r] = S2[r] \cap \{[\text{class-max}, 0.5]\}$	Update $S2$ with new ranges of the equiv. class
20 <b>for each</b> $r \in R$	Update $S3$ from the $S1$ values
21 <b>for each</b> $q \in R - \{r\}$	
22 $S3[q][r] = S1[r][q] \cap S1[q][r]$	Obtain the solution for which the internal border $r$ is equal to $q$
23 <b>for each</b> $r \in R$	Calculate the outlier region for each internal border
24 $A = \{\}$	$\beta$ for which the internal border $r$ contains the other internal border
25 <b>for each</b> $q \in R - \{r\}$	
26 $A = A \cup (S1[q][r] - S3[q][r] - S2[q])$	Update set $A$
27 $S[r] = \{[0, 0.5]\} - A - S2[r]$	Values for which the internal border $r$ has no internal border
28 <b>return</b> $S$	Return the solution

ALGORITHM 1: Pseudo-code of the BUILD\_ $\beta$ \_OUTLIER\_REGION algorithm.

<b>BUILD_<math>\mu</math>_OUTLIER_REGION (U, X, R): {M, ExcepDegree}</b>	
<i>Pseudo-code</i>	<i>Comments</i>
1 <b>for each</b> $e \in U$	For each element of the universe
2 <b>for each</b> $r \in R$	For each equiv. relation
3 $\text{class} = \text{CLASSIFY-ELEMENT}(U, r, e)$	Obtain the equiv. class of the element
4 $\lambda[e][r] = \min(c(\text{class}, X), 1 - c(\text{class}, X))$	Obtain the lowest $\beta$ higher than all values of $M[e][r]$
5 $M[e][r] = \{[0, \lambda[e][r]]\}$	Obtain the $\beta$ for which the element belongs to $r$
6 $h = 1.0$	
7 $\text{prev} = 0.0$	
8 <b>for each</b> $\text{inf} \in \text{SORT}(\lambda[e])$	For each infimum in the order
9 $\text{base} = \{\}$	Obtain $\beta$ ranges of height $m$
10 $\text{ExcepDegree}[e] = \text{ExcepDegree}[e] \cup \{[\text{prev}, \text{inf}] \times [0, h]\}$	Obtain the outlier rectangle
11 $\text{prev} = \text{inf}$	Save the value to form the next rectangle
12 $h = h - (1/ R )$	Reduce the outlier rectangle height
13 <b>return</b> $\langle M, \text{ExcepDegree} \rangle$	Return $M$ and ExcepDegree

ALGORITHM 2: Pseudo-code of the BUILD\_ $\mu$ \_OUTLIER\_REGION algorithm.

FIND_OUTLIER_REGION (U, X, R): OUTLIERS		
Pseudo-code		Comments
1 S = BUILD_β_OUTLIER_REGION (U, X, R)	Step 1: calculation of the dependences between internal borders	
2 <M, ExcepDegree> = BUILD_μ_OUTLIER_REGION (U, X, R)	Step 2: calculation of the outlier region	
3 for each e ∈ U	Integration of the regions	
4 D[e] = {}	For each element of the universe	
5 for each r ∈ R	Values where e belongs to an internal border with no other internal border	
6 D[e] = D[e] ∪ M[e][r] ∩ S[r]		
7 OUTLIERS[e] = ExcepDegree[e] ∩ {D[e] × [0, 1]}	Intersection between the outlier regions β and μ	
8 return OUTLIERS	Return all regions	

ALGORITHM 3: Pseudo-code of the FIND\_OUTLIER\_REGION algorithm.

TABLE 1: Calculation of the spatial and temporal complexity of the FIND\_OUTLIER\_REGION algorithm by calculating the complexities of each structure of each component algorithm.

Algorithm	Data structure	Spatial complexity (worst case)	Temporal complexity (worst case)
BUILD_β_OUTLIER_REGION	Case1[i][ec]	$O(n \times m)$	$O(n \times m \times c)$
	Case2[i][j][ec]	$O(n \times m^2)$	$O(n \times m^2 \times c)$
	S1[i][j]	$O(n \times m^2)$	$O(n \times m \times \log(n))$
	S2[i]	$O(m)$	$O(n \times m)$
	S3[i][j]	$O(n \times m^2)$	$O(n \times m^2)$
	S[i]	$O(n \times m^2)$	$O(n \times m^2 \times \log(m))$
BUILD_μ_OUTLIER_REGION		$O(n^2 \times m^2)$	$O(n \times m^2 \times \log(m))$
	λ[e][i]	$O(n \times m)$	$O(n \times m \times c)$
	M[e][i]	$O(n \times m)$	$O(n \times m \times c)$
	ExcepDegree[e]	$O(n \times m)$	$O(n \times m \times \log(m))$
FIND_OUTLIER_REGION		$O(n \times m)$	$O(n \times m \times \log(m))$
	D[e]	$O(n^2 \times m^2)$	$O(n^2 \times m^2 \times \log(m))$
	OUTLIERS[e]	$O(n^2 \times m^2)$	$O(n^2 \times m^2)$
		$O(n^2 \times m^2)$	$O(n^2 \times m^2 \times \log(m))$

### 3. Estimation of the Outlier Probability of Each Element

In the previous section, a theoretical framework was defined by expanding [1, 7], based on which the FIND\_OUTLIER\_REGION algorithm was constructed. This algorithm enables us to calculate all outlier regions for each element of the universe, and the complexity of this algorithm is almost linear. Ultimately, these results enable us to develop the solution proposed in this study (Figure 2): a computationally viable algorithm, valid for environments of large volumes of data, able to provide the outlier probability of each element of the universe. This algorithm was termed the  $\beta\mu\_PROB$  algorithm. Following a pattern similar to that followed in the previous section, first, a theoretical framework will be developed by expanding [1, 7], which will provide the

mathematical tools we need to build the solution. Subsequently, the spatial and temporal complexity of the algorithm will be analysed.

**3.1. Theoretical Framework: Probabilistic  $\beta\mu$  Method.** As mentioned above, the results from the previous section enable us to assess, for each  $e \in U$ , the region of  $\beta$  and  $\mu$  values in which such element is an outlier. Let us call  $OUTLIERS_e$  the region found for a given element,  $e \in U$ .

Considering  $\beta$  and  $\mu$  two random variables, let us call  $\varphi(\beta, \mu)$  the probability density function of the random vector  $(\beta, \mu)$ . Then, the distribution function of  $(\beta, \mu)$  would be

$$P(\beta \leq i, \mu \leq j) = \int_{-\infty}^i \int_{-\infty}^j \varphi(\beta, \mu) d\beta d\mu. \quad (6)$$



$\beta\mu\_PROB(U, X, R, PDF()); P$	
<i>Pseudo-code</i>	<i>Comments</i>
1 OUTLIERS = FIND_OUTLIER_REGION(U, X, R)	Apply probability distribution PDF for each region
2 <b>for each</b> $e \in U$	For every element of the universe
$P[e] = 0$	Initial probability
3 <b>for each</b> rect $\in$ OUTLIERS[e]	For each rectangle of exceptionality
4 $P[e] = P[e] + PDF(rect)$	Accumulate the probability of each rectangle
5 <b>return</b> P	Return P

ALGORITHM 4: Pseudo-code of the  $\beta\mu\_PROB$  algorithm.

Then, the probability that we are interested in calculating,  $P_e$ , that is, the probability that  $e \in U$  is an outlier knowing  $OUTLIERS_e$  can be calculated from (6) using the following formula:

$$P_e = P((\beta, \mu) \in OUTLIERS_e) = \int_{OUTLIERS_e} \varphi(\beta, \mu) d\beta d\mu. \quad (7)$$

Considering that  $e$  is an outlier of  $\beta$  and  $\mu$  values belonging to  $OUTLIERS_e$ .

Because  $\beta$  and  $\mu$  are two independent random variables, then:  $\varphi(\beta, \mu) = f(\beta)g(\mu)$ , where  $f(\beta)$  and  $g(\mu)$  are the probability density functions of  $\beta$  and  $\mu$ , respectively. Therefore,

$$P_e = \int_{OUTLIERS_e} f(\beta) \cdot g(\mu) d\beta d\mu. \quad (8)$$

We only have to replace the probability density functions of the parameters  $\beta$  and  $\mu$  in (8) to calculate  $P_e$  and then calculate the resulting integral. In practice, most commonly, no information about the distribution of the parameters  $\beta$  and  $\mu$  is available. Therefore, they will be both assumed to be uniformly distributed. If, in any context, this distribution is different from the expected, it is sufficient to calculate  $P_e$  with new functions, using some numerical method to calculate the integral if necessary. Based on this assumption, the resulting integral is easily calculated. Because  $0 \leq \beta < 0.5$  and  $0 \leq \mu \leq 1$ , based on the *Uniformity hypothesis* for the values of these thresholds, its probability density function would be

$$\begin{aligned} f(\beta) &= \frac{1}{0.5 - 0} = 2, \\ g(\mu) &= \frac{1}{1 - 0} = 1. \end{aligned} \quad (9)$$

Replacing these values in (8), we have

$$P_e = 2 \int_{OUTLIERS_e} d\beta d\mu. \quad (10)$$

And because  $\int_{OUTLIERS_e} d\beta d\mu$  is the area of the  $OUTLIERS_e$  region,

$$P_e = 2 \text{Area}(OUTLIERS_e). \quad (11)$$

This result may be interpreted as

$$P_e = \frac{\text{Area}(OUTLIERS_e)}{0.5}. \quad (12)$$

This is precisely the quotient between the area of the favourable region (the region of values  $(\beta, \mu)$  for which  $e$  is an outlier) and the total area (the rectangle that defines the domain of the values  $(\beta, \mu)$  on the plane).

**3.2. Computational Implementation:  $\beta\mu\_PROB$  Algorithm.** The  $\beta\mu\_PROB$  algorithm input consists of the following: a universe  $U$  (dataset)  $|U| = n$ , a concept  $X \subseteq U \neq \emptyset$ , equivalence relations  $R = \{r_1, r_2, \dots, r_n\}$ , and a probability distribution function  $PDF()$ . Its output consists of estimating the probability  $P$  for each element of  $U$  in terms of their outlier status in the universe. Because the *FIND\_OUTLIER\_REGION* algorithm calculates the outlier region  $OUTLIERS$ , the probability is calculated using the formula shown in (12). A description in pseudo-code of the algorithm that implements the aforementioned aspects is presented in Algorithm 4.

The temporal complexity of the  $\beta\mu\_PROB$  algorithm is affected by the temporal complexity of the process for determining the outlier region:

- (i) Cost of determining the outlier region: temporal complexity *FIND-OUTLIER-REGION*:  $\mathbf{O}(n^2 \times m^2 \times \log(m))$
- (ii) Cost of determining the probability: (dataset)  $\times$  (total number of rectangles region  $\beta\text{-}\mu$ )  $= (n) \times (n \times m^2) \rightarrow \mathbf{O}(n^2 \times m^2)$

Therefore, the temporal complexity of the algorithm  $\beta\mu\_PROB$ , in the worst case, is  $\mathbf{O}(n^2 \times m^2 \times \log(m))$ .

The  $\beta\mu\_PROB$  algorithm solves two key problems: the lack of a specific algorithm to perform this calculation and the complexity of the calculation performed by combining existing algorithms [1, 7]; the resultant reduction

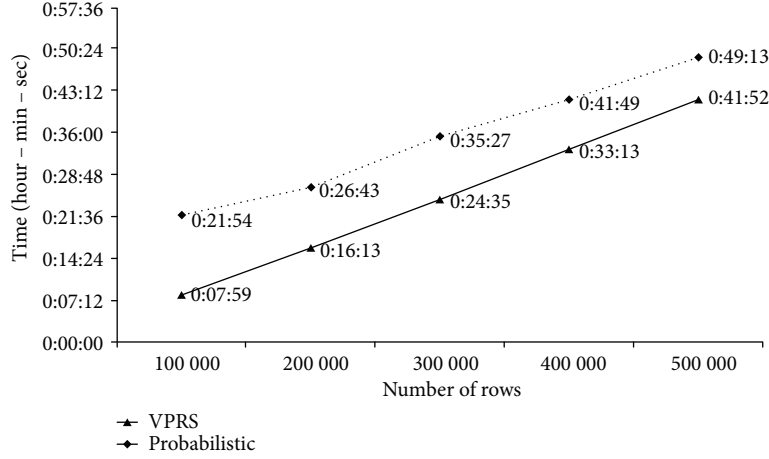


FIGURE 5: Comparison of run-times between the VPRS and  $\beta\mu\_PROB$  algorithms.

in complexity allows application of the algorithm to environments with large volumes of information.

#### 4. Validation of the Results

The algorithm validation tests have primarily focused on two aspects: comparing its run-times to those of the VPRS algorithm to obtain a realistic reference and assessing the detection quality of the  $\beta\mu\_PROB$  algorithm. For such purposes, automatically generated random datasets and real-world datasets were used. Although performing quantitative comparisons to all algorithms identified in the state of the art is usually senseless due to the different nature of their application and usefulness, a comparison that allows us to contextualise each of them can be very interesting. Accordingly, the rest of the section is structured as follows: (1) evaluation of the algorithm run-times and comparison to the VPRS case, (2) evaluation of the detection quality, which is also compared to that of the VPRS, and (3) comparison of all RS-based methods to algorithms based on conventional methods and comparison to the advantages and drawbacks of each RS-based method of the study.

**4.1. Run-Time Study.** The  $\beta\mu\_PROB$  algorithm run-time validation tests—compared to the VPRS algorithm [10]—are performed with large datasets having high dimensionality. Because similar results have been found in all the experiments, in this study, we show a specific example that is fully representative: multivariate synthetic data (random dataset automatically generated using statistical techniques that ensure a uniform distribution, among other aspects) with categorical and continuous attributes, with 500,000 records and with 100 columns. The number of equivalence relations covered is 100. The computing device used has the following characteristics: Intel(R) Core(TM)2 Quad processor CPU Q6600 @ 2.40 GHz, with 3.25 GB of memory running the Windows 7 Ultimate operating system.

Figure 5 shows the run-times assessed both for the  $\beta\mu\_PROB$  and the VPRS algorithms. The equivalence relations and the number of columns remain fixed for the comparison, varying the number of records.

The curves show that both algorithms behave similarly—regarding the run-time—and that they are computationally efficient when analysing a large dataset with high dimensionality. Furthermore, the run-times are linear and advantageously require no preset thresholds.

This finding shows that although the order of temporal complexity for the BM\_PROB algorithm is quadratic in the worst case, it may reach an almost linear order of temporal complexity when analysing datasets that are normally distributed.

**4.2. Detection Quality Validation.** Again, all experiments conducted yielded similar results; therefore, in this study, one of them is shown as a representative example. In this case, the dataset used was the Arrhythmia Data Set (data of patients with cardiovascular problems) from the UCI Machine Learning Data Repository [12]. These are multivariate data with real, complete, and categorical attributes. Here, 452 records from 279 fields were employed. The computing device used was an Intel(R) Core(TM) 2 Duo, CPU T5450 @ 1.66 GHz (with 2 CPUs), and 2046 MB of RAM running Windows Vista.

The concept  $C$  defined people with weight  $\leq 40$  kg, that is, low-weight people, and the following equivalence relations  $R$ :

- (i)  $r_1$ : was established from the attribute heart rate: mean number of heart beats per minute of each person. The equivalence relation partitions the dataset into two equivalence classes: [44, 61] and [62, 163]
- (ii)  $r_2$ : was established from the attribute number of intrinsic deflections: number of arterial bypasses of each person. The equivalence relation partitions the dataset into two equivalence classes: [0, 59] and [60, 100]
- (iii)  $r_3$ : was established from the attribute height: height of a person expressed in centimetres. The equivalence relation partitions the dataset into two equivalence classes: [60, 175] and [176, 190]

Here, 12 outliers with contradictory values for low-weight people were intentionally injected into the dataset. The normal values of the attributes considered in the equivalence relations for low-weight people are as follows: heart rate  $>65$ , intrinsic deflections  $<50$ , and height  $<170$  cm. Table 2 describes the outliers injected. The values in bold and italics represent contradictory values.

In the test, the following  $\mu$  values were analysed: 0.2, 0.4, 0.6, 0.8, and 1. For each  $\mu$  value,  $\beta$  was varied according to the following sequence of values: 0, 0.1, 0.2, and 0.3. The values 0.4 and 0.5 are not mentioned because the number of outliers detected remained 0 beyond  $\beta = 0.3$ . After applying the  $\beta\mu$ \_PROB algorithm, different subsets formed by  $k$  elements, with  $k \in (5, 10, 15, 20)$ , are taken from the dataset with the highest outlier probability. Then, the number of injected outliers found in each of these subsets is analysed. Figure 6 shows the results achieved on this occasion.

The number of most likely elements ( $k$ ) considered in each case shows that when  $k = 5$ , the 5 elements with the highest outlier probability are the 5 most contradictory elements of the dataset; when  $k = 10$ , the 10 elements with the highest outlier probability introduced in the dataset and, when  $k = 15$  and  $k = 20$ , the 12 outliers intentionally injected already appeared among the most likely  $k$ . In summary, the 12 injected elements were always found among those with the highest outlier probability after applying the  $\beta\mu$ \_PROB algorithm.

Table 3 presents the probability values determined using the  $\beta\mu$ \_PROB algorithm for outliers injected into the dataset.

**4.3. Comparison of the Outlier Detection Algorithms.** Most outlier detection techniques and algorithms analysed are designed, to a greater or lesser extent, to solve a specific type of problem, even in a specific case. Valid comparisons between these algorithms are difficult to perform because they will considerably depend on the search target. However, it is interesting to perform a comparative study of the different existing methods highlighting the advantages from the current proposal in its field—the unsupervised provision of general results regarding all elements of the data universe by establishing specific initial conditions: concept and equivalence relations. Considering the above, Table 4 details how the  $\beta\mu$ \_PROB algorithm may help to overcome the limitations of the methods studied when requiring generalisation.

The main advantage of RS-based proposals and, particularly, of the  $\beta\mu$ \_PROB algorithm relative to conventional methods lies in its generalist character. Unsurprisingly, an algorithm specially designed to detect a specific type of outliers is usually better, both in terms of detection quality and spatial and temporal complexity. However, having a generic algorithm that is capable of addressing different types of problems, with different types of data, and able to behave reasonably with large volumes of data is a very interesting option that avoids having to design different algorithms each time new problems emerge or when the conditions of previously solved problems change.

TABLE 2: Outliers injected into the test dataset.

ID	Weight (kg)	Heart rate	# intrinsic deflections	Height (cm)
1	15	<b>60</b>	17	<b>180</b>
2	31	93	<b>68</b>	<b>178</b>
3	39	<b>50</b>	<b>82</b>	130
4	10	<b>53</b>	16	<b>188</b>
5	19	<b>45</b>	<b>90</b>	<b>190</b>
6	20	<b>48</b>	<b>86</b>	<b>183</b>
7	25	<b>50</b>	<b>71</b>	<b>180</b>
8	29	<b>55</b>	<b>75</b>	<b>179</b>
9	33	90	<b>60</b>	<b>176</b>
10	40	<b>61</b>	20	<b>186</b>
11	26	<b>50</b>	<b>99</b>	<b>180</b>
12	38	92	<b>100</b>	<b>178</b>

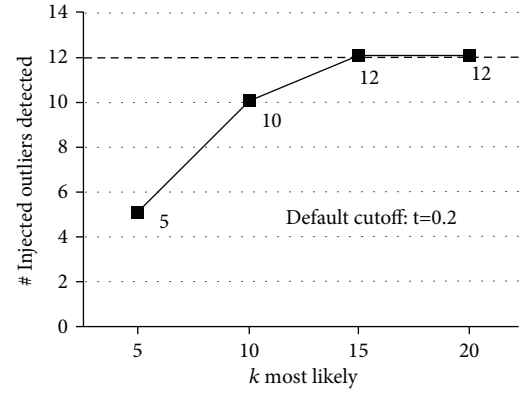


FIGURE 6: Number of injected outliers found between the  $k$  elements with the highest outlier probability.

After comparing algorithms based on conventional techniques and algorithms based on the RS model, a summary of the comparative study conducted between different RS algorithms and the proposed  $\beta\mu$ \_PROB algorithm is presented in Table 5, outlining the advantages and disadvantages of each algorithm and highlighting the usefulness of the proposed algorithm.

## 5. Conclusions

Whereas VPRS has been applied to problems in multiple fields [13–16], particularly in the field of statistics [17], this study aimed to develop a new application of this model to the outlier detection problem, breaking with the traditional scheme followed by most existing detection methods. By defining the desired concept and equivalence relations, the algorithm provides unsupervised—and without needing to define neither the outlier threshold nor the classification error, which are both dependent on the problem—general results regarding all elements of the dataset. More specifically, it provides the outlier probability of each element from such universe. Therefore, this result is transcendent and

TABLE 3: Outlier probability of the 12 elements injected into the dataset.

ID	Weight (Kg)	Heart rate	# of intrinsic deflections	Height (cm)	Outlier probability
1	15	<b>60</b>	17	<b>180</b>	0.61884
2	31	93	<b>68</b>	<b>178</b>	0.7557252
3	39	<b>50</b>	<b>82</b>	130	0.6151009
4	10	<b>53</b>	16	<b>188</b>	0.61884
5	19	<b>45</b>	<b>90</b>	<b>190</b>	<b>0.8779342</b>
6	20	<b>48</b>	<b>86</b>	<b>183</b>	<b>0.8779342</b>
7	25	<b>50</b>	<b>71</b>	<b>180</b>	<b>0.8779342</b>
8	29	<b>55</b>	<b>75</b>	<b>179</b>	<b>0.8779342</b>
9	33	90	<b>60</b>	<b>176</b>	0.7557252
10	40	<b>61</b>	20	<b>186</b>	0.61884
11	26	<b>50</b>	<b>99</b>	<b>180</b>	<b>0.8779342</b>
12	38	92	<b>100</b>	<b>178</b>	0.7557252

TABLE 4: Characteristics of the RS-based methods compared to the limitations of conventional methods.

*Comparison to STATISTICAL and DISTANCE-BASED METHODS*

- (i) Applicability to datasets with a mixture of continuous and discrete attributes. Equivalence relationships are a natural way to discretise continuous data.
- (ii) Neither knowing the data distribution nor establishing data *distance* criteria is required.
- (iii) Specifically, for  $\beta = 0$ , the quadratic temporal complexity problem of most *distance*-based methods is solved.
- (iv) The dimensionality and dataset size do not limit the execution of the algorithms.

*Comparison to DENSITY- and DEPTH-BASED METHODS*

- (i) There is no need to establish data density criteria in the dataset.
- (ii) The dimensionality of the dataset does not limit the execution of the algorithm.
- (iii) No time-consuming calculations are necessary, including calculating the *convex wrap*, which is required in most *depth*-based methods.
- (iv) *FIND\_OUTLIER\_REGION* and  $\beta\mu\_PROB$  provide unsupervised results without requiring the user to preset, before running the algorithm, the value of specific analysis parameters, which is necessary in *density*-based methods, such as *DBSCAN*.
- (v) *Pawlak rough sets* and *VPRS* improve the temporal complexity compared to *depth*-based methods.

*Comparison to METHODS BASED ON NEURAL NETWORKS*

- (i) No time-consuming processes must be previously established, for example, network training, required in some neuronal network models to ensure their learning.
- (ii) The dimensionality of the dataset does not limit the execution of the algorithms.
- (iii) The functionality of the algorithms does not depend on data *density* criteria, in contrast to some supervised models.
- (iv) There is no need to model the data *distribution*, in contrast to some supervised models.
- (v) Some approaches based on supervised networks establish the use of thresholds for various purposes in the *outlier* detection process. This is solved in the concept of the *FIND\_OUTLIER\_REGION* and  $\beta\mu\_PROB$  algorithms.

*Comparison to GENERAL OUTLIER DETECTION METHODS*

- (i) In contrast to most detection methods, which require successive executions of the algorithm until obtaining the set of outliers that actually meets the analysis criteria,  $\beta\_PROB$  algorithm performs the single-run, unsupervised determination of the outlier probability of each element from a specific universe of data.

original because it paves the way for the analysis and solution of other particular problems. It allows us to have an overview of the data and thus to test its representativeness.

The algorithms presented demonstrate the computational feasibility of the proposed methods. Furthermore, they provide efficient computational solutions—in terms of temporal and spatial complexity—to the problems for which they were conceived.

The method proposed solved, in addition, other limitations of several detection methods: it may be applied to datasets with a mixture of types of attributes (continuous and discrete); its application requires no prior knowledge about the data distribution; within the scope of its application, the size and dimensionality of the dataset do not limit its correct operation; and no distance or density criteria must be established for the dataset to apply this algorithm.

TABLE 5: Comparative table of RS-based algorithms.

Advantages	Disadvantages
<i>Pawlak rough sets algorithm</i>	
(i) Shows the computational viability of the <i>Pawlak rough sets</i> -based detection method.	(i) DETERMINISTIC classification.
(ii) Linear temporal and spatial lineal complexity regarding the cardinality of the dataset.	(ii) The user must define the <i>outlier threshold</i> .
<i>VPRS algorithm</i>	
(i) Shows the computational viability of the <i>VPRS</i> -based detection method.	(i) The user must define the <i>outlier threshold</i> and the <i>classification error</i> .
(ii) Linear temporal and spatial lineal complexity regarding the cardinality of the dataset.	(ii) An inadequate selection of the <i>error</i> may lead to unsatisfactory results. Requires sufficient knowledge of specific aspects of the <i>dataset</i> .
(iii) NONDETERMINISTIC classification.	
<i>FIND_OUTLIER_REGION algorithm</i>	
(i) Shows the computational viability of the $\beta\mu$ Method.	
(ii) Maintains the nondeterminism of <i>VPRS</i> .	
(iii) Any specific result that could be obtained with the <i>Pawlak rough sets</i> and <i>VPRS</i> algorithms can be determined from the result obtained.	(i) Temporal complexity: $O(n^2 \times m^2 \times \log(m))$ in the worst case.
(iv) The obtained region allows us to establish a stochastic approach to solving the problem of determining the outlier probability of a given element from a given dataset.	(ii) Spatial complexity: $O(n^2 \times m^2)$ in the worst case.
(v) Its use is especially feasible when needing to determine the <i>outlier</i> condition of the elements of the <i>dataset</i> for a given set of threshold values.	
<i><math>\beta\mu</math>_PROB algorithm</i>	
(i) Shows the computational viability of the method defined.	
(ii) Maintains the nondeterminism of <i>VPRS</i> .	
(iii) Has the same advantages as the <i>FIND_OUTLIER_REGION</i> algorithm.	(i) Temporal complexity: $O(n^2 \times m^2 \times \log(m))$ in the worst case.
(iv) The user does not need to define the outlier threshold, the allowed classification error, or other criteria, such as distance or density.	(ii) Spatial complexity: $O(n^2 \times m^2)$ in the worst case.
(v) No specific knowledge of the dataset is required, such as its distribution.	
(vi) The result obtained is more general than that obtained with <i>Pawlak rough sets</i> and <i>VPRS</i> .	
(vii) Is valid for datasets with mixed types of attributes (continuous and discrete).	

The results reported in the present study are the beginning of an in-depth study in the context of the general problem of outlier detection based on the RS model. Therefore, several problems that have not yet been solved may be identified and may be the next objectives of this on-going study. Accordingly, the following objectives have been identified: (a) to further improve the run-time of the algorithms by creating a distributed execution mechanism to use the computational power of several machines in one domain. In the current version of the algorithms, the user has to execute them on a single personal computer (PC), and (b) in the current version of the  $\beta\mu$ \_PROB algorithm, the  $\beta$  threshold domain is  $[0; 0.5]$ . However, the establishment of a new upper bound could allow us to gain precision in the probability calculation, especially in the case of very contradictory elements for few  $\beta$  values. Accordingly, the BM/probabilistic algorithm should be modified to automatically determine the most appropriate value for a given level.

## Data Availability

The main dataset used to support the findings of this study is public and you can access it in Maching Learning Repository: Arrhythmia Data Set at URL <https://archive.ics.uci.edu/ml/datasets/arrhythmia>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Funding

The authors received Fund no. TIN2016-78103-C2-2-R.

## Acknowledgments

This work has been supported by University of Alicante projects GRE14-02 and Smart University.

## References

- [1] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [2] S. W. Han and J.-Y. Kim, "Rough set-based decision tree using the core attributes concept," in *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, p. 298, Kumamoto, Japan, 2007, IEEE Computer Society.
- [3] C. Cheng, Y. Chen, and J. Chen, "Classifying initial returns of electronic Firm\_s IPOs using entropy based rough sets in Taiwan trading systems," in *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, p. 82, Kumamoto, Japan, 2007, IEEE Computer Society.



- [4] M. Hirokane, H. Konishi, A. Miyamoto, and F. Nishimura, "Extraction of minimal decision algorithm using rough sets and genetic algorithm," *Systems and Computers in Japan*, vol. 38, no. 4, pp. 39–51, 2007.
- [5] F. Jiang, Y. Sui, and C. Cao, "Outlier detection using rough set theory," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pp. 79–87, Springer, 2005.
- [6] F. Maciá-Pérez, J. V. Berna-Martínez, A. Fernández Oliva, and M. A. Abreu Ortega, "Algorithm for the detection of outliers based on the theory of rough sets," *Decision Support Systems*, vol. 75, pp. 63–75, 2015.
- [7] W. Ziarko, "Variable precision rough set model," *Journal of Computer and System Sciences*, vol. 46, no. 1, pp. 39–59, 1993.
- [8] W. P. Ziarko, Ed., *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer, 1994.
- [9] W. Ziarko, "Probabilistic decision tables in the variable precision rough set model," *Computational Intelligence*, vol. 17, no. 3, pp. 593–603, 2001.
- [10] A. Fernández Oliva, M. Abreu Ortega, M. C. Fernández Baizán, and F. Maciá Pérez, "Método de detección no determinista de outliers basado en el modelo de conjuntos aproximados de precisión variable," in *Jornadas para el Desarrollo de Grandes Aplicaciones de Red (JDARE'09)*, pp. 131–148, Alicante, España, 2009.
- [11] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Springer, 1991.
- [12] "UCI machine learning repository," May 2009, <https://cml.ics.uci.edu>.
- [13] Z. T. Gong, B. Z. Sun, Y. B. Shao, D. G. Chen, and Q. He, "Variable precision rough set model based on general relation," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, pp. 2490–2494, Shanghai, China, 2004.
- [14] M. J. Beynon and N. Driffield, "An illustration of variable precision rough sets model: an analysis of the findings of the UK monopolies and mergers commission," *Computers & Operations Research*, vol. 32, no. 7, pp. 1739–1759, 2005.
- [15] C. T. Su and J. H. Hsu, "Precision parameter in the variable precision rough sets model: an application," *Omega*, vol. 34, no. 2, pp. 149–157, 2006.
- [16] V. U. Maheswari, A. Siromoney, and K. M. Mehata, "The variable precision rough set model for web usage mining," in *Web Intelligence Research and Development*, vol. 2198 of Lecture Notes in Computer Science, pp. 520–524, Springer, Berlin, Heidelberg, 2001.
- [17] W. Ziarko, "Decision making with probabilistic decision tables," in *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC' 99)*, vol. 1711 of Lecture Notes in Computer Science, pp. 463–471, Springer, Yamaguchi, Japan, 1999.

## Research Article

# Case-Based Reasoning: The Search for Similar Solutions and Identification of Outliers

P. S. Szczepaniak and A. Duraj 

*Institute of Information Technology, Lodz University of Technology, Ul. Wólczanska 215, 90-924 Lodz, Poland*

Correspondence should be addressed to A. Duraj; [agnieszka.duraj@p.lodz.pl](mailto:agnieszka.duraj@p.lodz.pl)

Received 20 April 2018; Revised 24 June 2018; Accepted 8 July 2018; Published 26 August 2018

Academic Editor: David Gil

Copyright © 2018 P. S. Szczepaniak and A. Duraj. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The present paper applies the case-based reasoning (CBR) technique to the problem of outlier detection. Although CBR is a widely investigated method with a variety of successful applications in the academic domain, so far, it has not been explored from an outlier detection perspective. This study seeks to address this research gap by defining the outlier case and the underlining specificity of the outlier detection process within the CBR approach. Moreover, the case-based classification (CBC) method is discussed as a task type of CBR. This is followed by the computational illustration of the approach using selected classification methods, that is, linear regression, distance-based classifier, and the Bayes classifier.

## 1. Introduction

Case-based reasoning (CBR) is a computational problem-solving method that can be effectively applied to a variety of problems [1–10]. Broadly construed, CBR is the process of solving newly encountered problems by adapting previously effective solutions to similar problems (cases). Very important results concerning the equivalence of the learning power of symbolic and case-based methods were presented by Globig and Wess [7]. The authors introduced a case-based classification (CBC) as a variant of the CBR approach and integrated it with basic learning techniques. In particular, they presented the relationship between the case base, the measure of distance, and the target concept of the learning process, while constructing a number of algorithms of great practical significance. Those results justify the validity of the approach to outlier detection proposed in this paper.

In a negative scenario of the CBR cycle execution, the assessment of the nearest neighbour case or other proposed similar cases is negative, which implies that probably no neighbouring cases are useful. In this situation, the current case under consideration is a new one and becomes a

candidate to be called an outlier. In such case, the solution must be determined in a different way, but after the solution has been positively revised, the case should be included into the case base of the CBR system. Moreover, some of the cases already included in the case base which were never or hardly ever invoked and adapted by a large number of CBR system uses can be considered outliers. Interesting works that are worth mentioning in the context of outlier processing are those by Smyth and Keane [8, 9] and Richter et al. [10].

A considerable amount of literature has been published on outlier detection and analysis. These studies deal with diverse problem domains involving various types of data, including numeric, textual, categorical, and mixed-attribute records [11–18]. However, to the best of the authors' knowledge, no previous study has investigated case-based reasoning (CBR) from an outlier detection perspective. In this paper, outliers are defined more generally than they used to be to date, that is, as cases in the sense of CBR.

The paper is organized as follows: in Section 2, the principles of the case-based reasoning technique are presented. In Section 3, new definitions of case outlier in relation to CBR are given. Next, case-based classification is described, and computational illustration of outlier detection is given.

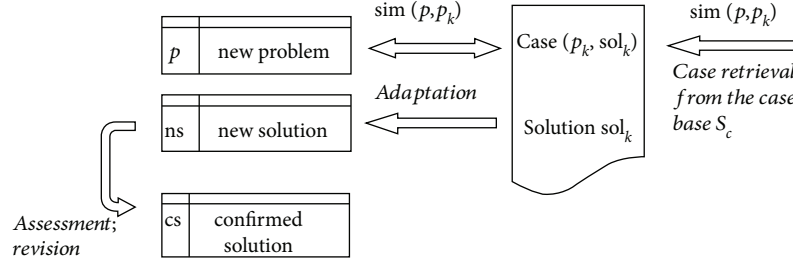


FIGURE 1: The CBR principle.

Finally, the last section gives a brief summary and critique of the findings.

## 2. Case-Based Reasoning (CBR)

*Case-based reasoning* (CBR) is considered a method for problem solving [1–3, 10], and *case-based classification* (CBC) is a task type of CBR. A detailed explanation of CBC is provided in Sections 4 and 5.

In the simplest definition within the CBR methodology, the *case* is understood as an ordered pair:

$$c_i = (p, \text{sol})_i, \quad (1)$$

or, in an extended form, as a triple

$$c_i = (p, \text{sol}, \text{eff})_i. \quad (2)$$

In (1), the previously examined situation (problem) is stored with its solution, and implicitly, the solution was a success. The effect manifested in (2) describes the results obtained through the implementation of the solution.

Medicine-related terms are the following: *p*—set of symptoms; *sol*—diagnosis or diagnosis with treatment; and *eff*—prognosis.

Some relevant data structures need to be used for the proper representation of both: problems and solutions. Here, the *attribute-value representation* is of practical importance. The possible attributes are name, set of values assigned to the name, or a variable. Another term frequently used to describe an attribute is *feature*.

The concept of *similarity* and its proper application is crucial for the implementation of the CBR system. In general, there are two ways for the computationally applicable similarity representation—relation or function. To reduce the theoretical considerations, the following assumptions are made, which immediately refer to the concept of case:

$$\begin{aligned} 0 &\leq \text{sim}(c_i, c_j) \leq 1, \\ \text{sim}(c_i, c_i) &= 1. \end{aligned} \quad (3)$$

The solution of a new problem *p* starts with the retrieval of the most similar case (according to the selection criterion); say that this is the *k*th case, from the base of previously solved cases ( $S_c$ ). The search for the nearest neighbour comes from the hypothesis that *similar problems have similar solutions*, although the nearest neighbour is not the only reasonable approach. Two situations are possible: (a) the features of both

entire cases—the query and the candidate ones—may be compared, or (b) relevant, significant portions of cases can be considered. Then, one considers the associated solution  $\text{sol}_k$ , which is either accepted in the given form or must be modified to be useful for the given new problem. This process is referred to as *case adaptation* (Figure 1). It is recommended that the *internal assessment* of the proposed solution of the current problem is performed within the CBR system. An external validation, called *revision*, is the definitive proof for correctness or practical usefulness of the proposed solution—*confirmed solution*. A case is added to the case base if it is recognized as a new one.

As an alternative to the concept of similarity, the concept of distance can be applied to the implementation of the CBR system. However, from the theoretical point of view, these two notions not only reflect different aspects of interpretation but also differ in terms of computational implementation.

Similarity and distance are considered objective notions. From the practical point of view, one looks for useful tools. Usefulness is considered a subjective notion, which can be stated a posteriori. Yet, it can be in some sense expressed by the notion of acceptance interpreted on the basis of the *preference relation* ([1] Chapter 2, [19]):

Given the query *q*,  $c_i >_q c_j$  means that case  $c_i$  is preferable to case  $c_j$ .

Both similarity and distance can determine preference relations:

$$c_i >_q c_j, \quad \text{if} \quad \text{sim}(q, c_i) \geq \text{sim}(q, c_j), \quad (4)$$

$$c_i >_q c_j, \quad \text{if} \quad \text{dis}(q, c_i) \leq \text{dis}(q, c_j). \quad (5)$$

The intuitive explanation is that in (4) we look for inclusive arguments, whereas in (5) we prefer rejection or being out of the cluster. The usability of the CBR system is an important feature, which depends strongly on the size and growth of the case base. In larger case bases, the retrieval stages are more expensive. To keep the size of the case base within the limits ensuring the efficiency and proper performance of the system, it is necessary to apply appropriate deletion policies.

In [8, 10], the authors described how the competence of a CBR system can be modelled and how deletion policies can exploit this model to guard against competence depletion while controlling the size of the case base in a manner that guards against the swamping problem. For this reason, the authors found it useful to consider four basic competence

categories of cases: *auxiliary*, *spanning*, *support*, and *pivotal*. They are defined using the concepts of coverage and reachability which are formulated as follows:

**Definition 1.** Coverage. Given a case base  $S_c = \{c_i\}$ ,  $i \in I$ . For  $c_i \in S_c$ ,

$$\text{coverage}(c_i) = \{c_j \in S_c : \text{adaptable}(c_i, c_j)\}. \quad (6)$$

**Definition 2.** Reachability. Given a case base  $S_c = \{c_i\}$ ,  $i \in I$ . For  $c_i \in S_c$ ,

$$\text{reachable}(c_i) = \{c_j \in S_c : \text{adaptable}(c_j, c_i)\}. \quad (7)$$

The coverage of a case is the set of target problems that can be used to solve. The reachability of a target problem is the set of cases that can be used to provide a solution for the target.

A case is an *auxiliary case* if the coverage it provides is subsumed by the coverage of one of its reachable cases. The cases of this category end to lie within clusters of cases and they do not affect competence at all. Their deletion only reduces the efficiency of the CBR system. Competence is not reduced because if one case is deleted then a nearby case can be used to solve any target that the deleted auxiliary could solve.

The coverage spaces of *spanning cases* span regions of the problem space that are independently covered by other cases. If cases from these linked regions are deleted, then the spanning case may be necessary. In general, they do not directly affect the competence of the system.

*Support cases* are a special class of spanning cases. They exist in groups, each support providing coverage similar to the others in a group. They also do not affect competence directly. While the deletion of any case (or any proper subset) of a support group does not reduce competence, the removal of the group as a whole is analogous to deleting a pivot and does reduce competence.

A case is called a *pivotal case* if its deletion directly reduces the competence of the system (irrespective of the other cases in the case base). Using the above estimates of coverage and reachability, a case is pivotal if it is reachable by no other case but itself.

The above-mentioned case categories provide a means of ordering cases for deletion in terms of their competence contributions. The auxiliary cases are the least important as they make no direct contribution to competence; next are the support cases, then the spanning cases and, finally, the pivotal cases. The following sections of the paper focus on the last-mentioned of these categories.

The implementation of CBR phases is determined by the *domain* of application, for example, engineering, medicine, or business. For example, medical diagnosis may be considered a simple classification task or a complicated reasoning problem in which one deals with incomplete information that requires to be supplemented by redefinition (e.g., extension) of the cases during the repetition of CBR cycles.

The CBR-based approach may be useful in several *task types*, such as information retrieval, planning, design, and

classification. The discussion presented in Section 4 focuses on the latter type.

### 3. Outlier Case

In the literature, there is no single, universally applicable definition of the term outlier, since the formulation of such a definition depends largely on a particular area of application. Thus, the term *outlier* is used to refer to a multitude of concepts, as reflected by the following definitions:

- (i) An outlier is an observation which deviates so much from the other observations to arouse suspicions that it was generated by a different mechanism [15].
- (ii) Outliers are noise points lying outside the set which defines the clusters, or alternatively, outliers can be defined as points lying outside the set of clusters but are separated from the noise [11].
- (iii) An outlier is an observation which deviates so much from the other observations to arouse suspicions that it was generated by a different mechanism [12].
- (iv) An observation (or subset of observations) appears to be inconsistent with the remainder of that set of data [14].
- (v) A point  $p$  in a data set is an outlier with respect to the parameters  $k$  and  $\lambda$ , if no more than  $k$  points in the data set are at a distance  $\lambda$  or less from  $p$  [13].
- (vi) Let  $X = \{x_1, x_2, \dots, x_N\}$  for  $N \in \mathbb{N}$  be a finite, non-empty set of objects. Let  $S$  be a finite, nonempty set of attributes (features) of the set of objects  $X$ :  $S = \{s_1, s_2, \dots, s_n\}$ . Then a subset of objects  $X_{\text{out}} \in X$  will be called outliers in the set  $X$  if and only if for any subset of attributes  $s_i \in S$ . The cardinality of subset  $X_{\text{out}}$  is determined by the linguistic quantifier  $Q$ , that is, “little,” “few,” “very few,” “very little,” “almost no,” and the like [20].

The last ten years have seen increasingly rapid advances in the field of outlier detection, and a variety of outlier detection methods have been proposed, for example, [17, 21–27]. In general, two main approaches to this problem may be distinguished. One way seeks to develop innovative outlier detection algorithms assuming the general definition of an outlier. The other approach, regardless of the application domain, is to employ similar or even the same algorithms, while considering different definitions of the outlier.

For the CBR technique, the following three definitions of outlier case  $c_{\text{out}} = (p, \text{sol})_{\text{out}}$  or  $c_{\text{out}} = (p, \text{sol}, \text{eff})_{\text{out}}$  can be considered:

- (1). An *outlier* can be in general understood as a *pivotal case* (cf. Section 3). Formally, it is defined as follows [8–10]:

$$c_{\text{out}} = \text{pivot}(c), \quad \text{iff} \quad \text{reachable}(c) - \{c\} = \emptyset. \quad (8)$$

Outliers are too isolated to be solved by any other case.

The inconsistency criterion leads to the following definition:

- (2). Outlier case  $c_{\text{out}}$  is understood as the case that appears to be inconsistent with the other cases of  $S_c$ . The inconsistency is due to the process of internal assessment or final revision.

When the distance has been defined, the following definition may also be employed:

- (3). A case  $c_i$  in a case base is called outlier  $c_{\text{out}}$  with respect to the parameters  $k$  and  $\lambda$ , if no more than  $k$  cases in that base are at a distance  $\lambda$  or less from  $c_i$ . It is assumed that values  $k$  and  $\lambda$  confirm the claim about outlieriness.

The practical result of finding an outlier is that no useful modification of the solution of the nearest neighbour is possible within the CBR system, which means that the system will not generate an effective, satisfying, and useful solution. In this case, it is necessary to find a solution outside the system and then add it as a new case to the  $S_c$  case base.

It is also possible to verify if

- (4). Outliers are some of the cases included in  $S_c$  which were never or hardly ever adapted by a large number of uses of the CBR system.

However, the above definition (4) is just an observation concerning the work of the CBR system and gives no insight into the nature of the cases under examination, which in fact do not have to represent outliers (items possessing anomalous features).

#### 4. Case-Based Classification (CBC)

A *classifier* is a function which transforms  $S$  into  $K$ , where  $K$  denotes the number of subsets  $S_k$  identified in  $S$ ; that is,  $S_k \in S, k \in K$ . In other words,  $K$  can be understood as the number of labels which can be assigned to objects in  $S$ . For a case-based classifier, the following notation is used:

$$(S_c, \text{sim}), \quad (9)$$

where  $S_c \subset S$ , while  $\text{sim}$  is defined on  $S \times S_c$ .

The class of a new object  $c_i$  is determined using the defined form of  $\text{sim}$  assuming that other objects used for comparison are already labelled. Usually, the nearest labelled neighbour is sought or another similar approach is applied; for example,  $k$  most similar cases are found and voting for the choice of a proper neighbour is performed. A new case in *case-based classification* is given by the description of an object (problem), and the goal is to assign the correct label (solution) to this object. In CBC, case  $c_i = (p, \text{sol})_i$  as defined in (1) is determined entirely by the problem, because the label (class) is uniquely assigned to the object (multilabel classification is assumed to be beyond the scope of the present discussion). In other words, if  $k$  is identified as the label (class) of case  $c_i$ , then  $c_i \in S_k$ .

Within the CBC, it is assumed that if for two cases  $c_i = (p, \text{sol})_i$  and  $c_j = (p, \text{sol})_j$  the problems' descriptions  $p_i$  and  $p_j$  are similar, then both cases  $c_i$  and  $c_j$  can be assigned the same class or similar classes. However, the notion *similarity of the classes* must be defined.

It should be mentioned that within the CBR (CBC), learning can be performed, and thus, an initially approximate classifier function can be adjusted. The learning can be performed by modifying the similarity (or distance) measure or by supplementing the case base with new instances.

The characteristic stage of any CBR system is *case adaptation*. With the CBC, the procedure tends to be simpler. For example, if the retrieved similar case is the nearest labelled neighbour, then its solution is the best-known one, and the new solution can be proposed only by performing an external validation, called *revision*. In some situations, this can lead to the introduction of a new label  $k$  and consequently extension of set  $K$ .

When we seek for the cases that lie at the greatest distance from the already labelled numerous and dominating group of cases, then two situations are possible:

- (i) The case is a single one, and an introduction of a new outlier class is necessary.
- (ii) The case belongs to one of the existing outlier clusters.

However, when working with preference relation (5), one looks for objects which do not belong to the dominating class of many similar cases. Such objects are called outliers, and in some situations, they may require the introduction of a new label  $k$ .

As an example, let us consider the CBC-supported medical diagnosis. The term "supported" needs to be emphasized, because the CBC system only suggests the possible diagnosis, and the final statement falls exclusively within the competence of the physician.

Let the case examined by the system be of the following form:

$$c_i = (p, \text{sol})_i, \quad (10)$$

where  $p$  and  $\text{sol}$  denote the sets of symptoms and diagnosis, respectively. The solution  $\text{sol}_p$  proposed by the supporting system as the nearest neighbour must be revised by the physician. It may be the case that another solution  $\text{sol}_{co}$  is indicated by the expert as the correct one. The next step is to verify if the case  $c = (p, \text{sol}_{co})$  already exists. If not, it needs to be included into the case base of the CBC system. If it lies at a great distance from the already classified cases, it may be referred to as an outlier.

#### 5. Selected Computational Approaches

Classification as a method of supervised learning uses labelled observations, with the labels taking nominal values. The purpose of learning is to create a classifier that will assign objects to classes.



The model of the classifier is created according to the pattern of decision classes, most often prepared by an expert. Outliers are deviations from this model. There are many classification methods available, such as decision trees, probabilistic models such as the Bayesian classifier,  $k$ -nearest neighbour algorithm, support vector machine, and neural networks, to name a few. Let us examine the outliers in the data set shown in Figure 2. It can be easily noticed that the data belong to the two classes highlighted in green and blue. Note that points A, B, and C lie at a great distance from the rest of the objects. The classification of this data set using the  $k$ -NN classifier may result in assigning object A to the blue class, object B to the green one, while object C, depending on the number of neighbours of the  $k$ -NN algorithm, may be assigned to the green or to the blue one. In any case, this results in an increased classification error.

A crucial issue related to classification-based outlier detection is the selection of an appropriate classification model. This is a difficult task due to the rarity and atypical character of the feature vector which describes the outlier. While building a classification model, the expert determines decision functions for a particular set of features (a set which is known and often occurring). According to the definitions proposed by Hawkins [15] or Aggarwal [12], the outlier is a vector of atypical and rare characteristics that are not foreseeable for an expert. Therefore, there is a problem with class imbalance or lack of indication of the class with features that point to the existence of outliers.

Another problem associated with the classification of outlying objects is the lack of possibility to balance classes. The layered sampling technique does not provide equivalence of classes. It does not perform well in the case of outliers. There are few outlying objects in the whole set. In the process of layered sampling, the records are first separated according to their classes, and the classes with a small number of objects are selected. In the next step, the objects of the dominant class are randomly selected (the class is regarded as dominant if the majority of objects in the analyzed set belong to it). Yet, an object may appear that has not been assigned to any class.

The characteristic stage of the CBR working cycle, as described in Section 2, is case adaptation. Within CBC, the situation is simple. The assignment to the class occurs as a result of the defined similarity or distance function. For example, if a similar case is found and it is the closest labelled neighbour, its solution is known, and the only way to propose a new solution is by performing external validation. In some situations, this may lead to the introduction of a new label  $k$  and, consequently, the extension of the set of classes.

There are many different ways to construct CBC classes. For example, one can

- (i) consider the distance between objects;
- (ii) determine the number of neighbours that should be used for prediction;
- (iii) use an additional weight or include all variables as equally important in the classification;

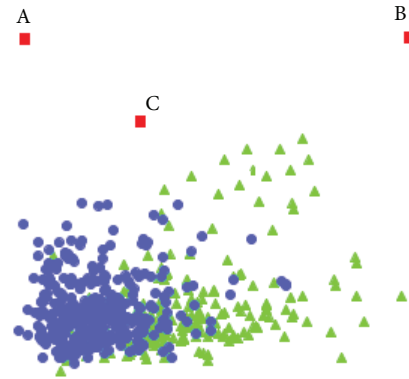


FIGURE 2: Example of objects A, B, and C being distance outliers.

- (iv) apply a specific kind of standardization.

One way to approach outlier identification is to employ a binary statement about whether an object is an outlier or not. This method relies on the subjective opinion of the expert. Another way is to estimate and determine the degree to which the indicated object is an outlier. According to Aggarwal [12], the most interesting observations are those for which the degree of dissimilarity is the highest.

In classification tasks, two approaches to outlier detection can be distinguished, namely, the statistical approach and the approach based on the distance measure between objects. Relating this to the CBC approach, solution  $sol$  of problem  $p$  can be determined either by the use of a properly defined density function or by the use of a chosen similarity measure. In other words, the search is performed for objects that are at the maximum distance from the already labelled objects, that is, those that are least probable.

The following two approaches are possible:

- (A) Determination of outliers without preliminary analysis of the considered set of cases

- (B) Two-stage procedure:

Stage 1: dividing the analyzed set of cases into subgroups on the basis of an additional classification criterion (e.g., the medical criterion: healthy or ill);

Stage 2: determination of outliers

In both (A) and (B), the chosen classification method is applied, that is, distance-based outlier detection, Bayesian classifier, and linear regression by calculating Cook's measure.

The statistical approach is directly related to the probability distribution. It assumes that the values of objects in the analyzed set have a specified probability distribution. The objects for which the values of attributes deviate from the distribution are referred to as outliers. In this case, one can specify

- (i) nonconformity tests for different probability distributions;
- (ii) tests for known or unknown values of probability distribution parameters (distribution characteristics such as mean and standard deviation);
- (iii) others

**5.1. Regression for Outlier CBR Search.** The statistical approach has certain limitations. The tests conducted pertain to a single attribute and, therefore, are not very useful or appropriate for multidimensional data. An additional difficulty may be the complexity and cost of the performed calculations related to the estimation of unknown parameters of polynomial probability distributions. For a more profound discussion, the reader is referred to [12, 14, 15, 28].

In the statistical approach, the so-called loss function is introduced, which enables the calculation of the cost of the classifier's error (mistake). An example of a loss function might be in the form of (11), where 1 means an incorrect decision, while 0 denotes a correct one.

$$L(r, s) = \begin{cases} 1, & r \neq s, \\ 0, & r = s. \end{cases} \quad (11)$$

The main idea of regression is to determine the vector of weights of each independent variable in order to minimize errors. In the regression model, the original independent variables are transformed into independent weighed variables.

Given the notation introduced in Sections 2 and 4, the CBR for the classification of data using regression should be considered as follows: the notation given in Section 3 can be defined as follows:

- (i)  $p$  (problem)—to find the best distinction between objects of different classes
- (ii)  $sol$  (solution)—to determine the values of regression parameters in such a way to enable the best adjustment of the values to a given data set
- (iii)  $eff$  (effect)—the influence of the object on the regression model

The comparison of the two cases of  $c_k$  and  $c_i$  in the regression model consists in predicting the difference in the output attributes between them according to

$$y_i - y_k = a_1(x_{i1} - x_{k1}) + a_2(x_{i2} - x_{k2}) + \dots + a_n(x_{in} - x_{kn}). \quad (12)$$

We can distinguish the following types of exceptional cases:

- (i) Case of  $c_k$  is an exceptional case  $c_{out}$  if in a matrix of differences between two successive values of attributes for the compared cases, there are values different from 0 or greater than the set threshold  $pr$ .
- (ii) Case of  $c_k$  is an exceptional case  $c_{out}$  if there has been a significant change in the regression model

coefficient and the estimated measures DFFITS, DFBETAS, and Cook (or even one of them) take values above the determined threshold  $eff$ .

The regression model is influenced by the so-called high leverage points, which do not necessarily correspond to outliers. In the case of a regression-based classification, outliers are detected on the basis of measures which determine the impact of a given object on the regression model used. For example, Cook's measure determines the level of influence of an object on the model by calculating the squares of the difference between the predicted values of the response variable across the whole sample (the whole set) and the values in the model where the  $i$ th observation ( $i$ th object) was omitted.  $eff$  can be defined as Cook's measure based on Cook's equation (13).

$$D_i = \frac{(y_i - \hat{y})^2}{ps^2} \frac{h_i}{(1 - h_i)^2} = \frac{e_i^2}{pMSE} \frac{h_i}{(1 - h_i)^2}, \quad (13)$$

where  $D_i$  is the residual of  $i$ th observation,  $p$  is the number of parameters in the model,  $h_i$  is the influential value of this observation,  $s$  is the standard error of the estimator, and MSE is the average square error. Factors  $e_i^2/pMSE$  and  $h_i/(1 - h_i)^2$  are called the measure of variability and the measure of the leverage of a given observation, respectively.

The high  $eff$  value, which is Cook's measure as defined by (13) (value of  $D_i > 1$  is considered high), indicates that the deletion of the  $i$ th observation from the population has a strong influence on the regression model and, thus, that observation is considered to be influential. Other popular measures that determine the impact of an outlying object on a regression model are DFFITS (difference in FITS) and DFBETAS (difference in betas).

An object is an outlier if for a small sample the value  $eff = |DFFITS_i|$  or  $eff = |DFBETAS_j(i)|$ ,  $|DFFITS_i|$ , and  $|DFBETAS_j(i)|$  is greater than 1.0. For a large sample, an object is an outlier if the value  $|DFFITS_i|$  exceeds  $2\sqrt{p/n}$  and  $|DFBETAS_j(i)|$  exceeds  $2/\sqrt{n}$ . More details can be found in [6].

Depending on the measure adopted to determine the influence of a given object on the regression equation, the  $eff$  value must be greater than 1 or, for large samples, greater than  $2\sqrt{p/n}$  or  $2/\sqrt{n}$ .

**5.2. Bayes Outlier Case-Based Model.** The Bayesian CBR model defines cases according to (1) and (2), as introduced in Section 2. The case is defined by problem  $p$  and solution  $sol$ . The problem  $p$  is a description of objects with characteristic features, for example, a collection of dishes or food products. The solution  $sol$  is an allocation of an object to a class, a quintessential observation that it is the best representation of the class. The  $sim$  function defining the similarity of the objects is defined by the density function. The new object to be classified belongs to the  $i$ th case if the density function is the largest. For the outlier case, the probability function obtained does not indicate the maximum value, but the smallest probability.

Let us consider problem  $p$  described using the attributes.

$A = \{A_1, A_2, \dots, A_m\}$ ,  $m \in N$ . The Bayesian case-based classifier assigns the label  $k$  for the case of  $c_i$  ( $c_i \in S_k$ ). The case  $c_i$  is a “prototype” representation of the class (in a sense) of similar observations and is encoded as the vector:

$$c_i = (P_i(a_{11}), \dots, P_i(a_{1m}), \dots, P_k(a_{m1}), \dots, P_k(a_{mn})), \quad (14)$$

where  $P_i(a_{ij})$  expresses the probability that the  $A_i$  attribute has the value of  $a_{ij}$  in the class  $k$ .

Of course, the  $c$  case database consists of  $t$  cases  $c_1, \dots, c_t$ , each of which is provided with a unique  $c_k$  label. Initially, cases are defined by an expert (alternatively, they may come from a large observation database using statistical clustering methods).

The designated conditional probability (a posteriori)  $P(c_k | X)$  means that the object  $x_i \in X$  is classified into the case  $c_{kj} \in K$ .

Let  $X = \{x_1, x_2, \dots, x_N\}$  be the set of objects and  $k = \{1, \dots, p\}$ . Let the distribution of objects be a discrete probability distribution or probability density  $P(x | k) \equiv f_k(x)$ . Let us introduce the following designations:

- (i)  $P(K)$ —unconditional probability (a priori) of the occurrence of the case  $K$
- (ii)  $P(X | K)$ —conditional probability, where the object  $X$  belongs to the case  $K$
- (iii)  $P(X)$ —unconditional probability of the occurrence of the object  $x_i$ .

$$P(K|X) = \frac{P(K) * P(X|K)}{P(X)}. \quad (15)$$

An object belongs to the case  $c_k$  if it fulfils the maximum likelihood principle or the maximum a posteriori principle. The maximum likelihood principle (ML) selects the case  $c_{kj} \in K$ , which maximizes the conditional probability of the given objects  $o \in OT$  (OT objects used as training data).

$$K_{ML} = \arg \max P(O|k). \quad (16)$$

The maximum a posteriori principle (MAP) consists in selecting the case  $c_{kj} \in K$  with the maximum probability a posteriori:

$$K_{MAP} = \arg \max P(k|O). \quad (17)$$

The case receives a new label (the outlier label) if its  $c_{out} = (p, sol)_{out}$  is the same as for at least two other different cases. The maximum likelihood principle or the maximum a posteriori principle is not met.

The case receives a new label (the outlier label) if the probability for each previously defined case is much smaller than the threshold assumed by the expert; for example, the value is smaller than 25% of the value of the smallest probability determining the given case.

For the classifiers based on the probability theory (especially for the Bayes classifiers), it is possible to introduce classification weights, which have an impact on the a

priori probability value of the decision classes. The other estimates remain unchanged [12]. A special case occurs when the highest probability a posteriori is obtained for several classes. In this situation, it is not possible to unambiguously state to which class the object should be classified. In addition, according to [12, 13, 29, 30], it is not in any case justified to assign an object to the class with the highest probability a posteriori. The authors then propose a classification threshold.

*Example 5.1.* Let  $o_i$  denote objects  $O = \{o_1, o_2, \dots, o_n\}$  for  $n \in N$  and  $r, s$  be classes to which we assign new objects. If for object  $o_i$  the estimated probability for class  $r$  is 0.6 and for class  $s$  is 0.4, according to Bayer’s rules of the classifier, the  $o_i$  object is assigned to class  $r$ . However, if the threshold for class  $s$  is 0.35, then the object  $o_i$  is assigned to class  $s$ .

*5.3. Distance-Based Outlier Detection.* Another way to detect outliers is to calculate the distance between objects according to a selected measure.

Taking into account the denotations introduced in Section 3 and definitions for detecting outliers using the distance-based algorithm, we have the following:

- (i)  $p$ —a problem, that is, the division of objects into  $c_i$  classes
- (ii)  $sol$ —a solution that assigns an object to a class
- (iii)  $dis$ —the distance between two objects

Assigning a new object to a given case takes place after the distance of that case to the labelled cases is determined. Case  $c_i$  gets the  $c_{out}$  outlier label if the distance of this case to the other cases exceeds the designated  $d_{tc}$  threshold.

In most data classification tasks (similar to those described above), the detection of outlying objects is based on the distance threshold criterion ( $d_{tc}$ ). If the distance of object  $o_i$ , defined by the expert, to object  $o_k$  is greater than the specified threshold, then object  $o_i$  is considered as an outlier.

$$dis(o_i, o_k) > d_{tc}. \quad (18)$$

These objects may represent unusual and previously unknown behaviours or operations. They have a small number of neighbours. They are not removed from the set and still participate in the data analysis but are considered outliers.

We can also say that object  $o_i$  is a distance-based outlier in the data set  $O = \{o_1, o_2, \dots, o_n\}$ ,  $n \in N$  if and only if the distance of at most  $proc$  percentage of objects from set  $O$  is smaller than the distance  $dis$  an equation (19) is true where  $d(o_k, o_i)$  is the measure of the distance between objects  $o_k$  and  $o_i$ .

$$\frac{||o_i|d(o_k, o_i)| \leq dis|}{|O|} \leq proc. \quad (19)$$

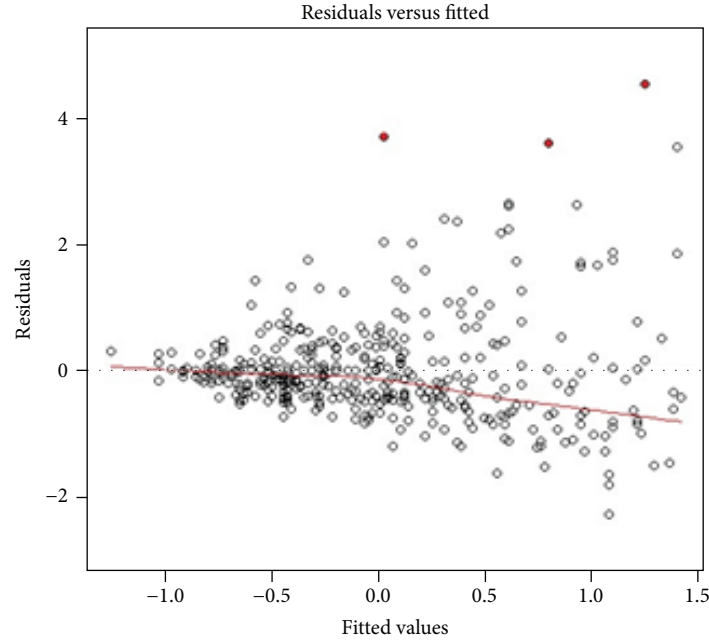


FIGURE 3: Graph of dependencies between the regression model for the original data set from [32] and residual values with three outliers market.

In terms of the CBC approach, the new case  $c_i$  is assigned to a known class if  $\text{dis}$  is the smallest or receives an outlier label if  $\text{dis}(o_i, o_k) > \text{dte}$  or (19) is true.

The classification-based distance outlier detection may be affected by difficulties due to a high number of dimensions. As the dimensionality increases, all objects are situated at a similar distance to each other. It may be the case that the distance between the object and its nearest neighbour approaches the distance to the furthest neighbour. Therefore, all parameters must be carefully selected. The essential advantage of using distance measures in outlier detection is the fact that it does not require a priori knowledge of the probability distributions.

Figure 2 also highlights the division of distance outliers into global distance outliers and local distance outliers. Objects A and B are global distance outliers because their distance from objects in the whole set is great. Object C can be considered in terms of its isolation degree relative to the nearest neighbourhood, that is, the object from the blue class which is closest to object C and the object from the green class which is closest to object C. Then, the local outlier factor (LOF) is determined. More details on this can be found in [31].

**5.4. Evaluation.** The evaluation of the performance of both methods was based on a mean square error and a matrix of errors. Cases of correct classification, that is, TP (true positive) and TN (true negative) as well as cases of incorrect classification, that is, FP (false positive) and FN (false positive), were taken into consideration in the matrices of errors. Sensitivity (SE), specificity (SP), and the accuracy were calculated, according to (20). The detection error was determined

as the ratio of the number of misclassifications to the sum of all detections.

$$\begin{aligned} \text{SE} &= \frac{\text{TP}}{(\text{TP} + \text{FN})}, \\ \text{SP} &= \frac{\text{TN}}{(\text{TN} + \text{FP})}, \\ \text{ACC} &= \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})}. \end{aligned} \quad (20)$$

It should be noted that a large number of FP or FN contribute to an increase in the classification error. The consequence of FP detection is the detection of outliers. This may also be the reason for creating a class with a new pattern.

## 6. Practical Example

**6.1. Classic Methods.** The experimental research was carried out using the benchmark (repository) database [32], which originally contained 868 records. The data collected included information concerning blood glucose, glucose (plas), blood pressure- (pres-) diastolic blood pressure, skin thickness (skin)—thickness of skin on triceps (mm), age, weight, BMI, pregnancies (preg)—the attribute stating the number of pregnancies of the patient, and inheritance risk ratio—the factor of the risk of inheriting diabetes. Over 2000 records were taken into account. The data set was examined for the presence of outliers using the classic regression method (cf. Figure 3). Three cases of outliers were detected in the set under examination using Cook's measure.

To make the experiments more reliable, the data set was extended by 1200 new records in which 9 known outliers



TABLE 1: Best results of outlier detection using classic methods.

Classic	Number of detected outliers	Percentage of correct detections (%)
Regression	7	58
Bayes	6	50
$k$ -NN	5	42

TABLE 2: Measures of rating classic methods.

Method	SE sensitivity	SP specificity	ACC accuracy	Classification error
Bayes	0.69	0.52	0.31	0.35
$k$ -NN	0.67	0.41	0.3	0.42

were incorporated. The resulting total number of data records was 2077, in which 12 known outliers were hidden.

The results obtained by three classic methods used for comparison are collected in Table 1. Measures of rating are shown in Table 2.

In the literature [33] referring to the  $k$ -NN method, the following formula is recommended for determining the optimal value of parameter  $k$ :  $k^* = \sqrt{N}$ , where  $N$  is the number of cases chosen to learn.

However, this recommended value  $k^* = 21$  did not work for the database examined and only 1 or 2 outliers were detected. In general, each value bigger than 10 was unsatisfactory. The best results were obtained for  $k = 5$  and  $k = 6$ , which were determined experimentally.

In the case of the  $k$ -nearest neighbour classifier, outliers are the objects whose distance from the nearest neighbour is much greater than that for the other objects. Thus, it is possible to specify that the distance between objects cannot be greater than the distance given by the expert. Figure 4 shows an example data dispersion where the circles are healthy persons, pluses (crosses) indicate the class of healthy people, and the rhombuses represent the outliers. The distance of the nearest neighbour in the case of 5 points is much higher than that in the other cases. Therefore, these points are likely to be classified as false positives or false negatives. This leads to an increased classification error.

**6.2. Regression CBC: Cook's Measure.** In the case of the CBR method with the use of linear regression, Cook's measure was applied to estimate the level of influence of the object on the model.

The outlying objects indicated by Cook's measure are shown in Figure 5. Cook's values above the line indicate the existence of outliers in the analyzed set. The graph of dependencies between the CBC regression model and residual values for 2077 cases with 10 outlier cases marked is shown in Figure 6.

**6.3. Naive Bayesian Classifier CBC.** The naive Bayesian classifier, as a probabilistic classifier, estimates the frequency of occurrence of objects with specified parameters for each class. In our case, outliers occur very rarely. Thus, it is

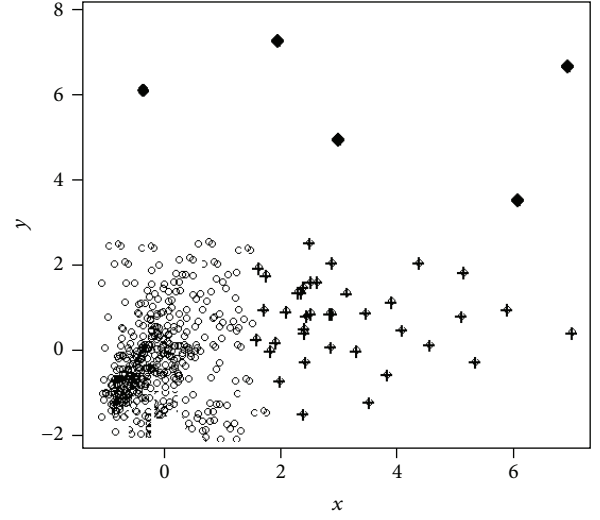
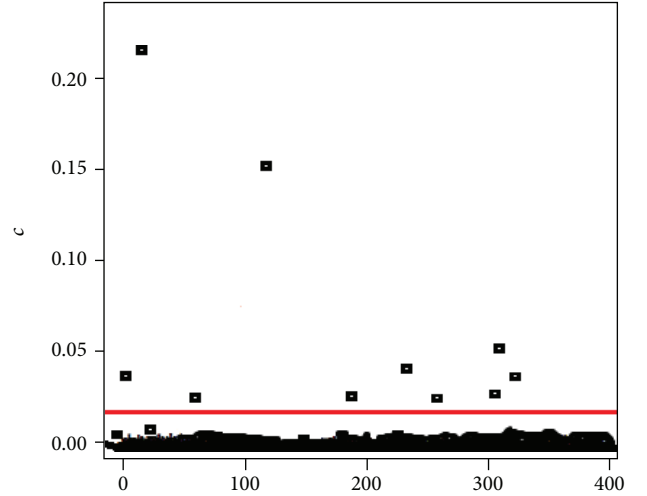
FIGURE 4: Outliers clearly distancing from the other objects—the result obtained by  $k$ -NN ( $k = 5$ ).

FIGURE 5: Outliers indicated using Cook's measure.

difficult to speak of the determination of occurrence frequency. It is not always helpful to use Laplace's expansion. In addition, the naive Bayesian classifier assumes that the total density of objects is a product of boundary densities. The testing of the CBR method with the Bayes classifier consisted of two stages. The classification was performed for all cases under consideration, that is, the whole given data set. Due to the fact that there were outliers in the analyzed database, the value of the classification error obtained was 0.27 (see Table 3).

The classification error decreased after using the Bayes outlier case-based reasoning method, in which a separate class of outlier cases was initially found, without preliminary classification. The results obtained using four evaluation measures are summarized in Table 3.

The classification error decreased after using the Bayes outlier case-based reasoning method, in which a separate class of outlier cases was initially found, without preliminary



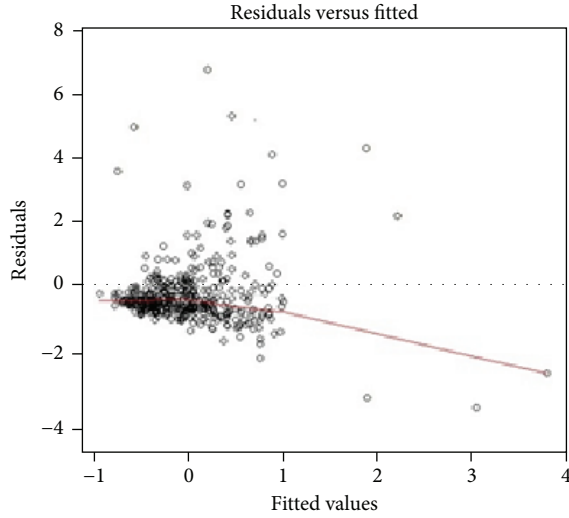


FIGURE 6: Position of ten outliers detected in the extended database.

TABLE 3: Measures of rating classifiers for Bayes case-based reasoning for two approaches (A) and (B).

Bayes	SE sensitivity	SP specificity	ACC accuracy	Classification error
A	0.75	0.66	0.27	0.27
B	0.8	0.7	0.23	0.21

TABLE 4: Measures of rating distance-based reasoning.

Distance- based CBC	SE sensitivity	SP specificity	ACC accuracy	Classification error
A	0.81	0.79	0.26	0.6
B	0.77	0.56	0.30	0.28

classification. The results using four evaluation measures are summarized in Table 3.

**6.4. Distance-Based CBC.** The distance-based classification, like the Bayesian classification, was used in two stages (A) and (B). The results are given in Table 4 and Figure 7.

## 7. Summary

The paper has presented the application of case-based reasoning (CBR) and case-based classification (CBC) to the problem of outlier detection. The formal definition of case outlier has been introduced. The study has demonstrated a CBC framework for the interpretation of several classification approaches. The method proposed here was validated using a practical example from the field of medicine.

The results obtained using the CBR approach, which are described in detail in Section 5, were significantly better than classic methods. For example, the graph of dependencies between the designated CBC regression model and residual values for 2077 cases with 10 outliers marked is shown in Figure 6. Better results were obtained also using the Bayes

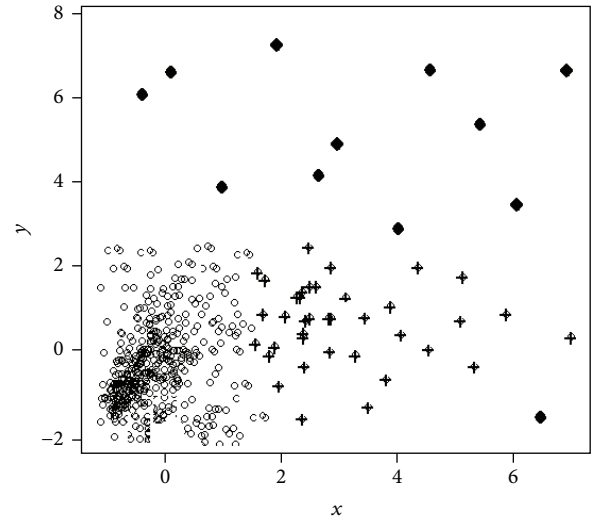


FIGURE 7: Outliers detected for distance-based CBC.

TABLE 5: Best results of outlier detection using CBR (CBC) approach.

CBR (CBC)	Number of detected outliers	Percentage of correct detections (%)
Regression	10	83
Bayes	11	92
Distance-based CBC	12	100

CBC method and distance-based CBC (Figure 7). The complete set of results is collected in Table 5.

As can be seen from Tables 3, 4, and 5, the presence of outliers in the data set makes the classification much more difficult. The implementation of CBR without creating a subgroup of cases resulted in the decrease in sensitivity and accuracy, while increasing the classification error.

The application of the case-based reasoning approach resulted in the classification error decreasing by 0.06 and 0.32 for the Bayes classifier and the distance-based CBC, (Tables 3 and 4), respectively. It should be emphasized that the strongest resistance to the occurrence of outliers was demonstrated by the Bayes outlier case-based classification. However, the results produced of the Bayes and distance-based classification method are of similar quality.

## Nomenclature

S:	The universe of all objects
$S_c$ :	The case base
$c_i$ :	The $i$ th case, $i \in I$
sim:	Similarity (formal definition is the appropriate approach)
dis:	Distance
$p$ :	Problem
sol:	Solution
eff:	Effect.

## Data Availability

In the research, public repositories were used, and they are available on the website <http://archive.ics.uci.edu/ml/>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] M. Lenz, B. Bartsch-Spörl, H. D. Burkhard, and S. Wess, *Case-Based Reasoning Technology: From Foundations to Applications. Volume 1400*, Springer, 2003.
- [2] S. K. Pal, T. S. Dillon, and D. S. Yeung, *Soft Computing in Case Based Reasoning*, Springer Science & Business Media, 2012.
- [3] P. Perner, *Case-Based Reasoning on Images and Signals. Volume 73*, Springer, 2008.
- [4] K. D. Althoff, R. Bergmann, S. Wess et al., "Case-based reasoning for medical decision support tasks: the inreca approach," *Artificial Intelligence in Medicine*, vol. 12, no. 1, pp. 25–41, 1998.
- [5] A. S. Ochi-Okorie, "Disease diagnosis validation in TROPIC using CBR," *Artificial Intelligence in Medicine*, vol. 12, no. 1, pp. 43–60, 1998.
- [6] D. A. Belsley, E. Kuh, and R. E. Welsch, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. Volume 571*, John Wiley & Sons, 2005.
- [7] C. Globig and S. Wess, "Learning in case-based classification algorithms," in *Algorithmic Learning for Knowledge-Based Systems. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*, vol. 961, K. P. Jantke and S. Lange, Eds., pp. 340–362, Springer, Berlin, Heidelberg, 1995.
- [8] B. Smyth and M. T. Keane, "Remembering to forget," in *IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence*, pp. 377–382, Montreal, Quebec, Canada, August 1995.
- [9] B. Smyth and M. T. Keane, "Footprint based retrieval," in *Case-Based Reasoning Research and Development. ICCBR 1999. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*, K. D. Althoff, R. Bergmann, and L. Branting, Eds., pp. 134–148, Springer, Berlin, Heidelberg, 1999.
- [10] M. M. Richter, R. O. Weber, and C. B. Reasoning, *A Textbook. Organic Chemistry*, John Wiley & Son, Inc, New York, NY, USA, 2013.
- [11] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *SIGMOD '01 Proceedings of the 2001 ACM SIGMOD International Conference on Management of data*, pp. 37–46, Santa Barbara, CA, USA, May 2001.
- [12] C. C. Aggarwal, *Outlier Analysis*, Springer Science & Business Media, 2013.
- [13] E. M. Knorr and R. T. Ng, "A unified notion of outliers: properties and computation," in *KDD'97 Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 219–222, Newport Beach, CA, USA, August 1997.
- [14] V. Barnett and T. Lewis, *Outliers in Statistical Data. Volume 3*, Wiley, New York, NY, USA, 1994.
- [15] D. M. Hawkins, *Identification of Outliers. Volume 11*, Springer, 1980.
- [16] A. Duraj, P. S. Szczepaniak, and J. Ochelska-Mierzejewska, "Detection of outlier information using linguistic summarization," in *Flexible Query Answering Systems 2015. Advances in Intelligent Systems and Computing*, pp. 101–113, Springer, 2016.
- [17] A. Duraj and P. S. Szczepaniak, "Information outliers and their detection," in *Information Studies and the Quest for Trans-disciplinarity. Volume 9, Chapter 15*, M. Burgin and W. Hofkirchner, Eds., pp. 413–437, World Scientific Publishing Company, 2017.
- [18] I. Jurisica, J. Mylopoulos, J. Glasgow, H. Shapiro, and R. F. Casper, "Case-based reasoning in IVF: prediction and knowledge mining," *Artificial Intelligence in Medicine*, vol. 12, no. 1, pp. 1–24, 1998.
- [19] J. N. Mordeson, D. S. Malik, and S. C. Cheng, *Fuzzy Mathematics in Medicine*, Springer-Verlag New York, Inc, 2000.
- [20] A. Duraj, A. Niewiadomski, and P. S. Szczepaniak, "Outlier detection using linguistically quantified statements," *International Journal of Intelligent Systems*, vol. 33, no. 8, pp. 1590–1601, 2018.
- [21] A. Duraj and L. Chomatek, "Outlier detection using the multi-objective genetic algorithm," *Journal of Applied Computer Science*, vol. 25, no. 1, pp. 29–42, 2017.
- [22] A. Duraj and D. Zakrzewska, "Effective outlier detection technique with adaptive choice of input parameters," in *Intelligent Systems'2014. Advances in Intelligent Systems and Computing*, pp. 535–546, Springer, 2015.
- [23] A. Duraj and L. Chomatek, "Supporting breast cancer diagnosis with multi-objective genetic algorithm for outlier detection," in *Advanced Solutions in Diagnostics and Fault Tolerant Control. DPS 2017. Advances in Intelligent Systems and Computing*, J. Kościelny, M. Syfert, and A. Szyber, Eds., pp. 304–315, Springer, 2017.
- [24] M. Kalisch, M. Michalak, M. Sikora, Ł. Wróbel, and P. Przysławka, "Data intensive vs sliding window outlier detection in the stream data an experimental approach," in *Artificial Intelligence and Soft Computing. ICAISC 2016. Lecture Notes in Computer Science*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds., pp. 73–87, Springer, 2016.
- [25] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 5, pp. 1369–1382, 2015.
- [26] G. O. Campos, A. Zimek, J. Sander et al., "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.
- [27] C. Titouna, M. Aliouat, and M. Gueroui, "Outlier detection approach using bayes classifiers in wireless sensor networks," *Wireless Personal Communications*, vol. 85, no. 3, pp. 1009–1023, 2015.
- [28] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Data Warehousing and Knowledge Discovery. DaWaK 2002. Lecture Notes in Computer Science*, Y. Kambayashi, W. Winiwarter, and M. Arikawa, Eds., pp. 170–180, Springer, Berlin, Heidelberg, 2002.
- [29] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.
- [30] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal*, vol. 8, no. 3–4, pp. 237–253, 2000.

- [31] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, Dallas, TX, USA, May 2000.
- [32] C. J. Merz and P. M. Murphy, *{UCI} Repository of Machine Learning Databases*, 1998.
- [33] G. G. Enas and S. C. Choi, “Choice of the smoothing parameter and efficiency of  $k$ -nearest neighbor classification,” in *Statistical Methods of Discrimination and Classification*, pp. 235–244, Elsevier, 1986.