Proceedings of the first

Scientific Workshop

On

Computer Science and New Technologies

CSTL Lab, University Of Mostaganem

Program Committee Chairs

Pr. SEHABA Karim

Pr. ATMANI Baghdad

Organizing committee Chairs

Dr. HENNI Fouad

Dr. DJEBBARA Rédha

Mostaganem, 8 December 2024

Table of contents

Computer Science and New Technologies Workshop

Call of papers of the Scientific Workshop

The CSNT'2024 workshop is part of the National Artificial Intelligence Plan 2020-2030. Initiated by the CSLT laboratory at the University of Mostaganem, the aim of this event is to stimulate research, innovation and development in artificial intelligence (AI) in Algeria, and to help consolidate the efforts of a dynamic and committed community in this field.

Based on the National Reference Framework of Research Priorities, the specific objective of this edition is to present the latest advances in AI research in relation to the country's socio-economic objectives. The aim is to bring together experts, researchers, professionals, students and anyone interested in the field of artificial intelligence to discuss, present and exchange ideas, discoveries, technological advances, challenges and opportunities.

All scientific, technological or methodological contributions, whether under development or completed, are likely to be presented, in particular, and in a non-exhaustive manner, on the themes of machine learning and deep learning, data science, massive data, knowledge representation and management, the Internet of Things, etc. The fields of application may relate to, in particular, health and well-being, industry and agriculture, etc.

This workshop will allow you to:

- Rub your new ideas with other researchers
- Get feedback on your work from the scientific community.
- Initiate or encourage collaborations and partnerships between researchers.
- Set up consortia on related or complementary topics with a view to setting up large-scale national or international projects (NRP, Erasmus, etc.).
- Produce a national map of skills in computer science. This mapping could be used to set up thesis defense juries, appraise and set up projects, etc.

Format

Authors are invited to submit an article of 4 to 8 pages maximum in Springer LNCS format (LaTex and MS Word templates can be downloaded from the https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines). Accepted submissions may lead, depending on their quality, to oral presentations with support or a poster presentation.

Papers must be submitted in PDF format via EasyChair: https://easychair.org/conferences/?conf=csnt2024

Important Dates

- Deadline for submission: 15 September
- Notification to authors: 30 September
- Final texts returned by authors: 6 October
- Workshop in Mostaganem : 8 December 2024

Committee Chairs

Program Committee:

- Pr. SEHABA Karim
- Pr. ATMANI Baghdad

Organizing committee:

- Dr. HENNI Fouad
- Dr. DJEBBARA Rédha

Program Committee

- Djamel Amar Bensaber, ESI of Sidi Bel Abbes
- Baghdad Atmani, University of Oran 1
- Ali Bahloul, University of Batna 2
- Ghanem Belalem, University of Oran 1
- Fatima Zohra Benidris, University of Mostaganem
- Charef Abdallah Bensalloua, University of Mostaganem
- Karim Bessaoud, University of Mostaganem
- Larbi Guezouli, University of Batna 2
- Mohamed Habib Zahmani, University of Mostaganem
- Hafid Haffaf, University of Oran 1
- Nadia Hocine, University of Mostaganem
- Bochra Kaid-Slimane, University of Mostaganem
- Bouabdellah Kechar, University of Oran 1
- Zakaria Laboudi, University of Oum El Bouaghi
- Mohammed Adnane Laredj, University of Mostaganem
- Moulay Driss Mechaoui, University of Mostaganem
- Mohammed Midoun, University of Mostaganem
- Mohammed Redjimi, Universty of Skikda
- Baroudi Rouba, University of Mostaganem
- Asma Saighi, University of Oum El Bouaghi
- Mohamed Sayah, University of Oran 1
- Karim Sehaba, University of Mostaganem
- Mohammed Tadlaoui, University of Tlemcen
- Nacer Eddine Zarour, University of Constantine 2

Recommendation Systems in Collaborative Learning Environment: PRISMA

ZEBLAH Ikhlas[1], SEHABA Karim[1], and HOCINE Nadia[1]

CSTL Lab, Université Abdelhamid Ibn Badis Mostaganem, Algeria.

Abstract. The combination of recommendation systems, learning analytics, and computer-supported collaborative learning (CSCL) is transforming the quickly developing field of e-learning. By using digital tools, CSCL helps students collaborate, share knowledge, and become more actively involved in their education, which improves student results. This study investigates how recommendation systems for peers and resources, can improve collaborative learning even further. Based on a PRISMA-compliant systematic literature review, the research looks at how recommendation systems, learning analytics, and collaboration interact.Two primary research questions are addressed by this study: (RQ1) How can recommendation systems improve collaboration within educational settings? and (RQ2)What relevant data informs these systems? Through an assessment of current approaches and results, this research seeks to close gaps and promote the creation of tailored, adaptive learning environments that successfully meet the varied needs of learners.

Keywords: Recommendation system · computer-supported collaborative learning · personalization · learning analytics

1 Introduction

E-learning and how students learn are rapidly evolving. Nowadays, learners can engage in educational pursuits and collaborate on projects online, removing time and place limits. Computer-Supported Collaborative Learning (CSCL) is an e-Learning subfield that focuses on the use of digital technologies to facilitate collaborative learning. CSCL focuses on how collaborative learning enabled by technology can improve peer interaction and group work, as well as facilitate knowledge and expertise exchange among community members [2]. The pedagogical concept of collaboration among students is based on the idea that individuals learn from each other by working together, which helps generate knowledge and teamwork skills that enable a more informed and engaged workforce [3].

Collaborative learning environments can be more efficient using learning analytics, or LA). Through the analysis of many data sources, LA can offer significant insights into the behaviors of students, the dynamics of groups, and the efficacy of cooperative learning initiatives. These insights can be utilized to determine which students are having difficulty, offer personalized support, improve group dynamics, evaluate the success of cooperative learning activities, and

provide guidance for instructional design. LA methods involve collecting and analyzing data on student profiles, learning styles, interactions, assessments, and behavior. Advanced analytics techniques are employed to extract meaningful insights from this data, which can be used to inform evidence-based decision-making and improve the overall effectiveness of collaborative learning experiences.

Although CSCL provides tools to enhance collaboration and LA methods detect collaboration, students still need more orientation and guidance to collaborate in the right way. From this instance, CSCL environment can have adaptive recommendations in order to customize group learning. Personalization methods, such as adaptive e-learning and recommendation systems, try to achieve these needs. The latter, in general, are based on machine learning methods and algorithms, and progress has been made [4]. Some of the research recommends activities and others recommend peers. The importance of collaboration in the learning environment leads to the problem of improving this latest among students with the supervision of their teachers. This doctoral research aims to develop an adaptive recommendation system within the context of computer-supported collaborative learning (CSCL). The recommendations will focus on enhancing collaboration by suggesting relevant pedagogical resources and suitable peers. To achieve this goal, two research questions will be addressed: RQ1 How can recommendation systems in collaborative learning environments promote collaboration among students? and RQ2 what is the relevant information utilized in the recommendation system in collaborative learning? A systematic literature review, following the PRISMA guidelines, was conducted to explore existing research that combines CSCL, recommendation systems, learning analytics (LA), and adaptation. PRISMA is a widely used reporting guideline for systematic reviews and meta-analyses, ensuring a transparent and reproducible research process [1] [9] [10].

2 Research method

The goal of this systematic literature review is to identify, synthesize, analyze, and categorize the essential components of previous research in recommendation systems in a collaborative learning environment, in light of a particular research question. For this report, we followed the Preferred Reporting Items for Systematic Reviews and Meta-Analysis for Scoping Reviews(PRISMA-ScR) to improve the methodological quality [5].

Although research on collaborative learning, recommendation systems, and learning analytics has gained significant traction, studies that examine the interplay of all three dimensions remain scarce. Existing literature predominantly concentrates on either distance education contexts, as exemplified by [7], or on the combination of only one or two of these concepts, as demonstrated in [8] and [6]. [8] adhere to the PRISMA framework, which highlights the value of systematic reviews in this domain. Our research seeks to address this gap by conducting a comprehensive exploration of the integration of collaboration, rec-

ommendation, and learning analytics, utilizing the PRISMA methodology to ensure rigor and reproducibility.

To that end, we investigate the following general Research Question (RQ): what is the current scientific knowledge about the development of personalized recommender systems to improve collaboration in education? Sub-questions mentioned in the Introduction RQ1 and RQ2 operationalize this global RQ. Then, the definitions of the research terminology and equations were given, and based on these, research began on several scientific databases. Next, articles were analyzed, utilizing inclusion and exclusion criteria.

Segment	Research terms
adaptive recommendation systems	(Recommendation OR scaffold) AND (adaptation OR personalization OR customization)
Learning analytics	("learning analytics" OR "machine learning")
Collaborative learning	("computer supported collaborative learning" OR collaboration)

Table 1. Research terms

2.1 Search queries

To obtain the relevant articles, we defined search queries based on the keywords mentioned in 1, which are divided into two sections: Learning analytics, Recommendation system, and Collaborative learning environment. The first segment includes several alternative terms for learning analytics. The second segment includes the terms related to recommendation systems. The third segment includes terms for collaborative learning and all its synonyms when they are used in the field of education. In each database, the query depends on its syntax, it's mentioned in Table 2

2.2 Pretreatment

IEEE (Institute of Electrical and Electronics Engineers)-Xplore, ScienceDirect, Web of Science, The ACM (Association for Computing Machinery Digital) Library, and SpringerLink were among the electronic databases searched for this literature study,Google scholar also some articles found from other sources like articles found through the bibliography of the selected articles or articles that mention a selected article. The papers were chosen in three processes, as suggested by the PRISMA standards, and depicted [5]. After deleting duplicates, merging all the articles in one Excel file, and adding a column selection (yes/no) this indicates whether the article is selected for in-depth reading/analysis or not. The title was checked for topic relevance in the first step. For this purpose, the title and the abstract have been checked for topic relevance in the first step. We then study the full text of the papers found in the second stage to identify

database	query
science direct	(recommendation OR scaffold) AND (adaptation OR personalization OR customization) AND "learning analytics" AND ("computer supported collaborative learning" OR "collaborative learning")
ACM	Abstract: computer supported collaborative learning] AND [[Abstract: recommendation] OR [Abstract: scaffold]] AND [[Abstract: personalization] OR [Abstract: adaptation] OR [Abstract: customization]] AND [Abstract: learning analytics]
Google Scholar / springerLink	(recommendation or scaffold) and (adapt* or personal* or custo*) and "learning analytics" and ("computer supported collaborative learning" or "collaborative learning")
IEEE	("All Metadata":recommend*) OR ("All Metadata":Scafold) AND ("All Metadata":adapt*) OR ("All Metadata":costum*) OR ("All Metadata":personaliz*) AND ("All Metadata":computer supported collaborative learning) AND ("All Metadata":collab*) AND ("All Metadata":Learning analytics)

Table 2. Research Queries

relevant research that matches the analyzing criteria mentioned in the next section. To include significant papers, we used the search terms shown in Table 1. The selected keyword searches were applied to each article's title, keywords, and abstract after that to the full text. The process of selection is demonstrated in the figure.

The literature search was conducted between February and June 2023, and the publication period was between 2010 and 2023. The goal of this first stage is to examine the title and abstract of each article to find relevant publications that answer the research questions. To that purpose, we employ the inclusion and exclusion criteria listed below:

- Inclusion criteria:
  1. Articles that focus on recommender or personalized or adaptive systems in a collaborative learning environment.
  2. Have been published between 2010 and 2023.
  3. Focus on the detection of collaboration with LA methods.
  4. Focus on improving collaboration.
- Exclusion criteria:
  1. The article discusses aspects not in relation to our research topic.
  2. Articles that do not present empirical research results.
  3. The article must be peer-reviewed; articles that have not been evaluated or judged by peers are excluded.
  4. Short papers, workshops, reviews, chapters, and books

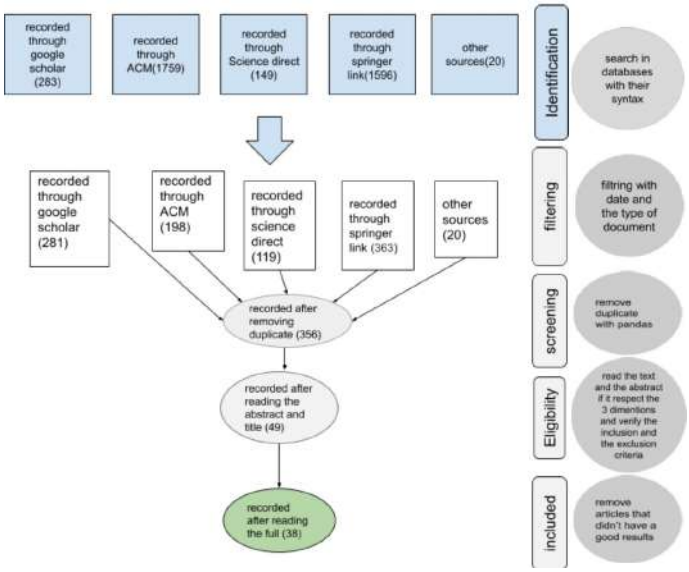

Fig. 1. The process of selection

2.3 Analyzing criteria

Analyzing criteria were searched for, in each article, they are listed below:

- Learning context: Information about the learning context, such as the subject of study, grade level, time constraints, and teaching mode, can influence recommendations by taking into account the specificities of the learning context.
- Approach: the proposed approach: recommendation/adaptation/guidance. . . .
- What? : what do they recommend/adapt ?
- Who? : the target of adaptation/recommendation: teacher or students (individual or group adaptation/recommendation)
- How?: How do they adapt or make recommendations? the technique and the model they focus on.
- Inputs (parameters): What are the inputs or information used to make decisions about adaptation/recommendation/guidance?
- Results: impact of the study. Whether the proposed technique influenced collaboration skills and performance.

3 Conclusion

In conclusion, this work applied the PRISMA methodology to ensure a rigorous and systematic review of literature on recommender and adaptive systems in Computer-Supported Collaborative Learning (CSCL). By following PRISMA's

structured approach, we identified, screened, and selected the most relevant studies, offering a comprehensive understanding of how these systems enhance collaborative learning environments. This method allowed us to filter out irrelevant research, ensuring a robust foundation for future analysis. In upcoming work, we will further analyze these selected studies in depth, identifying key trends and gaps. Additionally, we aim to propose a novel solution that integrates recommender and adaptive systems to optimize personalized learning and collaboration in CSCL settings.

References

1. Bremgartner, V., de Magalhães Netto, J.F.: Improving collaborative learning by personalization in virtual learning environments using agents and competency-based ontology. In: 2012 Frontiers in Education Conference Proceedings, pp. 1–6. IEEE (2012)
2. Lipponen, L.: Exploring foundations for computer-supported collaborative learning. International Society of the Learning Sciences (ISLS) (2002)
3. Zamecnik, A., Kovanović, V., Grossmann, G., Joksimović, S., Jolliffe, G., Gibson, D., Pardo, A.: Team interactions with learning analytics dashboards. Computers & Education **185**, 104514 (2022). Elsevier
4. Khanal, S.S., Prasad, P.W.C., Alsadoon, A., Maag, A.: A systematic review: machine learning based recommendation systems for e-learning. Education and Information Technologies **25**, 2635–2664 (2020). Springer
5. Liberati, A., Altman, D.G., Tetzlaff, J., Mulrow, C., Gøtzsche, P.C., Ioannidis, J.P.A., Clarke, M., Devereaux, P.J., Kleijnen, J., Moher, D.: The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. Annals of Internal Medicine **151**(4), W–65 (2009). American College of Physicians
6. Mena-Guacas, A.F., Urueña Rodríguez, J.A
7. da Silva, L.M., Dias, L.P.S., Barbosa, J.L.V., Rigo, S.J., dos Anjos, J., Geyer, C.F.R., Leithardt, V.R.Q.: Learning analytics and collaborative groups of learners in distance education: a systematic mapping study. Informatics in Education **21**(1), 113–146 (2022). Vilnius University Institute of Data Science and Digital Technologies
8. Bandono, A., Mukhlis, M., Susilo, A.K., Prabowo, A.R., Maksum, A.: Collaborative Learning in Higher Education in the Fourth Industrial Revolution: A Systematic Literature Review and Future Research. International Journal of Learning, Teaching and Educational Research **22**(10), 209–230 (2023)
9. Troussas, C., Giannakas, F., Sgouropoulou, C., Voyiatzis, I.: Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM. Interactive Learning Environments **31**(1), 54–67 (2023). Taylor & Francis
10. Rasheed, R.A., Kamsin, A., Abdullah, N.A.: Challenges in the online component of blended learning: A systematic review. Computers & Education **144**, 103701 (2020). Elsevier

Early Prediction Sepsis Using Parallel Deep Learning: Pre- and Post-Processing of Data from the 2019 PhysioNet/Computing in Cardiology Challenge Dataset

Hadj Ali Elmerahi[1], Baghdad Atmani[1,2] , Fatiha Barigou[1] , Belarbi Khemliche[3], Badreddine Errouane[3] and Mohammed bousmaha[3]

[1] Laboratoire d'Informatique d'Oran (LIO)
University of Oran 1 Ahmed Benbella
Oran, Algeria
[2] Computer Science and New Technologies Lab
University of Mostaganem Abdelhamid Ibn Badis
Mostaganem, Algeria
[3]Faculty of medicine
University of Oran 1 Ahmed Benbella
Oran, Algeria
el.merahi@gmail.com

Abstract. Sepsis is a major public health problem and a leading cause of death worldwide. Faced to these challenges, we propose a new early prediction model called Parallel Neural Networks Fusion Predictor of Sepsis (PNNFPS) for prediction sepsis within ICU patients. PNNFPS encompasses four deep learning (DL) Models: a Long Short-Term Memory (LSTM), two Deep Neural Networks (DNN1, DNN2) and an Artificial Neural Network (ANN), enabling the parallel processing of various clinical data types. By combining the strengths of the sequential hybrid LSTM-DNN1 and DNN2 model with feature fusion, PNNFPS produce sepsis probability score. Additionally, *PNNFPS* involves employing intelligent data analysis, which includes both pre-processing step, such as data imputation and segmentation and post-processing techniques like hyperparameter tuning and threshold optimization to enhance performance. In this paper, we detail PNNFPS architecture, the pre-processing and post-processing data steps using the 2019 PhysioNet / Computing in Cardiology Challenge dataset.

Keywords: Early prediction of sepsis, Machine learning, LSTM, DNN, Deep learning, Fusion.

1 Introduction

Sepsis is the major public health problem and a leading cause of death worldwide [1], responsible for nearly *35%* of all hospital-related deaths in the *United States* [2]. Early sepsis prediction system can significantly enhance patient outcomes by enabling timely interventions. While traditional methods rely on clinical judgment (*scoring*

systems), advancement in machine learning (ML) and deep learning (DL) have opened new possibilities, such as hybrid models based on fusion technologies for more accurate and timely prediction systems [3]. Feature fusion technologies, such as concatenation, element-wise addition, multiplication are detailed in [4].

In this paper, we propose a new early prediction model named *Parallel Neural Networks Fusion predictor of Sepsis (PNNFPS)* for prediction sepsis within *ICU* patients. *PNNFPS is a* parallel *DL* architecture that integrates *four DL* models: a *LSTM, two DNNs (DNN1, DNN2)* and *an ANN.* The main idea of this study is that *PNNFPS* operates the *sequential hybrid LSTM-DNN1* and *DNN2* models in *parallel*, with a *feature fusion technology* applied to their outputs. Finally, an *ANN* generates a sepsis probability score with a *threshold* used to classify *septic* and *non-septic* cases.

The expected impact of this study is to enhance early sepsis prediction in ICU settings, leading to timely interventions and reduced mortality rates. The hypotheses include that the PNNFPS model will outperform traditional approaches in predictive performance. The PNNFPS includes both pre-processing step, such as data imputation and segmentation to ensure data quality and post-processing techniques like hyperparameter tuning and threshold optimization to enhance performance.

In this study, we present the PNNFPS architecture, the pre-processing and pos-processing data steps, employing the 2019 PhysioNet/CinC Challenge [5]. The remainder of this paper is organized as follows: Section 2 reviews related works, Section 3 describes the proposed approach, Section 4 presents our experiments and Section 5 concludes the study.

2 Literature review

Recent advancements in sepsis prediction models demonstrate an evolving focus on leveraging DL architectures to improve early sepsis prediction and intervention.

Shashikumar, et al. [6] developed a DNN model for predicting sepsis 4 to 48 hours before clinical suspicion (sepsis-3) [2], designed to reduce false alarms. While their single-model approach demonstrated consistent performance across various care settings and validation cohorts, it struggled with temporal dependencies in sequential data.

Zhang et al. [7] proposed an *LSTM* model for predicting sepsis *4 hours* prior its onset *(sepsis-2)* [8] in the Emergency Department (*ED).* This model effectively manages temporal information; however, issues related to model interpretability and computational requirements remain significant limitations.

Lee et al. [9] employed the PhysioNet/CinC 2019 dataset [10] to develop a DL-based early warning system for sepsis prediction at various intervals of 4,6,8,12 hours prior clinical onset. The model demonstrated practical potential for real hospital environments, limitations remain, including the need for clinical validation and the challenge of managing ICU patients' data noise.

.

Our previous work [11] introduced a parallel LSTM- DNN model for early prediction sepsis using ICU dataset [5]. This approach demonstrated reasonable performance and effectively generalized to test data, despite relying solely on learning rate and neuron count optimization per layer. However, the model would benefit from further hyperparameter tuning and threshold optimization to improve predictive accuracy.

To further enhance predictive accuracy, our current approach PNNFPS addresses these identified limitations. We anticipate that PNNFPS will outperform previous approaches by leveraging increased model depth, refined data split strategies, comprehensive hyperparameter and threshold optimization. These improvements are expected to enhance the model's ability to capture temporal dependencies, reduce computational overhead, and achieve greater predictive accuracy in early sepsis prediction.

3 Proposed approach

In this paper, we propose PNNFPS, a parallel DL architecture based on a fusion technology, for predicting sepsis six hours prior to its clinical diagnosis (sepsis-3), within ICU patients. Initially, PNNFPS involves employing both pre-processing step, such as data imputation and segmentation and post-processing techniques like hyperparameter tuning and threshold optimization to enhance model performance. The PNNFPS model leverages a 16-hour Look Back Window (LBW) without hand-engineered feature extraction, utilizing four DL models: LSTM, two DNNs, and an ANN. The LSTM and DNN1 operate sequentially, forming a hybrid model, while the hybrid model and DNN2 run in parallel. An element-wise additive layer fuses the outputs, and the ANN predicts a sepsis probability score, followed by a threshold for classification. Figure 1 outlines the model's key phases, which are detailed in this section.

3.1 Dataset

The dataset used in this study is from the 2019 PhysioNet/CinC Challenge [5]. It includes clinical data from 40,336 ICU patients across two hospitals: 20,336 from Hospital A and 20,000 from Hospital B. Each patient's data, stored in Pipeline Separated Value (.PSV) format, consists of hourly time series for 40 clinical variables, including 8 vital signs, 26 laboratory tests, and 6 demographic factors. Detailed descriptions on these clinical variables are available in [5].

3.2 Data pre-processing

Before constructing the model, data preprocessing is essential to ensure quality. This involves two key steps: data imputation and segmentation.

Data imputation. The imputation process addresses three subsets of features: binary (2), frequent (8), and infrequent (26). The selection of these subsets is grounded in their relevance to the clinical context and their potential impact on the model's predictive performance. Binary features (Gender, Unit1) are handled using backward-fill followed by forward-fill, with any remaining missing values filled as 1, indicating a male patient from a medical ICU [12]. For frequent (Age, 7 vital signs) and infrequent features (26 laboratory tests) are first filled with the global mean, followed by backward-fill and forward-fill for any remaining gaps.

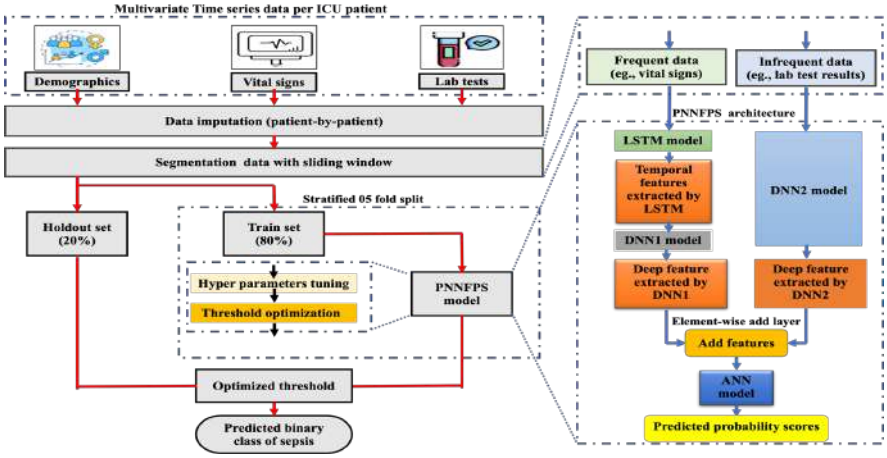

Fig. 1. General architecture of the proposed approach.

Data segmentation. Data segmentation is a crucial step in the preprocessing pipeline of PNNFPS, enabling the capture of temporal dependencies in sepsis prediction. Using an overlapping sliding window of 16 hours, advanced by one-hour increments, the dataset is divided into two sub-samples: FS1 (8 features, including vital signs and Age as a 16-hour multivariate time series) and FS2 (28 features, including lab tests, Gender, and Unit1 as median values). Sub-sample1 is represented as a 3D tensor with a size of $(N,LBW_{length},FS1_{length})$, while sub-sample2 is a 1D tensor with a size of $(N,FS2_{length})$. This differs from previous approaches using a 12-hour window over 40 features [13]. It is important to note that this segmentation process is identical to the one used in our previous method [11].

3.3 PNNFPS architecture.

As shown in figure 2*, PNNFPS* consists of *four* components: the *sequential hybrid LSTM-DNN1* model, the *DNN2* model, an e*lement-wise additive layer for fusion* and the *ANN* model.

Standard hybrid LSTM-DNN1 model. The standard hybrid LSTM-DNN1 model consists of two modules: an LSTM module for temporal feature extraction and a DNN1 module for further processing. The LSTM module uses the 3D-tensor as input to a bidirectional LSTM, concatenating outputs from both directions and passing them

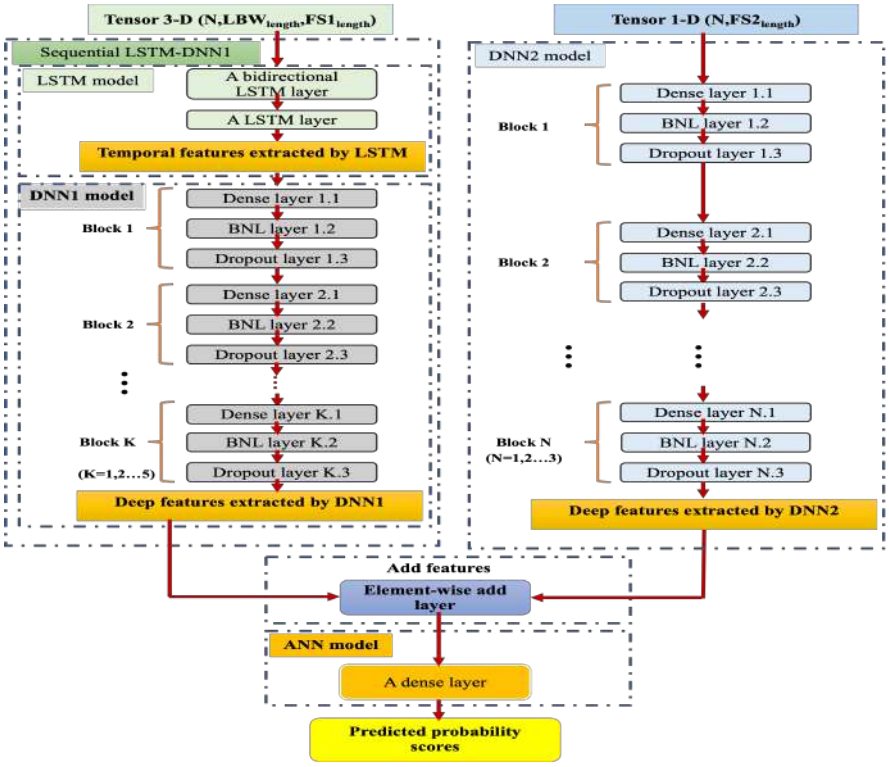

Fig. 2. PNNFPS architecture. BNL= *Batch Normalization Layer*

through an additional LSTM layer to extract temporal features. These features are then fed into the DNN1 module, which processes them through *K* blocks, each consisting of a dense layer, batch normalization, and dropout, to extract deep features and accelerate training.

DNN2 model. The 1D-tensor data is passed through *N* blocks of layers, each involving a dense layer followed by a BNL and a dropout layer. This process produces a set of deep features with same length as the deep features extracted from DNN1.

Element-wise additive layer for fusion. The outputs consisting the two sets of extracted deep features from the final dropout layers of both preceding models (the standard hybrid LSTM-DNN1 model and the DNN2 model) are fused (added), using an element-wise additive layer.

ANN model. The outputs of the element-wise additive layer are passed through a dense layer using the sigmoid activation function with 1 unit. During the prediction phase, this layer maps the probability scores for both classes. Finally, a threshold

value is applied to these probability scores to discriminate between septic and non-septic classes. Notably, with the exception of the dense layers of the ANN, the Rectified Linear Unit (ReLU) activation function is aplied to all other layers.

3.4 Data post-processing

During developing a model, data pos-processing is important to enhance its performance. This phase involves addressing two key aspects: hyperparameter tuning, and threshold optimization.

Hyperparameter tuning. The hyperparameters of the proposed PNNFPS will be tuned using a random search via Keras Tuner's "hypermodel" criterion [14], with a 5-fold stratified strategy applied to 80% of the dataset. This process results five preliminary models, each with unique hyperparameters, including the number of layers, neurons per layer, and learning rates, optimized based on the AUROC, the Binary Cross Entropy (BCE) loss function and the Adam optimizer [15].

Threshold optimization. The tuned hyperparameters from each fold will be used to retrain the final PNNFPS model, followed by threshold optimization to maximize the AUROC on the same fold. This process systematically adjusts threshold values applied to the prediction probabilities, selecting those that yielded the highest U score. The model corresponding to the fold with the highest U score, along with optimized thresholds, will then applied to the holdout set (20% of the dataset) for generating overall predictions, which will subsequently evaluated using the U score and AUROC metrics.

4 Experiment

4.1 PNNFPS development.

The proposed PNNFPS including pre-processing steps was developed in Python using Keras [16] with TensorFlow [17] for early sepsis prediction from hourly clinical data. The dataset of 40,336 patients was randomly split into 80% for training, using a 5-fold stratified strategy, and 20% as a holdout set for testing. The model will be trained for 100 epochs, using a single-patient per batch aspect. Folds were selected to contain approximately the same number of septic patients, ensuring that no patient' data appeared in more than one fold. This batching process preserves the temporal sequence of the patient's physiological data, enabling the model to accurately capture and predict the progression towards sepsis by analyzing individual patient trajectories.

4.2 PNNFPS architecture comparison.

The proposed approach differs from our previous method [11] in several aspects: the depth and nature of the architecture, the data split strategy, data imputation, threshold

optimization and training batch size. Notably, PNNFPS employs K blocks of layers in DNN1 and N blocks in DNN2, whereas our previous method [11] used only one block (K=N=1) in its architecture. This approach is also differs from the approaches that rely on sequential [12] or individual models [6] or ensemble model [13].

5 Conclusion.

In this paper, we present the PNNFPS, a parallel DL architecture based on a fusion technology, for predicting sepsis onset six hours before its clinical occurrence, utilizing a 16-hour window of historical data. In particular, this study discussed two main steps: pre-processing step, such as data imputation and segmentation to ensure data quality and post-processing techniques like hyperparameter tuning and threshold optimization to enhance performance. Unlike previous approaches that rely on sequential or individual models, our method processes data in parallel. Future work will focus on validating the PNNFPS model across diverse clinical environments, integrating additional clinical features, and simulating real-time predictions. Additionally, extensive hyperparameter tuning and performance assessments under varying data conditions will be conducted to enhance the model's adaptability and robustness in real-world settings.

Acknowledgment. This work is part of the "Medical Decision-Making Support Based on Artificial Intelligence and Reasoning for Intensive Care (AIR4AIC)project, conducted by the Artificial Intelligence and Reasoning (AIR) team in collaboration with the reanimation service of the University Hospital Establishment (UHE) of Oran, Algeria.

References

[1] N. Ocampo-Quintero, P. Vidal-Cortés, L. del Río Carbajo, F. Fdez-Riverola, M. Reboiro-Jato, and D. Glez-Peña, “Enhancing sepsis management through machine learning techniques: A review,” *Medicina Intensiva*, vol. 46, no. 3, pp. 140–156, Mar. 2022.

[2] M. Singer, C.S. Deutschman, C.W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G.R. Bernard, J.D. Chiche, C.M. Coopersmith, R. S. Hotchkiss, M.M. Levy, J.C. Marshall, G.S. Martin, S.M. Opal, G.D. Rubenfeld, T. van der Poll, J.L. Vincent, D.C. Angus, The third international consensus definitions for sepsis and septic shock (Sepsis-3), J. Am. Med. Assoc. 315 (2016) 801.

[3] Y. Duan, J. Huo, M. Chen, F. Hou, G. Yan, S. Li, and H. Wang. Early prediction of sepsis using double fusion of deep features and handcrafted features. Applied Intelligence. https://doi.org/res.

[4] N. Mungoli, “Adaptive Feature Fusion: Enhancing Generalization in Deep Learning Models,” 2023, *arXiv*.

[5] M.A. Reyna, C.S. Josef, R. Jeter, S.P. Shashikumar, M.B. Westover, S. Nemati, G. D. Clifford, A. Sharma, in: Early Prediction of Sepsis from Clinical Data:

the PhysioNet/Computing in Cardiology Challenge 2019, Critical Care Medicine vol. 48, Publisher: Lippincott Williams & Wilkins, 2020, pp. 210–217.

[6] S. P. Shashikumar, G. Wardi, A. Malhotra, and S. Nemati, "Artificial intelligence sepsis prediction algorithm learns to say 'I don't know,'" *npj Digit. Med.*, vol. 4, no. 1, p. 134, Sep. 2021.

[7] D. Zhang, C. Yin, K. M. Hunold, X. Jiang, J. M. Caterino, and P. Zhang, "An interpretable deep-learning model for early prediction of sepsis in the emergency department," *Patterns*, vol. 2, no. 2, p. 100196, Feb. 2021.

[8] M. M. Levy *et al.*, "2001 SCCM/ESICM/ACCP/ATS/SIS International Sepsis Definitions Conference:," *Critical Care Medicine*, vol. 31, no. 4, pp. 1250–1256, Apr. 2003, doi: 10.1097/01.CCM.0000050454.01978.3B.

[9] B. T. Lee, O.-Y. Kwon, H. Park, K.-J. Cho, J.-M. Kwon, and Y. Lee, "Graph Convolutional Networks-Based Noisy Data Imputation in Electronic Health Record," *Critical Care Medicine*, vol. 48, no. 11, pp. e1106–e1111, Nov. 2020.

[10] Reyna, Matthew *et al.*, "Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019." PhysioNet.

[11] H.A. Elmerahi, B.Atmani, F. Barigou, B. Khemliche, B. Errouane, M. Bousmaha, ' Parrallel LSTM-DNN Fusion model for Early Prediction of Sepsis in Intensive Care Units,' IEEE 4th International Conference on Embedded and Distributed Systems, Accepted for publication, Bechar, Algeria, 2024.

[12] A. Rafiei, A. Rezaee, F. Hajati, S. Gheisari, and M. Golzan, "SSP: Early prediction of sepsis using fully connected LSTM-CNN model," *Computers in Biology and Medicine*, vol. 128, p. 104110, Jan. 2021.

[13] X. Li, G. André Ng, and F. Schlindwein, "Convolutional and Recurrent Neural Networks for Early Detection of Sepsis Using Hourly Physiological Data from Patients in Intensive Care Unit," presented at the 2019 Computing in Cardiology Conference, Dec. 2019.

[14] Malley, Tom and Bursztein, Elie and Long, James and Chollet, Fran\c{c}ois and Jin, Haifeng and Invernizzi, Luca and others, "KerasTuner." Accessed: Oct. 02, 2023. [Online]. Available: https://github.com/keras-team/keras-tuner

[15] D.P. Kingma, J. Ba, Adam, A method for stochastic optimization, URL, 2017. http:// arxiv.org/pdf/1412.6980v9.

[16] Chollet and Francois, "Keras." [Online]. Available: https://github.com/fchollet/keras

[17] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, 'TensorFlow: A system for large-scale machine learning,' 2016.

Towards identifying multicriteria outliers: An approach based on PROMETHEE γ and DBSCAN algorithm

Toufik Achir[1] and Baroudi Rouba[1]

[1] Lab CSTL, Mostaganem University, 27000 Mostaganem, Algeria.

toufik.achir.etu@univ-mosta.dz
baroudi.rouba@univ-mosta.dz

Abstract. Outlier detection within the MCDA field has not been sufficiently explored in the existing literature. To address this gap, we propose a novel approach that leverages the preference indicators γ_{ij} and γ_{ji} derived from PROMETHEE γ to define new metrics (γ_i^+, γ_i^-) for each alternative. These metrics are subsequently used as inputs to the DBSCAN algorithm for outlier identification. We demonstrate the effectiveness of the proposed approach by applying it to the real-world case of the Human Development Index (HDI).

Keywords: Outlier detection; Multicriteria decision aid; promethee γ; DBSCAN.

1 Introduction

Multicriteria decision aid (MCDA) is a branch of Operational Research (OR) focused on creating methods to help decision-makers handle situations involving multiple, often conflicting criteria simultaneously [1]. Numerous MCDA methods have been suggested in the literature. Majumder [2] categorizes these methods into two primary types: Compensatory methods [3][4] and Outranking methods [5][6].
MCDA methods have been utilized across various fields, including agriculture [7], education [8], and finance [9], among others. In certain cases within the MCDA field, one alternative (or a group of alternatives) may significantly stand out from the rest, potentially signaling the presence of a multicriteria outlier.

As defined by Barnett and Lewis [10], an outlier is "an observation (or subset of observations) which appears to be inconsistent with the remaining data." These outliers often represent valuable and essential data in a database. Outlier (or anomaly) detection has been applied in the financial industry, fraud detection [11], wireless sensor networks [12], network intrusion [13], and various other fields. Extensive discussions on outlier detection techniques are available in the literature, where these methods are categorized into statistical, distance, density, depth, and clustering-based approaches.

Identifying outliers within the MCDA field represents a relatively new research avenue that has not been extensively explored in the literature. To our knowledge, only three studies have addressed this issue. The initial method was introduced by De Smet

& al [14]. The authors developed a model based on the distance measure introduced by De Smet and Montano [15], extending it to multiple samples of the set of alternatives. The approach relies on comparisons drawn from these samples, with bi-modal distributions serving as indicators of potential outliers. In a second paper**,** Rouba and Nait Bahloul [16], proposed generating preference relations via a multicriteria outranking method, where each alternative is identified by its relationships with others. To detect outliers, they applied the local outlier factor (LOF) algorithm [17] to the distributions of the outranking relations. In the third paper, the authors utilized statistical techniques to identify outliers [18]. The PROMETHEE method is employed to generate a net-flow value for each alternative, and the technique for identifying outliers is chosen according to the distribution of these values. If the net-flow values follow a normal distribution, the standard deviation method is used; in other cases, the interquartile range method is applied.

This paper introduces a novel method for detecting outliers in the MCDA field. The proposed approach is based on the mono-criterion net flow scores indicators (γ_{ij} , γ_{ji}) derived from PROMETHEE γ [19]. These indicators not only reflect the pairwise comparisons between specific alternatives but also consider how each alternative interacts with all other alternatives in the problem. For this reason, these indicators are strong candidates for use in the outlier detection process. In the proposed approach (γ_{ij} , γ_{ji}) indicators are used to establish new metrics (γ_i^+, γ_i) for each alternative. These metrics are subsequently **used** as input into the DBSCAN algorithm [20] to detect outliers.

The remainder of this paper is structured as follows: Section 2 introduces the classical PROMETHEE method and its variant, PROMETHEE γ. In Section 3, the proposed approach is detailed, accompanied by an application example. Finally, the paper concludes in Section 4.

2 PROEMTHEE Methods

In the MCDA field, methods are designed to help decision makers (DM) in the comparison of a set of alternatives, denoted as $A = \{a_1, a_2, ..., a_n\}$, which are evaluated on a set $F = \{f_1, f_2, ..., f_k\}$ of k conflicting criteria. The PROMETHEE methods were designed to help decision-makers rank alternatives evaluated across multiple criteria, offering either partial rankings (PROMETHEE I) or full rankings (PROMETHEE II).

1. Initially, we compute the difference between the evaluations of two alternatives (a_i, a_j) from the set $A \times A$ on each criterion c within the range 1 to k:

$$d_c(a_i, a_j) = f_c(a_i) - f_c(a_j) \quad \forall\, c \in 1, \dots, k \tag{1}$$

Where $f_c(a_i)$ represents the evaluation of alternative a_i on criterion c. To express these differences in terms of preference, we utilize the preference function P. $P_c[d_c(a_i, a_j)]$ denotes the degree of preference of a_i over a_j for the criterion c, with P_c being a positive, non-decreasing function that takes values

between 0 and 1. PROMETHEE method introduces six variants of preference functions, giving decision makers the ability to customize their preference modeling (For further details, refer to [21].).

2. The second step involves assigning a multicriteria preference index to each pair (a_i, a_j) in A×A:

$$\pi(a_i, a_j) = \sum_{c=1}^{k} w_c \, P_c(a_i, a_j), \quad (\sum_{c=1}^{k} w_c = 1) \tag{2}$$

where w_c (with c ranging from 1 to k) represent the standardized weights assigned to the criteria. This index is a positive real number ranging from 0 to 1, serving as an overall measure of how one alternative is preferred over another.

3. In the last step, we calculate the outranking flow scores—both positive and negative—for each alternative $a_i \in A$. The positive flow score shows the extent to which alternative a_i is preferred over other alternatives.

$$\phi^+(a_i) = \sum_{a_j \in A} \pi(a_i, a_j) \tag{3}$$

The negative flow score shows the extent to which other alternatives are preferred over alternative a_i.

$$\phi^-(a_i) = \sum_{a_j \in A} \pi(a_j, a_i) \tag{4}$$

The PROMETHEE II methods defines a unique score for each alternative defined as the difference between the positive and negative flow scores:

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i) \tag{5}$$

2.1 PROMETHEE γ

PROMETHEE γ is an extension of the PROMETHEE methods designed to address the rank reversal issue. This phenomenon occurs when the addition or removal of an alternative in the dataset leads to a reversal in the ranking order of two other alternatives.

In this section, we won't delve deeply into PROMETHEE γ; instead, we'll focus solely on the aspects pertinent to our approach, specifically the new preference indices proposed in this method. These indices, represented by γ_{ij} and γ_{ji}, are used when comparing two alternatives, a_i and a_j. γ_{ij} denotes the overall advantage of a_i over a_j within the whole dataset. It is the significance of the weight coalition that determines why a_i is preferable to a_j, with the weights being adjusted according to the difference in mono-criterion net flow scores.

$$\gamma_{ij} = \sum_{f^c(a_i) > f^c(a_j)} W_c \, . \left(\phi^c(a_i) - \phi^c(a_j)\right). \tag{6}$$

Where $\phi^c(a_i) = \frac{1}{n-1} \sum_{j=1}^{n} (\pi_{ij}^c - \pi_{ji}^c)$ represents the mono-criterion net flow of a_i on criterion c, $f^c(a_i)$ is the evaluation of a_i on criterion c, and W_c is the weight of criterion c.

This indicator γ_{ij} (respectively γ_{ji}) depend not only on the pairwise comparisons between the specific alternatives but also on how each alternative interacts with all other alternatives in the problem. Therefore, having high values for both γ_{ij} and γ_{ji} relative to other alternatives in the problem indicates strong conflicting information, suggesting that the alternative may be considered as an outlier.

In the following section we will explain how to integrate γ_{ij} (respectively γ_{ji}) indicators into the outlier detection process.

3 Proposed approach

The idea behind the proposed approach is to assess how an alternative a_i interacts with the entire dataset. Alternatives characterized by significant conflicting information are considered outliers. To achieve this, the proposed approach is divided into two main steps:

Step1: to measure how an alternative a_i interacts with the entire dataset, we propose computing two new metrics for each alternative: one representing the global advantage of a_i over all other alternatives, and the other representing the global advantage of all other alternatives over a_i.

Using the previously mentioned indices (γ_{ij} , γ_{ji})), our approach introduces two new indices for each alternative: $\gamma^{+}(a_i)$ and $\gamma^{-}(a_i)$. The index $\gamma^{+}(a_i)$ represents the average global advantage of a_i over all other alternatives in the entire dataset.

$$\gamma^{+}(a_i) = \frac{1}{n-1}\sum_{j=1}^{n}\gamma_{ij} \tag{7}$$

Similarly, $\gamma^{-}(a_i)$ represents the average global advantage of all other alternatives over a_i

$$\gamma^{-}(a_i) = \frac{1}{n-1}\sum_{j=1}^{n}\gamma_{ji} \tag{8}$$

Step2: Apply DBSCAN [20] clustering algorithm using $\gamma^{+}(.)$ and $\gamma^{-}(.)$ as input. Alternatives characterized by significant conflicting information are marked as noise.

To demonstrate our approach, we present a real-world case study involving the Human Development Index (HDI) problem. This study assesses 179 United Nations countries based on three criteria: life expectancy, education, and income index. For additional details on the parameters, refer to De Smet et al. [22].

Our approach involves selecting the top ten alternatives and the lowest-ranked alternative using the PROMETHEE method. This strategy ensures that the lowest-ranked alternative is distinct from the top ten, thereby identifying it as a potential outlier.

In the first step, we used PROMETHEE γ method to derive γ_{ij} and γ_{iji} for all pairs of alternatives, as shown in Table 1. Subsequently, $\gamma^{+}(.)$ and $\gamma^{-}(.)$ values for each alternative are calculated, as demonstrated in Table 2.

Table 1. γ_{ij} and γ_{iji} values for all alternatives

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{179}
a_1	0	0.165	0.561	0.924	1.089	1.122	1.485	1.485	0.99	1.188	3.036
a_2	0	0	0.396	0.759	1.089	0.957	1.32	1.485	0.825	1.023	2.871
a_3	0.099	0.099	0	0.528	1.188	0.825	0.924	1.584	0.528	0.726	2.574
a_4	0.231	0.231	0.297	0	0.792	0.429	0.693	1.188	0.825	0.495	2.343
a_5	0	0.165	0.561	0.396	0	0.396	0.957	0.396	0.561	0.66	1.947
a_6	0	0	0.165	0	0.363	0	0.561	0.759	0.396	0.264	1.914
a_7	0.099	0.099	0	0	0.66	0.297	0	0.66	0.528	0.198	1.65
a_8	0	0.165	0.561	0.396	0	0.396	0.561	0	0.561	0.66	1.551
a_9	0	0	0	0.528	0.66	0.528	0.924	1.056	0	0.627	2.046
a_{10}	0	0	0	0	0.561	0.198	0.396	0.957	0.429	0	1.848
a_{179}	0	0	0	0	0	0	0	0	0	0	0

Table 2. , $\gamma^+(.)$ and $\gamma^-(.)$ corresponding to each alternative.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{179}
$\gamma^+(.)$	1.095	0.975	0.825	0.684	0.549	0.402	0.381	0.441	0.579	0.399	**0**
$\gamma^-(.)$	0.039	0.084	0.231	0.321	0.582	0.468	0.711	0.87	0.513	0.531	**1.98**

.

In the second step, we used BDSCAN clustering algorithm with parameters with ε =0.5 and MinPts = 5. Alternative a_{179} has been detected as outliers as depicted in Fig 1.

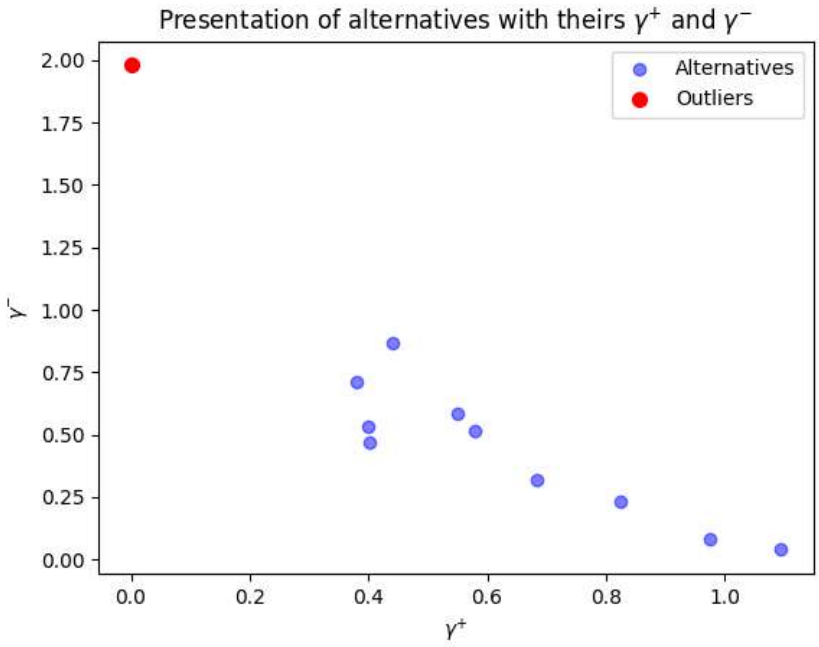

Fig. 1. Representation of Alternatives in Space Using their γ^+ and γ^- Measures.

4 Conclusion

Outlier detection within the MCDA field has not been adequately covered in existing literature. This paper introduces a new method for detecting outliers in MCDA. The proposed approach is based on detecting alternatives characterized by significant conflicting information. To measure this conflicting information, we proposed to compute, for each alternative, two metrics based on PROMETHEE γ. These metrics have been used as input in DBSCAN algorithm to detect outliers. To prove its applicability, the proposed method has been effectively tested on real-world data. This paper serves as an initial exploration and calls for further expansion. Future work will involve applying the method to larger datasets and comparing its performance with existing approaches to validate its effectiveness.

References

[1] C. Zopounidis and M. Doumpos, "Multi-criteria decision aid in financial decision making: Methodologies and literature review," *J. Multi-Criteria Decis. Anal.*, vol. 11, no. 4–5, pp. 167–186, 2002, doi: 10.1002/mcda.333.

[2] M. Majumder, "Multi Criteria Decision Making," pp. 35–47, 2015, doi: 10.1007/978-981-4560-73-3_2.

[3] D. V. O. N. Winterfeldt and G. W. Fischer, "Multi-attribute utility theory: models and assessment procedures**," no. Xi, pp. 47–85, 1975.

[4] T. J. Murray, L. L. Pipino, and J. P. Van Gigch, "A pilot study of fuzzy set modification of delphi," *Hum. Syst. Manag.*, vol. 5, no. 1, pp. 76–80, 1985, doi: 10.3233/HSM-1985-5111.

[5] J. P. Brans and P. Vincke, "Note—A Preference Ranking Organisation Method," *Manage. Sci.*, vol. 31, no. 6, pp. 647–656, 1985, doi: 10.1287/mnsc.31.6.647.

[6] V. Mousseau, B. Roy, and U. Paris-dauphine, "Chapter 4 Introduction : A Brief History," *Recherche*, vol. 78, no. 4, pp. 1–35, 2005, [Online]. Available: http://www.lamsade.dauphine.fr/dea103/ens/bouyssou/Outranking_Mousseau.pdf

[7] M. Yazdani, E. D. R. S. Gonzalez, and P. Chatterjee, "A multi-criteria decision-making framework for agriculture supply chain risk management under a circular economy context," *Manag. Decis.*, vol. 59, no. 8, pp. 1801–1826, 2019, doi: 10.1108/MD-10-2018-1088.

[8] R. A. Carrasco, P. Villar, M. J. Hornos, and E. Herrera-Viedma, "A Linguistic Multi-Criteria Decision Making Model Applied to the Integration of Education Questionnaires," *Int. J. Comput. Intell. Syst.*, vol. 4, no. 5, pp. 946–959, 2011, doi: 10.1080/18756891.2011.9727844.

[9] Y. J. Wang, "Applying FMCDM to evaluate financial performance of domestic airlines in Taiwan," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1837–1845, 2008, doi: 10.1016/j.eswa.2007.02.029.

[10] V. Barnett and T. Lewis, *Outliers in statistical data*, vol. 3, no. 1. Wiley New York, 1994.

[11] R. J. Bolton and D. J. Hand, "Unsupervised profiling methods for fraud

detection," *Credit scoring Credit Control VII*, pp. 235–255, 2001.

[12] Y.-L. Tsou, H.-M. Chu, C. Li, and S.-W. Yang, "Robust distributed anomaly detection using optimal weighted one-class random forests," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 1272–1277.

[13] Q. Ding, N. Katenka, P. Barford, E. Kolaczyk, and M. Crovella, "Intrusion as (anti) social communication: characterization and detection," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 886–894.

[14] Y. De Smet, J.-P. P. Hubinont, and J. Rosenfeld, "A note on the detection of outliers in a binary outranking relation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10173 LNCS, pp. 151–159, 2017, doi: 10.1007/978-3-319-54157-0_11.

[15] Y. De Smet and L. M. Guzmán, "Towards multicriteria clustering: An extension of the k-means algorithm," *Eur. J. Oper. Res.*, vol. 158, no. 2, pp. 390–398, 2004, doi: 10.1016/j.ejor.2003.06.012.

[16] B. Rouba and S. Nait-Bahloul, "Towards identifying multicriteria outliers: An outranking relation-based approach," *Int. J. Decis. Support Syst. Technol.*, vol. 10, no. 3, pp. 27–38, 2018, doi: 10.4018/IJDSST.2018070102.

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[18] B. Rouba, "A net-flow based approach to detect outliers in multicriteria decision problems," *Intell. Decis. Technol.*, vol. 15, no. 2, pp. 239–250, 2021, doi: 10.3233/IDT-200046.

[19] G. Dejaegere and Y. De Smet, "Promethee γ: A new Promethee based method for partial ranking based on valued coalitions of monocriterion net flow scores," *J. Multi-Criteria Decis. Anal.*, vol. 30, no. 3–4, pp. 147–160, 2023.

[20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, 1996, vol. 96, no. 34, pp. 226–231.

[21] J.-P. Brans, P. Vincke, and B. Mareschal, "How to select and how to rank projects: The PROMETHEE method," *Eur. J. Oper. Res.*, vol. 24, no. 2, pp. 228–238, 1986.

[22] Y. De Smet, P. Nemery, and R. Selvaraj, "An exact algorithm for the multicriteria ordered clustering problem," *Omega*, vol. 40, no. 6, pp. 861–869, 2012.

Investigating Gaps in Blockchain Scalability and Conflict Resolution: The Potential of CRDTs and AI in Decentralized Environments

Houcine Ourabah[1], Moulay Driss Mechaoui[1], and Abdessamad Imine[2]

[1] CSTL Lab and Abdelhamid Ibn Badis University, Mostaganem, Algeria
houcin.ourabah@gmail.com
[2] driss.mechaoui@univ-mosta.dz
[3] Lorraine University and LORIA-CNRS-INRIA Nancy Grand Est, Nancy, France
abdessamad.imine@loria.fr

Abstract. Decentralized systems, particularly blockchain, face persistent challenges regarding scalability, concurrency, and data consistency. CRDTs (Conflict-Free Replicated Data Types) offer innovative solutions for managing distributed data conflicts without centralized control. Simultaneously, Large Language Models (LLMs) like GPT-3 are transforming software development, especially in automating the complex process of code merge conflict resolution. Moreover, AI is revolutionizing conflict resolution strategies in distributed networks. This paper investigates the integration of CRDTs in blockchain technology, explores the potential of LLMs in distributed conflict management, and highlights AI's evolving role in enhancing coordination in decentralized systems. We present real-world examples, such as Hyperledger Fabric, and new blockchain architectures like OrderlessChain to exemplify the applicability of these technologies.

Keywords: CRDTs · Blockchain · LLMs · AI · Conflict Resolution · Distributed Systems · Collaborative Networks · Scalability · Concurrency

1 Introduction

Among recently developed technologies, blockchain has emerged as one of the most innovative approaches in decentralized systems. Its use spans a wide variety of industries, from finance and supply chain management to healthcare and many more. However, scalability and concurrency issues significantly hinder its adoption on a larger scale. Conventional blockchain systems are often based on consensus algorithms like PoW (Proof of Work) or PoS (Proof of Stake), which, eventually, at the cost of efficiency and speed, increase with the network [1]. Moreover, large-scale implementations face growing transaction latencies and bottlenecks as more participants join the network, and this scalability challenge has real-world consequences. For example, the Bitcoin network processes only about 7 TPS on average while Visa's global payment network can process up to 65,000 TPS [2].

CRDTs recently started gaining importance as a solution to concurrency and consistency problems in blockchains. They allow local updates on distributed replicas, which eventually converge to a consistent state across nodes without any need for a centralized authority or global ordering of transactions [3]. Though CRDT applications have begun to reach domains like collaborative text editing, for example, Google Docs, the use of CRDT in blockchain is at an early stage. The question is how CRDTs can bring the same efficiency to blockchains, overcoming the constraints imposed by current consensus protocols [4].

Large Language Models (LLMs) like GPT-3 are also making their mark insoftware engineering, particularly in automating conflict resolution during code merges. This presents opportunities for leveraging LLMs in blockchain conflict management as well, potentially reducing the reliance on manual interventionfor transaction conflicts [5]. AI's role in blockchain and CRDT systems extendsbeyond this, offering predictive analytics to optimize system performance andaccelerate research in decentralized coordination [6].

This paper aims to explore the convergence of these technologies—CRDTs, LLMs, and AI—within the blockchain ecosystem, using real-world examples and novel insights to showcase the transformative impact they could have on decentralized systems.

2 Background

2.1 Blockchain's Scalability Challenges

The scalability problem of blockchain networks is well-known. Public popular blockchains like Ethereum face problems while demand exceeds the processing capacity. That was vividly demonstrated during the CryptoKitties boom in 2017, where Ethereum's TPS slowed down significantly due to network congestion. Permissioned blockchains like Hyperledger Fabric are partial solutions since they focus on enterprise level scalability by using consensus mechanisms better suited for controlled environments [7]. However, these systems still struggle with transaction ordering, validation, and finality when network participation increases.

Hyperledger Fabric's Execute-Order-Validate (EOV) approach separates execution from transaction ordering to achieve parallelism, yet the system can still experience bottlenecks as transactions need to pass through a central orderingservice [7].

The challenge lies in eliminating this global ordering requirement without sacrificing consistency and security.

2.2 Conflict-Free Replicated Data Types (CRDTs)

CRDTs allow for data synchronization in distributed systems by having independent nodes update data at the same time. These converge over time to a consistent state with no need for coordination between the nodes [8]. This makes

CRDTs very pertinent for blockchain, where transaction conflicts of distributed participants are frequent. Unlike traditional blockchain consensus protocols relying on global ordering, CRDTs enable more parallelism, hence reduced need for strict sequential transaction validation.

A very prominent use of CRDT in the real world is Google Docs. Users can edit a document that is shared with others all at the same time, and the system makes sure that each change is reflected uniformly for all users without them needing to take turns. In reference to blockchain, this could apply to multiple nodes processing transactions at the same time, which may give way to new levels of scalability and efficiency.

3 CRDTs in Blockchain

3.1 The Promise of CRDTs for Blockchain Scalability

The scalability of blockchain networks is poor because the consensus mechanism is always centralized and also creates overhead in terms of transaction validation. CRDTs, by providing the facility for concurrent updates without a requirement of centralized control, indeed present a promising alternative that will improve scalability. Multiple nodes will be able to propose transactions simultaneously with CRDTs, without having to wait for a global consensus. This approach significantly improves transaction throughput and reduces bottlenecks associated with traditional consensus algorithms.

For instance, a permissioned blockchain platform like Hyperledger Fabric [7] might leverage CRDTs in order to move transaction validation much more efficiently. Rather than using a centralized consensus protocol for every single transaction, CRDTs would permit concurrency updates across the network, whereby all nodes would eventuate to the same state. However, incorporating CRDTs into Fabric remains a challenge, especially when it comes to ensuring the consistency and reducing the communication overhead between nodes in a large-scale network.

3.2 Case Studies: FabricCRDT and OrderlessChain

FabricCRDT: This approach extends the Hyperledger Fabric by integrating CRDTs in its architecture. The CRDT extension embeds conflict-free transactions that increase the throughput and reduce the transaction failures. CRDTs help resolve issues related to concurrency; conflicting transactions can automatically be merged without human intervention.

OrderlessChain: This system presents a different direction from traditional blockchain consensus mechanisms, as it uses CRDTs to enable coordination-free execution. By allowing all transactions to execute in parallel, with no particular order, OrderlessChain removes the overhead of the consensus protocols, achieving much better scalability and lower latency.

The above examples present the possibilities of CRDTs within blockchains, but efficient data propagation, handling large-scaled CRDTs, and reduction of communication overhead are challenges that require further research.

3.3 Future Research Directions

While CRDTs are a promising solution, there is still a lot of gap that needs to be addressed. First, it is important that CRDTs are scalable for large and dynamic datasets in real-world blockchain environments. Secondly, there is a need for more research on how CRDTs work with traditional consensus mechanisms and how AI can optimize the performance of CRDTs in decentralized systems.

4 LLMs in Blockchain Conflict Resolution

In this aspect, the integration of Large Language Models into conflict resolution mechanisms for blockchains is quite fitting and opens a completely new perspective on how some of the inherent problems in decentralized transaction validation can be tackled. Though these LLMs, GPT-3 among them, were created with only natural language comprehension and generation in mind, the scope of application is rapidly expanding to cover code conflict resolution and blockchain infrastructures. The section provides an overview of LLMs' potential in the blockchain ecosystem, the automation potential for transaction conflict resolution by LLMs, the challenges they face, and their future potential with respect to improving blockchain scalability and efficiency.

4.1 AI's Role in Optimizing Decentralized Systems

Some of the more advanced usages of AI in recent times revolve around predictive analytics and optimization functions in the management of decentralized networks. In dynamic environments where bandwidth and computing powers are at a premium, AI is adept at optimizing task distribution for improved transaction throughputs and allowing real-time detection of possible network bottlenecks [6]. This application is already manifesting in IoT networks, where blockchain systems are often faced with problems based on low-power devices and restricted connectivity.

With AI-boosted algorithms for CRDTs, adaptation could be made dynamically to preserve efficiency of conflict resolution with regards to fluctuation in the number of nodes. This may be crucial for future blockchains, which would require high scalability with low latency.

4.2 Large Language Models (LLMs) and Merge Conflict Resolution

Merge conflicts in software development arise when various contributors make modifications to the same piece of code. This process may get even more complicated in big projects with multiple branches and contributors. Large language

models like GPT-3 have already shown great potential for automated merge conflict resolution-for example, based on contextual understanding of the code changes and suggestions for appropriate merged versions [5].

For instance, GitHub's Copilot embeds an AI-powered assistant that helps developers to complete code and resolve conflicts. This could be further extended to blockchain environments, where a set of nodes propose possibly conflicting transactions. LLMs might analyze those transactions and propose an optimum conflict resolution, hence reducing the need for explicit conflict management and accelerating the consensus process.

4.3 Applying LLMs to Blockchain Transaction Conflicts

With the growth of software, LLMs have helped in resolving merge conflicts like GPT-3. From understanding changes in code, there can be an intelligent merge of conflicting modifications by LLMs, reducing manual intervention and human error. This is helpful for open-source projects, for example, GitHub repositories, which get updated quite frequently and have a large number of contributors. For instance, GitHub Copilot can even suggest smart merges, enabled through the OpenAI Codex model, and can write code based on context, saving developers much time and reducing the risk of conflicts that could disrupt code integrity.

Similar merge conflicts in blockchain occur when many nodes submit transactions that are conflicting with each other. The issue is all the more critical when there is no central authority in decentralized networks that can sort out disputes. Here, the integration of LLMs into the management of blockchain transactions could provide the system with the autonomy to analyze proposed transactions automatically for detection and also to propose resolutions that keep the integrity of the Distributed Ledger. The role of LLMs could go beyond the simple resolution of conflict. They could also:

- Understand the context of each transaction: Like code in the field of software development, blockchain transactions often depend on the context in which they have been made. An LLM would understand the relations of various transactions, be it financial, asset transfers, or even execution of smart contracts, and ensure their resolution in a manner that maintains the logical and general consistency of the system at large.
- Large-scale automation of transaction validation: In permissioned blockchain systems, multiple nodes participate in the process of transaction validation. The complexity and scale said above can be better handled using LLMs, which will support automated conflict resolution in context and hence reduce human validators' overhead and increase the system's speed and scalability factor.

For example, Hyperledger Fabric—a permissioned blockchain used in enterprise settings—requires consensus and validation from multiple nodes before transactions are confirmed. In the future, LLMs could be employed to facilitate real-time decision-making and ensure that conflicting transactions are automatically

identified and resolved without manual intervention, boosting overall network throughput and reducing latency.

4.4 Challenges and Potential

While the potential benefits of LLMs in blockchain conflict resolution are clear, there are several key challenges to consider.

- Nature of Blockchain Transactions: Blockchain transactions are strictly of a structured format, following some protocols such as JSON or Protobuf, whereas this is not so with natural language. In other words, LLMs would have to be adjusted or fine-tuned to make sense of these particular types of transactions. It probably needs to be specially trained on domain-specific data ahead of it venturing into blockchain environments.
  A real-world example of this challenge is the transition from general-purpose language models to those specialized in certain domains, like legal or medical language. For example, LawGeex has trained AI models to review legal contracts, showing how domain-specific models can outperform general-purpose ones. Similarly, LLMs in blockchain will likely need to undergo fine-tuning with data from blockchain-specific applications, like smart contracts or token transactions, to understand the intricacies of these systems.
- Decentralized nature of blockchain: Naturally, blockchain systems come inherently with no single source of truth or global context that the LLM may rely on to make sense of all the transactions in the network. The LLM would thus need to work in a distributed fashion, coordinating amongst nodes without a centralized command-that most machine learning usually leverages. This further decentralization also challenges the straightforward application of LLMs, whose designs and architectures should cover the distributed nature of blockchain.
  However, recent innovations in federated learning and decentralized AI could provide solutions to this issue. Federated learning enables models to train locally on distributed devices without exchanging raw data between them. This could allow LLMs to operate on individual blockchain nodes, each contributing to the model's learning process while preserving data privacy and security.
- Diversity of transaction types: Blockchains can support a wider range of transaction types with different characteristics and demands than has ever been possible. Some-transactions based on smart contracts, for example-may involve complex state changes, while simple asset transfers only need to model flows of monetary value. The successful LLMs will be able to generalize across a wide variety of transactions, which in this case can be rather complex because of the diversity in the underlying transaction logic and systems where these shall be executed.
  One way to address this could be through the development of multi-modal models that integrate not only natural language data but also numerical and transactional data. For instance, an LLM could process natural language

descriptions of a smart contract, analyze transaction logs, and consider historical context to propose resolution strategies that balance between system requirements and user goals.

Despite these challenges, the potential for LLMs to automate conflict resolution in blockchain systems is immense. As LLMs continue to evolve and become more specialized in understanding domain-specific languages and transaction formats, their ability to handle complex conflicts in decentralized systems will increase. For instance, OrderlessChain, a CRDT-based blockchain, relies on decentralized coordination without requiring global transaction ordering. LLMs could play a key role in such systems by analyzing the incoming transactions, predicting conflicts, and proposing conflict-free merges in real-time, thereby improving performance and scalability.

5 Conclusion

This paper has explored the integration of CRDTs, LLMs, and AI in blockchain systems, illustrating their potential to address the longstanding issues of scalability, concurrency, and conflict resolution. CRDTs can provide an effective way to improve blockchain scalability by allowing for concurrent updates without the need for global transaction ordering. Real-world applications, such as Hyperledger Fabric and OrderlessChain, offer tangible evidence of these improvements. Meanwhile, LLMs offer a novel way to automate conflict resolution in decentralized systems, reducing the need for manual intervention. Lastly, AI's role in optimizing resource allocation and conflict management presents new avenues for research and development. As blockchain systems continue to evolve, these technologies will be key to unlocking their full potential, allowing for more efficient, scalable, and robust decentralized networks.

References

1. Rahul Arulkumaran, Dignesh Kumar Khatri, Viharika Bhimanapati, Anshika Aggarwal, and Vikhyat Gupta. 2023. AI-Driven Optimization of Proof-of-Stake Blockchain Validators. Innovative Research Thoughts, 9(5), 315–333. https://doi.org/10.36676/irt.v9.i5.1490
2. Berneis M, Bartsch D, Winkler H. Applications of Blockchain Technology in Logistics and Supply Chain Management—Insights from a Systematic Literature Review. Logistics. 2021; 5(3):43. https://doi.org/10.3390/logistics5030043
3. Marc Shapiro. Nuno Pregui¸ ca, Carlos Baquero. Conflict-free replicated data types (crdts). Encyclopedia of Big Data Technologies, Springer International Publishing, 2018. https://doi.org/10.48550/arXiv.1805.06358
4. Pezhman Nasirifard, Ruben Mayer, and Hans-Arno Jacobsen. Fabriccrdt: A conflict-free replicated datatypes approach to permissioned blockchains. In Pro ceedings of the 20th International Middleware Conference, Middleware '19, page 110–122, New York, NY, USA, 2019. Association for Computing Machinery. https://doi.org/10.1145/3361525.3361540

5. Jialu Zhang, Todd Mytkowicz, Mike Kaufman, Ruzica Piskac, and Shuvendu K. Lahiri. 2022. Using pre-trained language models to resolve textual and semantic merge conflicts (experience paper). In Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2022). Association for Computing Machinery, New York, NY, USA, 77–88. https://doi.org/10.1145/3533767.3534396
6. Umoga, Uchenna, Sodiya, Enoch, Ugwuanyi, Ejike, Jacks, Boma, Lottu, Oluwaseun, Daraojimba, Obinna, Obaigbena and Alexander. 2024. Exploring the potential of AI-driven optimization in enhancing network performance and efficiency. Magna Scientia Advanced Research and Reviews. 10. 368-378. 10.30574/msarr.2024.10.1.0028.
7. Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstanti nos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukoli´ c, Sharon Weed Cocco, and Jason Yellick. Hyper ledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, EuroSys '18, New York, NY, USA, 2018. https://doi.org/10.1145/3190508.3190538
8. Mihai Letia, Nuno Pregui¸ ca, and Marc Shapiro. Consistency without concurrency control in large, dynamic systems. SIGOPS Oper. Syst. Rev., 44(2):29–34, apr 2010. https://doi.org/10.1145/1773912.1773921

Enhance Container Security using Neural Networks.

Wissam Boudjahfa[1][0009-0003-7305-3760] and Fatima Zohra Filali[1][✉] and Belabbes Yagoubi[1][✉]

[1] Lab CSTL, Mostaganem University, 27000 Mostaganem, Algeria
wissamboudjahfa@gmail.com
fatimazohra.fillali@univ-mosta.dz
byagoubi31@gmail.com

Abstract. Virtualization technology is the key driver in cloud environments. Being able to use more resources without the need of physical machines is impressive. However it has security consequences. Virtualization has multiple types: full virtualization, paravirtualization and OS-level virtualization. Each type faces specific security threats. The most delicate virtualization type is the OS level virtualization. The OS level containerization, also known as uni-kernel virtualization, encounters multiple challenges in today's environment. The ability to multiple instances in the same platform led the IT technology to a revolutionary phase. The main challenge in containers technology is to detect whether the instance is secure or not. As a solution to provide a protected environment to run the application across ubiquitous platforms, the paper suggests a deep learning based model to handle OS virtualization level security risks. The goal of this model is to manage containers vulnerabilities through system calls. The system calls are the common way to differentiate the behavior of a container.

Keywords: Container Security, CVE, virtualization, Unsupervised learning, Gated Recurrent Units (GRU), Autoencoder (AE).

1 INTRODUCTION

Containers are key driver in the IT revolution. Cloud Environment are facing a huge evolution due to the advantages offered by virtualization technology. This technology is beneficial for both hardware and software resources. It provides agility, adaptability, scalability and elasticity[1]. virtualization plays a pivotal role in cloud environments. Due to its abstraction potential, it allows to isolate the software from the underlying hardware. There are multiple types of virtualization: full virtualization, paravirtualization , OS virtualization [1]. This paper focuses on OS-level virtualization. OS virtualization known as containers is a technology that allows multiple instances to run across diverse platforms. The wide adoption of containers became remarkably challenging. Running multiple instances on the same host is more efficient and productive. The fast boot-up, elasticity and low cost oriented IT and non IT services to depend more on OS-level virtualization. With the indispensable container usage, security

risks became more challenging. According to microsoft the attack risks have increased significantly. In 2023, 65% source code vulnerabilities were exhibited during the software development process [2]. Containers' inherent dynamism and ephemeral nature establish them as an optimal choice for swift scalability and effective resource management, yet, this attribute presents security challenges. The transient life-cycle of containers elevates the complexity of maintaining consistent monitoring, applying patches, and ensuring accurate configuration. Therefore, it could potentially expose them to more vulnerabilities and breaches [3] [4] [5]. Container vulnerabilities vary based on different factors. These factors could be outdated software, misconfiguration and weak isolation due to the shared resources. These factors may potentially expose the containers to security threats. Container security threats are classified into categories: vulnerable images, container risks, orchestrator risks, host OS risks and Implementation risks [6]. These risks are caused by different types of vulnerabilities. The vulnerabilities are increasing significantly, Fig.1 shows the statistics of docker vulnerabilities over the last years.

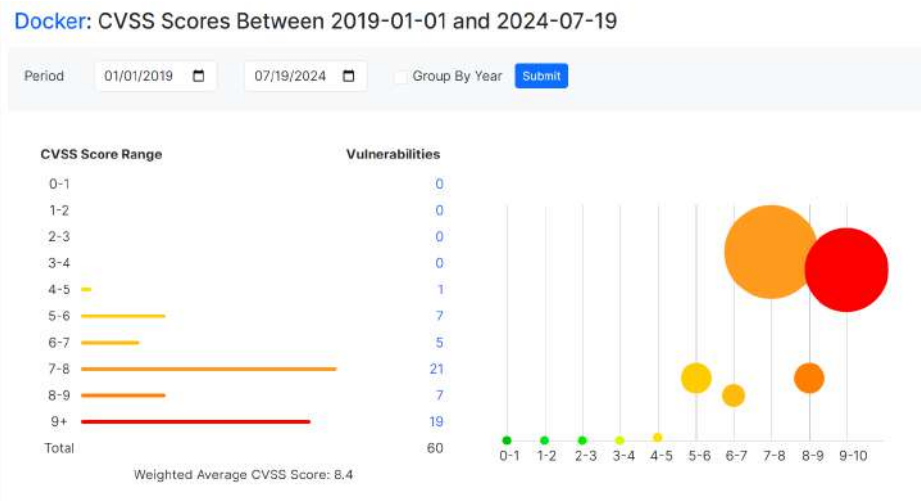

Fig. 1. Docker vulnerability statistics based on severity [7].

The proposed solutions are inefficient, as they either utilize image scanning tools like inspec, rely on basic Linux security modules or improve isolation by adding an additional virtualization layer [8] [9] [10]. The interaction between a platform and its running instances is done via system calls. System calls are the best way to examine an anomalous behavior. Other solutions like system calls whitelisting were proposed[11]. System calls are dynamic sequential data. The traditional solutions appear to be limited in terms of dealing with the core issues. The vulnerabilities exploit is the principale issue in the field of managing container security. The desire to find a better solution to enhance the containers security oriented the researchers to investigate new dynamic solutions. The solutions are based on machine learning and deep learning techniques. These techniques proved their power in the security field [12] [13]. The

paper follow the lead of previous researches [14] [15]. It proposes an unsupervised Gated recurrent Units (GRU) based autoencoder. Our team chose the GRU due to its advantages concerning sequential data [16]. GRU is a special recurrent neural network. The model is designed to handle long dependencies and sequential data. GRU addresses the computational complexity by using two gates: the update gate, and the reset gate. Maintaining and discarding data in GRU is done by the reset gate. GRU offers benefits that includes agile sequence process and low usage of computational resources. The autoencoder, in the other hand, reduces noise and extract relevant features[17] [18]. The paper merges the powers of the two techniques to benefit from their advantages. To detail more, we organized the paper as follows: Section 1 gives a brief overview and introduces our work. Section 2 highlights the previous works related to our research. Section 3 provides more detail about : the experiment, the model, the results and the limitations. Finally, we conclude our paper in section 4.

2 CONTRIBUTION

The purpose of this paper is to introduce an interactive, dynamic solution that will reinforce security mechanisms. The solution includes the use of a neural network model to minimize the OS-level virtualization security issues. The paper's main contribution is as follows:

1. *The Integration of System Calls:* to inspect the low-level interaction between container and its host environment. The use of system calls is needed, their analysis reveals the behavioral changes of the instance. These changes may indicate a potential threat.
2. *The neural network model for anomalies detection:* we provide an unsupervised GRU based AE(Autoencoder). The to monitor the behavior of the sequential data and minimize the noise. The point from combining these models relies in the AE ability to extract relevant features from system calls and the GRU capacity to learn sequential dependencies efficiently, while treating and handling the problems of gradient descent. The model learns the system calls patterns which is the main identifier for the containers malicious or benign acting.

3 RELATED WORK.

The machine learning and deep learning are playing a crucial role in the security field, they have proved their effectiveness in detecting defects and anomalies. In the following subsections we will investigate some of the related works. Fatih.E.(2019) uses a hybrid model that combines Long Short Term Memory (LSTM), Bayesian model and Support vector machine (SVM) to improve anomaly detection in network attacks [12]. Tao et al. (2018) proposes a new framework. The authors claim that this solution outperforms K-nearest neighbor (KNN) and neural networks [13]. The experiments concerning the use of Machine learning (ML) and Deep Learning (DL) solutions reached the containerization security. The shared OS-kernel is the main actor for con-

tainers execution. A 2023 vulnerability analysis revealed an alarming number of security gaps, with Overflow and Memory Corruption vulnerabilities responsible for 127 instances [19]. Segregated by type, DoS attacks emerged as the most prevalent, accounting for 27 vulnerabilities, followed by privilege escalation with 20 vulnerabilities [19], along with various other types including bypass and information leaks. The short-lived nature and dynamicity of containers elevate the level of risks and security challenges. To address these significant challenges caused by the complex and rapid-nature of containers, recent research has employed ML and DL techniques and demonstrated their impact in mitigating and detecting containers threats. Chen X, et al. (2021) proposes a system call behavioral analysis using LSTM [20]. According to the authors, data collection was interactive using *ptrace*. The model has 4 layers and the learning rate is equal to 0.005. They used cross-entropy loss function, slide window and Adam optimization. The model's results were satisfactory, reaching an accuracy of 91.24%. Pinnamaneni et al. (2022) focuses on vulnerabilities detection at the code-level [21]. The authors use a static method to analyze the code within the image. The paper aims to detect the vulnerable Python libraries. The machine learning techniques used are: Decision tree (DT), Naïve Bayes, SVM, Random Forest (RF), KNN, Gradient boost (GB), X-Gradient Boost (XGB). The accuracy of the models varies between 73.5% and 90%. The papers Lin et al. (2020) and O.Tunde-Onadele et al. (2019) focus on dynamic analysis of system calls using different types of machine learning algorithms [14] [15]. These methods are K-means, KNN, KNN + PCA, Self-Organizing Map (SOM)[15] and autoencoders [14]. The papers use a dataset of system calls to detect the behavior of containers. Autoencoders achieved a detection rate of 93.9%, while the SOM achieved 78.5% [14] [15]. Gantikow et al.(2020) seeks to evaluate the performance of GRU, LSTM and a hybrid model [23]. Each model has two layers. These models aim to examine the sequences of system calls. The authors did not mention any metric values.

4 EXPERIMENT.

Containers are prone to vulnerabilities, which are classified according to their type of exploit [19][3][4]. The types of exploits are: code execution, bypass, privilege escalation, SQL-injection, information leak, denial of service, memory corruption, overflow etc. To address these vulnerabilities, different solutions were proposed to minimize the attack surfaces. The countermeasures are, basically, static scanning tools [12] [13], and isolation layer (hardware, virtualization...). Given the complex nature of vulnerabilities and the limitations that static solutions face with containerized application related threats, researchers are moving towards interactive real-time solutions. They opt for solutions that involve machine learning and deep learning. This orientation led us to explore the dynamic solutions. This section describes the model proposed by the paper.

4.1 Hardware

The experiment was conducted on a local machine. The machine is a hp Zbook workstation, equipped with an 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz processor, a disk capacity of 1TB SSD, and 32 GB of RAM. This workstation has 16 threads, and two graphical cards: an integrated, Intel® UHD Graphics, and a dedicated, NVIDIA RTX A2000.

4.2 Dataset

In this study we use a system calls based dataset [23]. It consists of benign and malicious behavior collected by Cui.P. et al. (2020) using sysdig [25]. The authors used Docker containerization platform version 18.03.1-ce, build 9ee9f40. Sysdig recorded a detailed summary of each container calls. The collected calls are: the system call name, caller process, timestamp, and passed arguments. The paper proposes an LSTM based framework to pre-process the data [24]. The dataset contains 1.36 field of benign behavior arguments. The attacks have more fields of arguments. The malicious behavior was collected from seven attack behaviors, which are categorized into the following types: 1/ Brute force login, 2/ Docker Escape, 3/ Malicious script, 4/ Meterpreter, 5/ Remote shell, 6/ SQL injection and 7/ SQL misbehavior. The Attacks were triggered in a simple containerized MySQL server application. The malicious behavior is accumulated from *bind, chdir, readlink, socketpair, get pid* system calls The primary behavior class analysis permits to learn patterns and long dependencies. The main purpose is to discern suspicious behavior triggered by exploiting vulnerabilities. This paper suggests a GRU based Autoencoder to monitor system calls behavior. The results of the model's performance are discussed in Section 4.

4.3 Model explanation and results

The proposed model consists of an unsupervised autoencoder. The model loads balanced data between the normal and malicious behaviors. A Z-score normalization is used to normalize the balanced dataset. The encoder is defined by 3 GRU layers containing 224 nodes, a latent space of 16 and a decoder of 3 GRU layers containing 224 nodes. The dataset is split into a 0.8/0.2 of train/test ratio. The Adam optimisation and the MSE loss function are used to train the model.The batch size, num-epoch and learning rate are 256, 50 and 0.001 respectively. The model uses early stopping with patience of 5 and a dropout probability of 20% to reduce overfitting. In the training phase the values of the loss function and validation function were decreasing and low. According to the results in Fig.2, the model had an early stopping at the epoch 41.

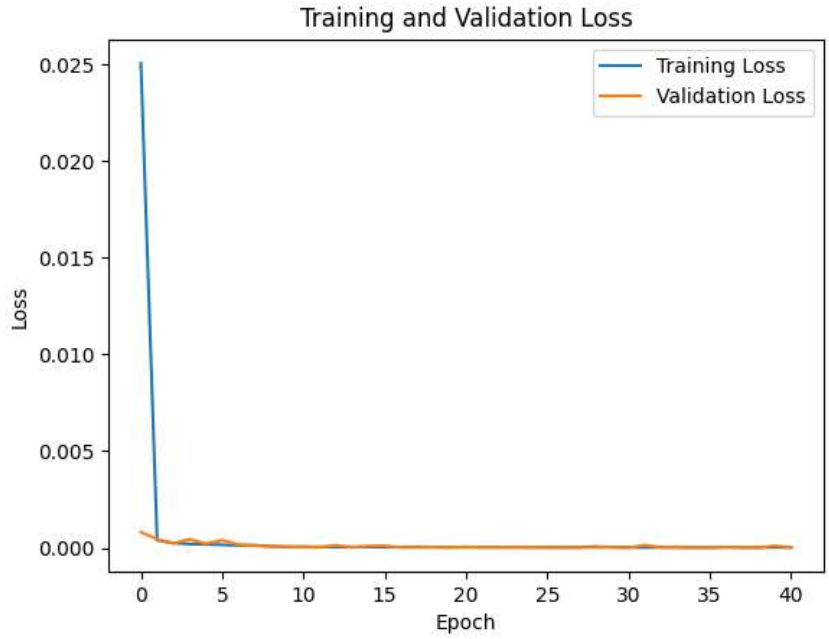

Fig. 2. Training and validation loss values over 41 epochs.

We applied the trained model on the test dataset. The metrics used for testing are: Mean Reconstruction Error, Standard Deviation of Reconstruction Error, Mean Squared Error (MSE) and Mean Absolute Error (MAE). Despite the model's complexity, the values obtained demonstrates its high performance in converging and learning patterns. Table 1 summarize the values obtained.

Table 1. Summary of the calculated errors.

Errors	Values
Mean Reconstruction Error	0.0006021165754646063
Standard Deviation of Reconstruction Error	0.004362785257399082
Mean Squared Error (MSE)	1.9396442439756356e-05
Mean Absolute Error (MAE)	0.0006021165754646063

4.4 Limitations and future work.

The overall performance of the model was satisfying. The models gave good results in the test phase. However, the model needs to be tested on larger datasets and in a production environment. Due to the dynamic nature of real world vulnerabilities, we think that the model needs modifications to be able to address the real word interac-

tion vulnerabilities. As a future work we suggest to experiment other datasets and use new learning technique to adapt the model for the interactive nature of vulnerabilities.

5 CONCLUSION.

The aim of this paper is to find a solution that allows the containers security reinforcement. Monitoring container behaviors was necessary for the ability to adress their security challenges. System calls are the best way to examine the interaction between the paltform and its running instances. Our team focuses on proposing a solution that inspects system calls and analyze the behavior of containers. The purpose of the solution is to monitor the changes on system calls in order to manage vulnerabilities. To do so, our team suggests a GRU based autoencoder model. This model observes behavioral attacks in containers. The main actor is the infected containers system calls. These calls are collected through sysdig and preprocessed using an LSTM based framework[24]. The preprocessed data is fed to the AE model. The model showed a good performance. Our model converges and learn patterns efficiently. The model and its results are explained in Section 3. Finally, and as a future work, we plan to test the model on different datasets and adapt it to address real world scenarios.

References

1. Bhardwaj, A., Krishna, C.R. Virtualization in Cloud Computing: Moving from Hypervisor to Containerization—A Survey. Arab J Sci Eng 46, 8585–8601 (2021). https://doi.org/10.1007/s13369-021-05553-3.
2. Microsoft risk rapport [online]: https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/2024-State-of-Multicloud-Security-Risk-Report.pdf?culture=en-us&country=us; last accesd 2024/7/30.
3. Container Security Site, container breakout vulnerabilities https://www.container-security.site/attackers/container_breakout_vulnerabilities.html, last accessed, 2024/07/19.
4. Container Security Site CVE_list: https://www.container-security.site/general_information/container_cve_list.html; last accessed, 2024/07/19.
5. Amazon Linux Security Center [Online]: https://alas.aws.amazon.com/index.html last accessed,2024/07/04.
6. UK GOV,DWP procurement: security policies and standards [Online]: https://assets.publishing.service.gov.uk/media/669a2ed2ce1fd0da7b5928bd/dwp-ss-011-security-standard-containerisation.pdf; last accessed; 2023/01/31.
7. Docker vulnerabilities charts [Online]: https://www.cvedetails.com/cvss-score-charts.php?fromform=1&vendor_id=13534&product_id=&startdate=2019-01-01&enddate=2024-07-19, last accessed, 2024/07/19.
8. Chen, J. et al. (2019) ‘A container-based DoS attack-resilient control framework for real-time UAV systems’, 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.1222–1227.

9. Tian, D. et al. (2019) ‘A practical Intel SGX setting for Linux containers in the cloud’, Ninth ACM Conference on Data and Application Security and Privacy, pp.255–266.
10. Inspec [Online]: https://community.chef.io/tools/chef-inspec; last accessed, 2024/07/11.
11. S. Kim, B. J. Kim and D. H. Lee, "Prof-gen: Practical Study on System Call Whitelist Generation for Container Attack Surface Reduction," 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 2021, pp. 278-287, doi: 10.1109/CLOUD53861.2021.00041.
12. Rekha Gangula, V. Murali Mohan, Ranjeeth Kumar,A comprehence study of DDoS attack detecting algorithm using GRU-BWFA classifier, Measurement: Sensors, Vol .24, (2022),https://doi.org/10.1016/j.measen.2022.100570.
13. I I Kurochkin and S S Volkov, IOP Conf. Series: Materials Science and Engineering 927 (2020), doi:10.1088/1757-899X/927/1/012035.
14. Lin, Y., Tunde-Onadele, O. and Gu, X., 2020, December. CDL: Classified Distributed Learning for Detecting Security Attacks in Containerized Applications. In Annual Computer Security Applications Conference (pp. 179-188) https://doi.org/10.1145/3427228.3427236.
15. O.Tunde-Onadele, J. He, T.Daiand X.Gu, "A Study on Container Vulnerability Exploit Detection,"2019 IEEE International Conference on Cloud Engineering (IC2E), Prague, Czech Republic, 2019, pp. 121-127, doi: 10.1109/IC2E.2019.00026.
16. Sepp, H., Jürgen, S.: Long Short-Term Memory, Neural Computation, 9(8), pp: 1735-1780, (1997). doi :10.1162/neco.1997.9.8.1735
17. Raffaella, Esposito., Monica, Casella., Nicola, Milano., Davide, Marocco. (2023). Autoencoders as a Tool to Detect Nonlinear Relationships in Latent Variables Models. 1012-1016. doi: 10.1109/metroxraine58569.2023.10405761
18. Sander, de, Bruin., Vadim, Liventsev., Milan, Petkovic. (2021). Autoencoders as Tools for Program Synthesis. arXiv: Artificial Intelligence.
19. Docker vulnerabilities platform: https://www.cvedetails.com/vendor/13534/Docker.html, last accessed, 2024/07/19.
20. CHEN Xing-shu, JIN Yi-ling, WANG Yu-long, JIANG Chao, WANG Qi-xu. Anomaly Detection of Processes Behavior in Container Based on LSTM Neural Network[J]. Acta Electronica Sinica, 2021, 49(1): 149-156. https://doi.org/10.12263/DZXB.20190220.
21. J. Pinnamaneni, N. S and P. Honnavalli, "Identifying Vulnerabilities in Docker Image Code using ML Techniques," 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Ravet, India, 2022, pp. 1-5.
22. H. Gantikow, T. Zöhner and C. Reich, "Container Anomaly Detection Using Neural Networks Analyzing System Calls," 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Västerås, Sweden, 2020, pp. 408-412, doi: 10.1109/PDP50117.2020.00069.
23. Dataset-master [Online]: https://github.com/PinchenCui/Dataset/tree/master; last accessed, 2024/06/30.
24. Pinchen Cui and David Umphress. 2020. Towards Unsupervised Introspection of Containerized Application. In Proceedings of the 2020 10th International Conference on Communication and Network Security (ICCNS '20). Association for Computing Machinery, New York, NY, USA, 42–51. https://doi.org/10.1145/3442520.3442530.
25. Sysdig [Online]: https://sysdig.com, last accessed, 2024/07/15.

Data Synergy in Healthcare: Exploring Approaches to Medical Data Integration

Medjahed Amina Fatima Zohra[1], Guerroudji Meddah Fatiha[2] ,Ougouti Naïma Souâd[3]

[1] *University of Sciences and Technology of Oran - Mohamed Boudiaf*
[2] *University of Sciences and Technology of Oran - Mohamed Boudiaf*
[3] *University of Sciences and Technology of Oran - Mohamed Boudiaf*

Abstract. The rapid digitization of healthcare has resulted in an unprecedented influx of diverse and heterogeneous data sources, including electronic health records (EHRs), biometric data, medical imaging, and information from wearable devices. This paper addresses the critical challenges of integrating these varied data types, which often differ significantly in semantics, structure, and syntax. The complexity of this integration poses substantial barriers to effective medical data management, hindering healthcare professionals' ability to obtain a comprehensive view of patient information. We explore current methodologies and propose innovative frameworks for data synergy that enhance accessibility and usability, ultimately aiming to improve patient care and health outcomes. Furthermore, our findings highlight the urgent need to address privacy and compliance concerns while leveraging advanced technologies to facilitate seamless data integration within the healthcare sector.

Keywords: Data integration, Big Data, Heterogeneous data, medical data, machine learning

1 Introduction

The mere presence of data, even in large volumes, does not guarantee that information needs will be effectively and promptly addressed. In the realm of Big Data, deriving meaningful insights from these assets presents significant challenges, particularly concerning the integration of diverse and heterogeneous data sources. This complexity is especially evident in the medical field, where the rapid digitization of patient records—including X-rays, scans, sensor readings, lab results, prescriptions, and data from monitoring devices—has led to an overwhelming influx of biomedical information. The variety of these data sources, which differ in semantics, structure, and syntax, creates substantial hurdles in achieving a cohesive view essential for efficient medical data management [1]. Predictions indicate that the exponential growth of healthcare data will continue, encompassing electronic health records (EHR), patient-reported outcomes, biometric data, medical imaging, biomarker data, wearable devices, and genomic information. This data emerges from a wide array of heterogeneous sources, including medical service providers, pharmaceutical companies, public health organizations, researchers, and insurance providers, as illustrated in Figure 1.

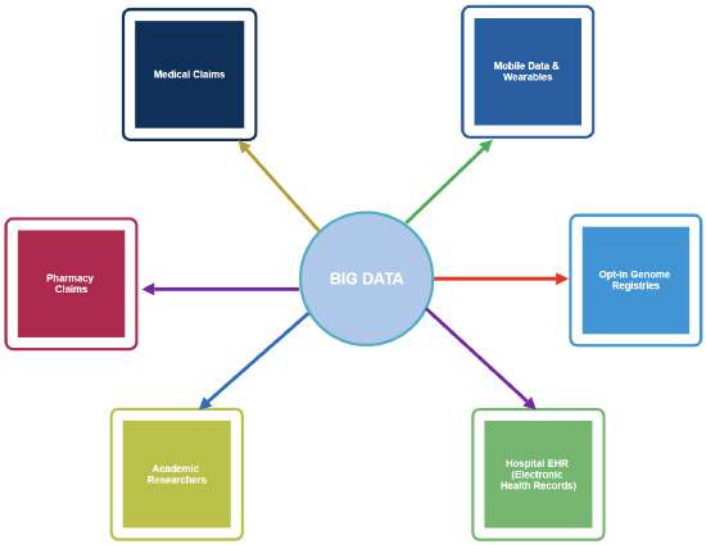

Fig. 1. Sources of big data in healthcare **[4].**

Integrating extensive real-world clinical datasets—such as combining Electronic Medical Records (EHR) with omics data and targeted biochemical and hormonal analyses—enables the identification of new diagnostic and therapeutic tools and aids in capturing the intricate complexities of diseases [4]. An integrated view of medical data is crucial for generating new insights and knowledge. Zhang et al. [2] highlighted the importance of synthesizing information from multiple sources in the healthcare domain, emphasizing that this integration is crucial for thoroughly assessing the diverse risk factors associated with disease manifestation. Their work underscores the need for a comprehensive approach to data integration, combining various datasets to gain a more complete understanding of the factors that contribute to health conditions. This holistic view, achieved by unifying disparate data sources, is essential for improving diagnosis, treatment, and overall patient care outcomes

2 Background

2.1 Data Integration

Data integration is essential in today's data-driven landscape, as it addresses the growing complexity and size of datasets by bringing together disparate data sources, harmonizing them, and providing a unified view. This unified access enables organizations to fully leverage their data for enhanced understanding, informed decision-making, and operational efficiency [4]. According to Kamil and Amyotte [5], there are two main

motivations behind integrating multiple data sources. First, it simplifies information access by streamlining and centralizing data from various systems. Second, it creates a more comprehensive dataset by combining complementary data from different sources, providing a richer foundation for analysis and decision-making. Similarly, Ziegler and Dittrich [3] emphasize that integration not only facilitates better access by offering a unified view of different systems but also enhances the overall utility of the data by merging information from diverse sources. By combining these perspectives, data integration emerges as a critical process for gaining deeper insights and improving efficiency across various domains.

2.2 Heterogeneity Problems

Data integration faces a variety of challenges, with one of the most significant being the heterogeneity of data sources [6]. Dong and Naumann [7] highlight several key issues in this area, including heterogeneity at both the schema and instance levels. At the schema level, different data sources often use distinct schemas to describe the same domain, while at the instance level, the same real-world entity may be represented in different ways across various sources. To address these challenges, several solutions have been proposed, the table below summarizes key integration methods along with their advantages and disadvantages:

Table 1. Various approaches used in integrating big medical data [6][7].

Approach	Advantages	Drawbacks
Schema Mapping	Enhances interoperability between different systems, allowing for smoother data exchange and integration.	Can be complex and time-consuming to set up, requiring careful planning and expertise.
Schema Matching	Automates the identification of equivalent elements across different schemas, improving efficiency in data integration processes.	May struggle with ambiguous or incomplete data, potentially leading to mismatches.
Data Fusion	Improves overall data quality by consolidating information from multiple sources into unified records, reducing redundancy.	Conflict resolution can be challenging, particularly when dealing with conflicting information from different sources.

2.3 Big Data

Over the past two decades, advancements in technology have led to a surge of data across various fields, including healthcare, scientific sensors, user-generated content, financial records, and more. the term "Big Data" has been widely defined in the literature as datasets too large or complex to be efficiently processed by traditional IT systems, software, or hardware within a reasonable timeframe. Initially, Big Data was characterized by the 3V model, which refers to the high volume, high velocity, and high variety of data [1]. More recently, this concept has been expanded to a 5V model, adding two new dimensions: Value and Veracity, further refining the definition of Big Data [14].In the healthcare sector, Big Data has been defined in various ways, with one notable definition describing it as "high volume, high diversity in biological, clinical, environmental, and lifestyle information collected from individual patients to large cohorts, across multiple time points, and related to their health and wellness status. Addressing the challenges of Big Data in healthcare has led to the development of several advanced solutions, leveraging state-of-the-art technologies such as the Apache Hadoop framework, NoSQL databases, and Cloud computing [15].

3 INTEGRATION OF BIG HEALTHCARE DATA - THE SURVEY

The field of biomedicine is a typical example of a sector where the number and volume of data sources have experienced exponential growth over the past decade, with projections indicating that this growth will continue at a rapid pace in the coming decade. It is expected to surpass one zettabyte per year by 2025 [8]. Let's consider a scenario. During each visit, doctors and nurses capture the patient's medical history, including allergies, previous medical interventions, and medications. Additionally, there are increasing opportunities to collect health data. Patients use wearable medical devices and other health monitoring devices, and take advantage of telemedicine services that allow them to receive health information and appointments through telecommunication devices, leading to an even greater and enormous data flow.
Health data comes from a multitude of sources, including medical devices and connected objects that upload data 24/7. While this trend helps individuals become more involved in their health, it generates an excessive amount of wearable data. This is a positive trend that encourages patients to take greater responsibility for their health, but it also results in a huge volume of data that raises concerns about privacy and compliance.[9]. To provide healthcare professionals with a comprehensive picture of their patients, it is necessary not only to process and integrate data from various health-related sources but also to present it in a user-friendly manner for doctors, nurses, researchers, and patients themselves.

3.1 Data Formats

Data in the medical domain exists in a variety of formats, including structured, semi-structured, and unstructured types. Structured data, while coming in many forms, is a primary source of information, encompassing elements such as patient medical histories and biological measurements. On the other hand, semi-structured data, often found in hard copy format, must be digitized to facilitate integration. Although it follows certain formats, like medication lists or lab results, semi-structured data still presents challenges when it comes to harmonization [10]. Unstructured data, such as handwritten prescriptions or doctor's notes, adds another level of complexity. This data varies significantly between sources and often requires digitization using Optical Character Recognition (OCR) technology before it can be integrated into a database. Additionally, unstructured images, such as those from surgical procedures or patient-submitted photos, present unique difficulties due to variations in size, orientation, and other attributes. Pathological data, which is predominantly quantitative but also includes guidelines, reference ranges, and measurements, may require digital storage in different formats. The process of extraction and loading involves gathering data from various sources, transforming it, and then standardizing it into a common format by applying recognized columns for uniformity [11]. In today's data-driven landscape, transforming this vast amount of information into actionable insights demands the development of scalable and innovative tools and techniques [9]. The following are some of the key methods proposed for integrating medical data to support informed decision-making in diagnosis and treatment.

3.2 Overview of Approaches for Big Medical Data Integration

The integration of large-scale medical data necessitates approaches capable of addressing the inherent challenges associated with heterogeneous data sources, structures, and formats. Following a comprehensive evaluation of various integration strategies, Data Consolidation, Data Virtualization, and Data Propagation were selected for their distinct advantages and proven applicability in healthcare data integration. This selection was informed by previous research in the fields of healthcare and big data integration Sreemathy et al. [13] highlighted these approaches as particularly effective in managing the complexity and real-time demands characteristic of medical data systems. As shown in Table 2, these approaches each offer unique benefits:

Table 2. Various approaches used in integrating big medical data [12][13].

Approach	Advantages	Drawbacks
Data Consolidation	- Enables filtering and cleaning of the imported data. - Transforms and structures retrieved data into a more precise format.	- Requires frequent data refreshes to maintain up-to-date content for users. - The resulting structure may not always accommodate ad-hoc queries or unforeseen questions

Data Virtualization	- Keeps data in the original source, avoiding the need to copy large volumes. - Provides real-time access to current information.	-Schema changes in the source require updates to the federated schema. - Data cleansing must be done on-the-fly, which can be challenging. - Performance can suffer due to reliance on the query capabilities of federated data sources.
Data Propagation	- Supports near real-time updates across integrated data sources. - ETL processes can be combined with propagation for real-time data warehousing. - Ensures transparent integration of data sources in terms of location, source, and structure.	- Achieving high performance requires specialized tools and technologies to handle synchronization.

3.3 Machine Learning's Impact on Healthcare integration

The identified trends in machine learning (ML) applications in medical and healthcare analytics hold the potential to transform medical practice, research, and policymaking. According to Ahmed et al. [16], ML algorithms enhance diagnostic accuracy and enable early detection of diseases by analysing intricate medical data, such as imaging, genetic information, and patient records. These tools leverage pattern recognition and predictive modelling to detect subtle anomalies, assess patient risk, and facilitate timely interventions, thereby improving patient outcomes and reducing healthcare costs. For instance, DeepMind's development of an algorithm to predict acute kidney injuries in hospitalized patients, using a large dataset from Veterans Affairs hospitals, demonstrates the potential of combining electronic health record (EHR) data with ML to enhance predictive capabilities and enable early interventions.[22] Despite its promising results, challenges remain in implementing such algorithms effectively in real-world clinical settings. In their work, Wang et al. [17] highlight how ML algorithms are also crucial in personalized therapy and precision medicine. These algorithms help create customized treatment plans based on individual patient characteristics, preferences, and genetic profiles. By analysing vast datasets, ML can identify biomarkers, predict treatment outcomes, and optimize therapeutic plans, particularly for complex diseases like cancer, cardiovascular conditions, and neurological disorders.

A major goal in artificial intelligence (AI) is to develop systems that can autonomously learn from data and make predictions without human intervention. Autonomous machine learning, along with automated machine learning (AutoML), aims to fully automate the ML process, including generating predictions for new datasets independently. These technologies have shown considerable success in real-world applications, such as automatic speech recognition, self-driving vehicles, and even mastering complex tasks.[18]. Innovations in wearable devices, as noted in recent developments, offer a truly patient-centric approach, where each patient's long-term data is compared to their own historical records, rather than those of other patients with similar conditions. This personalized comparison could lead to more targeted and individualized health interventions [16]. For AI and ML to gain widespread adoption in healthcare, key challenges such as reproducibility, cost-efficiency, data acquisition time, resolution, and confidence in results must be addressed. Given the variability in patient cohorts, data quality, and protocols, ML systems will do more than just classify data. As noted by experts, these systems will assist healthcare professionals in discovering new insights, generating novel questions, and understanding complex biological processes [19].

3.4 Ethical Considerations

Investigate the privacy concerns related to collecting health data through wearable (e.g., fitness trackers, continuous glucose monitors). Key concerns include data sensitivity, security vulnerabilities, and unclear ownership. To address these, encryption, authentication, and anonymization are used to protect data. [21] . Additionally, analyse the effectiveness of informed consent processes for individuals contributing their health data, exploring strategies to enhance transparency and ensure that individuals fully understand how their data will be used [20].

4 Conclusion

In conclusion, this paper underscores the transformative potential of integrating diverse medical data sources to advance healthcare and improve patient outcomes. Although not demonstrating specific cases, we highlight how data integration can bridge information gaps, leading to more informed, efficient care delivery. Realizing this vision requires close collaboration among healthcare providers, researchers, and technology developers to tackle the challenges of growing data complexity and volume. Machine learning along side traditional integration methods, offers powerful tools to analyse large datasets, uncover meaning patterns, predict outcomes and predict and personalise treatments. By overcoming existing barriers to data accessibility and usability, effective data integration can enable a more responsive and patient-centred healthcare system

5 References

1. Thirumahal, R., Sudha Sadasivam, G., & Shruti, P. (2022). Semantic integration of heterogeneous data sources using ontology-based domain knowledge modeling for early detection of COVID-19. *SN Computer Science*, *3*(6), 428.
2. Zhang, H., Guo, Y., Li, Q., George, T. J., Shenkman, E., Modave, F., & Bian, J. (2018). An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival. *BMC medical informatics and decision making*, *18*, 129-147.
3. Ziegler, P., & Dittrich, K. R. (2007). Data integration—problems, approaches, and perspectives. In *Conceptual modelling in information systems engineering* (pp. 39-58). Berlin, Heidelberg: Springer Berlin Heidelberg..
4. Farooqui, N. A., & Mehra, R. (2018, December). Design of a data warehouse for medical information system using data mining techniques. In 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 199-203). IEEE.
5. Kamil, M. Z., Khan, F., Amyotte, P., & Ahmed, S. (2024). Multi-source heterogeneous data integration for incident likelihood analysis. Computers & Chemical Engineering, 108677.
6. Bergamaschi, S., Beneventano, D., Guerra, F., & Orsini, M. (2011). Data integration. *Handbook of conceptual modeling: theory, practice, and research challenges*, 441-476.
7. X. L. Dong and F. Naumann, ''Data fusion,'' VLDB Endowment, vol. 2, no. 2, pp. 1654–1655, Aug. 2009. doi: 10.14778/1687553.1687620.
8. Vidal, M. E., Jozashoori, S., & Sakor, A. (2019, June). Semantic data integration techniques for transforming big biomedical data into actionable knowledge. In 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS) (pp. 563-566). IEEE
9. Wu, Z., & Trigo, V. (2021). Impact of information system integration on the healthcare management and medical services. International Journal of Healthcare Management, 14(4), 1348-1356.
10. Mirza, B., Wang, W., Wang, J., Choi, H., Chung, N. C., & Ping, P. (2019). Machine learning and integrative analysis of biomedical big data. Genes, 10(2), 87.
11. Narayanan, M., & Cherukuri, A. K. (2018). Verification of cloud based information integration architecture using colored petri nets. International Journal of Computer Network and Information Security, 15(2), 1.
12. Mousa, A. H., & Shiratuddin, N. (2015, December). Data warehouse and data virtualization comparative study. In *2015 international conference on developments of E-systems engineering (DeSE)* (pp. 369-372). IEEE.
13. Sreemathy, J., Durai, K. N., Priya, E. L., Deebika, R., Suganthi, K., & Aisshwarya, P. T. (2021, March). Data integration and ETL: a theoretical perspective. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 1655-1660). IEEE.
14. Deshpande, P., Rasin, A., Brown, E., Furst, J., Raicu, D. S., Montner, S. M., & Armato, S. G. (2018, October). Big data integration case study for radiology data sources. In *2018 IEEE Life Sciences Conference (LSC)* (pp. 195-198). IEEE.
15. Jung, H., & Chung, K. (2021). Social mining-based clustering process for big-data integration. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 589-600.
16. Wang, R. C., & Wang, Z. (2023). Precision medicine: disease subtyping and tailored treatment. *Cancers*, *15*(15), 3837.
17. Ahmed, Z., Mohamed, K., Zeeshan, S., & Dong, X. (2020). Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine. *Database*, *2020*, baaa010.

18. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 308-318).
19. Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep learning for healthcare: review, opportunities and challenges. Briefings in bioinformatics, 19(6), 1236-1246.
20. Cilliers, L. (2020). Wearable devices in healthcare: Privacy and information security issues. *Health information management journal*, *49*(2-3), 150-156.
21. Awotunde, J. B., Jimoh, R. G., Folorunso, S. O., Adeniyi, E. A., Abiodun, K. M., & Banjo, O. O. (2021). Privacy and security concerns in IoT-based healthcare systems. In *The fusion of internet of things, artificial intelligence, and cloud computing in health care* (pp. 105-134). Cham: Springer International Publishing.
22. Keizer, R. J., Dvergsten, E., Kolacevski, A., Black, A., Karovic, S., Goswami, S., & Maitland, M. L. (2020). Get real: integration of real-world data to improve patient care. *Clinical Pharmacology & Therapeutics*, (4), 722-725.

Hybrid approach for task scheduling optimization in cloud computing environments.

Hadjer Fatima Rouam Serik[1] and Baghdad Atmani[1]

[1] Lab CTSL, Mostaganem University, 27000 Mostaganem, Algeria

hadjerserik6@gmail.com

baghdad.atmani@univ-mosta.dz

Abstract. In a world where the demand for cloud services continues to grow, optimizing the scheduling of independent tasks is essential for ensuring optimal efficiency. This research aims to improve this scheduling by proposing a hybrid approach that combines machine learning techniques, particularly Q-learning, with classic meta-heuristics, such as the Grey Wolf Optimizer (GWO). The primary goal of this hybridization is to optimize resource allocation while adapting to variations in the workload of VMs, thereby offering more robust and effective solutions. Current contributions include the formulation of a conceptual model that integrates these two approaches to overcome the limitations of traditional methods, such as premature convergence and the difficulty of exploring solution spaces. Future perspectives focus on the experimental implementation of this model using the CloudSim simulator, as well as evaluating the performance of the hybrid approach in terms of resource usage costs and waiting times. The current state of advancement reflects a solid theoretical foundation and readiness for practical testing.

Keywords: Cloud Computing, Task Scheduling, Machine Learning, Reinforcement Learning, meta-heuristic.

1. Introduction

Task management in cloud computing environments has become a critical issue in a world where the demand for cloud services continues to grow. The challenges associated with the effective scheduling of independent tasks include resource management, adaptation to workload variations, and optimization of system performance. The importance of this issue lies in its ability to ensure optimal resource utilization while minimizing operational costs. Effective scheduling is therefore essential to ensure user satisfaction and the competitiveness of cloud service providers [1].

The objectives of this paper are to improve the dynamic scheduling of tasks in cloud computing environments by proposing a hybrid approach that combines machine learning techniques, particularly Q-learning, with classical meta-heuristic algorithms, such as the Grey Wolf Optimization algorithm. This motivation is based on the idea that the

hybridization of these techniques can overcome the limitations of traditional methods, such as premature convergence and difficulties in exploring solution spaces.

This research focuses on optimizing the dynamic scheduling of tasks in cloud computing environments. Section 2 provides a synthesis of the state of the art, examining machine learning-based methods as well as those based on the hybridization of metaheuristics and machine learning. Section 3 presents a general description of the proposed hybrid approach. Section 4 introduces a conceptual model of the approach to optimize task scheduling. Finally, Section 5 summarizes the contributions of the research, including the current progress, and outlines future perspectives, particularly the experimental implementation and evaluation of the hybrid approach's performance

2. Review of Task Scheduling Algorithms

This section provides a literature review on dynamic scheduling techniques for independent tasks in cloud computing. Researchers often use Reinforcement Learning (RL) for its ability to learn and easily adapt to changes in workload and resources, which is essential for quick decision-making in cloud computing. Furthermore, neural networks are also utilized to adjust decisions based on variations in environments, thanks to their capacity to handle large amounts of data. Additionally, Deep Reinforcement Learning (DRL) techniques have gained popularity as they enable the management of more complex environments, especially when the state space becomes too vast, allowing for better adaptation to the dynamic scenarios of cloud computing.

Ding et al. [2] presented an approach based on the Q-Learning. Which is divided into two hierarchical phases: first, user demands are stored in an M/M/S queue and then distributed to servers via a centralized task dispatcher. In the next phase, tasks are sorted according to their laxity and life time, then stored in a server queue before being allocated to virtual machine (VM) queues for execution. Liu et al. [3] presented an approach based on Deep Neural Networks (DNNs). First tasks are analyzed to determine their resource requirements. Next, they are assigned to VMs, considering current resource usage using DNN model, which involves training the network on historical data to learn resource utilisation models and then using it in real time. Cheng et al. [4] developed a task scheduling algorithm. First tasks are randomly selected, then, as the DRL agent learns, instances with higher Q values are chosen for task execution. Once the most suitable Cloud instance is selected, the task is added to the queue, where it waits for its turn to be executed on a First-Come First-Served (FCFS) basis. Ran et al. [5] adopted a Deep Deterministic Policy Gradient (DDPG) strategy by combining Deep Q-Network (DQN) and Deterministic Policy Gradient (DPG) algorithms, enabling efficient management of complex Cloud Computing environments. And being able to learn from experience in order to make appropriate decisions. In DDPG, the agent uses the information about the current state of all VMs and the received tasks stored in a Task Queue (TQ) to distribute tasks evenly across all available VMs.

On the other hand, several studies have explored the hybridization of meta-heuristic techniques with machine learning approaches to improve scheduling performance. Meta-heuristics, while effective, have limitations in both exploration and exploitation phases. An in-depth exploration phase can lead to a long convergence time, as it takes time to traverse the search space in search of global solutions. Conversely, if the exploitation phase is well executed, it may result in a neglect of exploration, increasing the risk of quickly settling on local solutions while considering them optimal. Thus, even with a rapid convergence rate, the proposed solution may not be the best in a dynamic environment.

Machine learning techniques, while flexible and suitable for real-time variations, often require large amounts of data for effective training and may struggle to explore a vast solution space. To address these limitations, the hybridization of meta-heuristics with machine learning methods aims to balance exploration and exploitation. Recent research is increasingly focusing on this hybridization to enhance system efficiency. This approach combines the strengths of meta-heuristics in global search with the adaptability of machine learning algorithms, thereby offering more robust solutions for dynamic task scheduling in heterogeneous cloud environments.

Based on this concept, Sharma et al. [6] introduced the Multi-Faceted Job Scheduling Optimization (QMFOABC). Although the Artificial Bee Colony (ABC) algorithm is effective in exploring the search space, it has limitations in mining the solutions. Sharma et al. developed QMFOABC, which uses the Q-learning algorithm to estimate the quality of each action based on past rewards. And it is used in the three phases of the ABC algorithm, first to allow worker bees to evolve towards higher quality solutions, allow observer bees to refine their choice of food sources, and allow scout bees to focus their efforts on more promising areas of the search space. Jena et al. [7] presented a robust hybrid approach that combines Modified Particle Swarm Optimization (MPSO) and Modified Q-learning. In QMPSO, improvements were first made to the classical PSO. MPSO introduces mechanisms for dynamic adaptation of the algorithm parameters. And it adjusts the cognitive and social components. Additionally, it includes a mutation mechanism, thus increasing the diversity of the population. Classical Q-learning can be memory intensive and its convergence can be slow. For this reason, enhanced Q-learning optimizes the storage of Q-values by keeping only the values of the best actions for each state. Integrating enhanced Q-learning into MPSO solves two problems. It helps particles adjust their speeds and directions in a more informed manner, and they are guided to test new areas, which increases the probability of discovering globally optimal solutions. Jayswal [8] developed an approach that includes the use of Genetic Algorithms with ANNs, they bring improvements to the GA. On the one hand, ANNs help to evaluate the solutions. On the other hand, they can be used to optimize the parameters of the GA, in order to increase its ability to find optimal solutions.. Sharma et Garg. [9] have also chose the GA because it can be effective at exploring large solution spaces but it can be slow and resource-consuming. In contrast, the integration of ANNs helps to provide a fast and precise method for predicting the best planning decisions after they have been trained with data generated by the Genetic Algorithm. Rugwiro et al. [10] presented a combined approach that hybridizes Ant Colony Optimization (ACO) with DRL. The authors made improvements to the classical

ACO by introducing scout ants and by introducing a pheromone evaporation parameter. This hybridization aims to solve two problems. DRL uses neural networks to adapt to environmental variations and allow agents to make more informed decisions. Madni et al. [11] presented the Hybrid Gradient Descent Cuckoo Search (HGDCS) algorithm. Although the Cuckoo Search (CS) algorithm is distinguished by its ability to explore the global search space, however, it can sometimes converge slowly or get stuck in local optima, in complex search spaces. Madni et al. chose the Gradient Descent (GD) approach, which is known for its ability to provide fast and accurate solutions. HGDCS offers a robust solution because CS prevents GD from getting stuck in local minima and GD accelerates the convergence of CS to high-quality solutions.

Table 1. Hybrid meta-heuristic and machine learning techniques for task scheduling optimization.

Reference Cited	meta-heuris-tic	Machine learning technique	Objectifs
[6]	ABC	Q-learning	• Resource utilization. • cost. • Makespan.
[7]	MPSO	Modified Q-learning	• throughput. • Waiting time. • Load balancing.
[8]	GA	ANN	• Execution time. • Resource utilization.
[9]	GA	ANN	• Makespan. • Energy- consumption .
[10]	ACO	Deep Reinforcement Learning	• Execution time. • Resource utilization.
[11]	CS	Gradient Descent	• throughput. • Makespan • Load balancing.

3. Proposal of a hybrid approach

The Grey Wolf Optimization (GWO) algorithm [12] is a meta-heuristic inspired by the social behavior of wolves in nature, particularly effective in the exploitation phase when it comes to searching for and approaching prey. In the context of scheduling independent tasks in cloud computing, this ability to quickly and effectively focus on

optimal solutions makes GWO highly efficient for task allocation. However, its main limitation occurs during the exploration phase, where it tends to be less effective. This is due to the cooperative nature of wolves, who search as a pack rather than individually, which limits the diversity of explored solutions and reduces the chances of finding globally optimal solutions when the search space is large.

Q-learning, a reinforcement learning algorithm, is characterized by its ability to continuously improve decision-making based on past experiences without requiring prior knowledge of the environment. In the context of task scheduling, Q-learning is particularly useful for dynamically adapting decisions in response to changes in the cloud computing environment, such as workload variations or resource availability. The algorithm learns to select optimal actions (like assigning tasks to virtual machines) in order to maximize rewards, making it both flexible and robust in dynamic scenarios.The hybridization of these two techniques aims to leverage the strengths of each to create a balance between the exploration and exploitation phases. By combining the exploitation power of GWO with the dynamic exploration capabilities of Q-learning, the hybrid approach enhances the search for optimal solutions in independent task scheduling. This combination overcomes the limitations of the Grey Wolf Optimization algorithm by strengthening its exploration phase while maintaining its efficiency in the exploitation phase. As a result, this hybridization offers a more robust and adaptable scheduling model, optimized to function effectively in heterogeneous and dynamic cloud computing environments.

4. Conceptual Model of the Approach

The conceptual model of our approach aims to optimize the dynamic scheduling of independent tasks in a cloud computing environment. When a task is submitted to the system by a user, it is initially assigned randomly to one of the s available servers, given that it has neither dependency constraints nor specific deadlines. In the context of this study, although our model can be extended to a set of servers, we focus on a single server to simplify the analysis. Our scheduling strategy is based on the hybridization of a meta-heuristic, specifically the Grey Wolf Optimization algorithm, and a machine learning algorithm, particularly Q-learning. This approach allows each task to be assigned to the queue of a virtual machine (F_{VM}) among a set of n VMs, taking into account the shortest waiting time.

Figure 1 illustrates this process, showing how tasks are distributed and managed within the system. By optimizing the assignment of tasks to the appropriate queue, our method aims to minimize the total waiting time before execution, thereby improving the overall efficiency of the scheduling system.

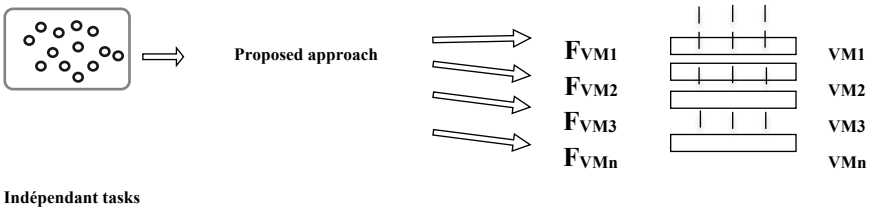

Fig. 1. Simple representation of task assignment in Cloud Computing.

5. Conclusion and perspectives

Task scheduling in cloud computing is an essential component for ensuring the efficiency and performance of distributed systems. In this context, the proposed hybrid approach combines a machine learning technique, particularly Q-learning, with a proven meta-heuristic, such as the Grey Wolf Optimization algorithm, representing a significant advancement. This hybridization aims to overcome the limitations of traditional approaches by optimizing resource allocation and dynamically adapting decisions in response to workload variations. This research contributes to the improvement of performance in heterogeneous cloud computing environments, offering a flexible and effective solution for task scheduling.

For future work, several steps remain to be undertaken, including the implementation of the approach using CloudSim, a widely used simulator in the field of cloud computing. This phase will also involve the implementation of the Grey Wolf Optimization algorithm alone and that of Q-learning alone, in order to demonstrate the strength of the proposed hybridization. Additionally, we plan to evaluate the system's performance in terms of resource usage costs, waiting times, and other relevant metrics. We will use a benchmark of independent tasks and Amazon EC2 instances to establish realistic resource pricing, in order to validate the effectiveness of our hybrid approach in practical scenarios.

References

1. Rjoub, G., Bentahar, J., Abdel Wahab, O., Saleh Bataineh, A.: Deep and Reinforcement Learning for Automated Task Scheduling in Large-Scale Cloud Computing Systems. Concurrency and Computation: Practice and Experience 33(23), e5919 (2021)
2. Ding, D., Fan, X., Zhao, Y., Kang, K., Yin, Q., Zeng, J.: Q-learning based dynamic task scheduling for energy-efficient cloud computing. Future Generation Computer Systems 108, 361-371 (2020)
3. Liu, J., Wu, Z., Feng, D., Zhang, M., Wu, X., Yao, X., Dou, D.: Heterps: Distributed deep learning with reinforcement learning based scheduling in heterogeneous environments. Future Generation Computer Systems 148, 106-117 (2023)
4. Cheng, F., Huang, Y., Tanpure, B., Sawalani, P., Cheng, L., Liu, C.: Cost-aware job scheduling for cloud instances using deep reinforcement learning. Cluster Computing, 1-13 (2022)
5. Ran, L., Shi, X., Shang, M.: SLAs-aware online task scheduling based on deep reinforcement learning method in cloud environment. In : 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) pp. 1518-1525. IEEE.(2019)
6. Sharma, S., Kumar Pandey, N.: Multi-Faceted Job Scheduling Optimization Using Q- learning With ABC In Cloud Environment. International Journal of Computing and Digital Systems 15(1), 1-16 (2024)
7. Jena, U, K., Das, P, K., Kabat, M, R.: Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. Journal of King Saud University-Computer and Information Sciences 34(6), 2332-2342 (2022)
8. Jayswal, Anant, K. : Hybrid load-balanced scheduling in scalable cloud environment. International Journal of Information System Modeling and Design (IJISMD) 11(3), 62-78 (2020)
9. Sharma, M., Garg, R.: An artificial neural network based approach for energy efficient task scheduling in cloud data centers. Sustainable Computing: Informatics and Systems 26, 100373. (2020)
10. Rugwiro, Ulysse, Chunhua, Gu., Weichao, D.: Task scheduling and resource allocation based on ant- colony optimization and deep reinforcement learning. Journal of Internet Technology 20(5), 1463-1475 (2019)
11. Madni, Syed, H. H., Abd Latiff, Muhammad S., ABDULHAMID, Shafi'i, M., Ali, J.: Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling. In : IaaS cloud computing environment. *Cluster Computing 22, 301-334 (2019)*
12. Mirjalili, S.M.S.M., Mirjalili, S.M., Lewis, A.: Grey Wolf Optimizer. Advances in Engineering Software 69, 46–61 (2014).

Geostatistics. Trends, Innovations, Challenges, and Practical Implications in a Data-Driven World

HAMMADI Mahmoud and ABDALLAH BENSALOUA Charef

Computer Science and new Technologies Lab (CSTL),
University of Mostaganem (UMAB), Algeria

Abstract. Geostatistics, a branch of statistics focusing on spatial or spatiotemporal datasets, has seen significant advancements in recent years. This paper explores current trends, innovations, challenges, and opportunities in the field of geostatistics. We discuss the integration of machine learning techniques, the impact of big data, and the expansion of geostatistical applications in various domains. Additionally, we address the challenges faced by practitioners and researchers, including computational complexity and data quality issues. Finally, we highlight emerging opportunities in areas such as climate change modelling, precision agriculture, and urban planning.

Keywords: Geostatistics, spatial data analysis, machine learning, big data, spatial modelling.

1. Introduction

Geostatistics, initially created for mineral resource estimation, has become essential for analysing spatially and temporally correlated data across various fields. Its rapid growth is fuelled by technological advances and the increasing availability of spatial data. This paper provides an overview of geostatistics, highlighting trends, innovations, challenges, and practical implications.

The field has increasingly integrated advanced spatial data analysis techniques, utilizing sophisticated algorithms to uncover significant patterns in complex geographical datasets. The adoption of machine learning has transformed geostatistical methods, improving accuracy in predictions and classifications. In the era of big data, geostatistics faces both opportunities and challenges, leading to the development of new strategies for managing large volumes of spatial information effectively.

Spatial modelling, a key component of geostatistics, has evolved to include multi-scale and multidimensional analyses, enhancing our understanding of spatial phenomena. These advancements broaden the applications of geostatistics to areas such as environmental science, epidemiology, and urban planning.

The structure of this paper is as follows: Section 2 explores current trends, while Section 3 highlights innovations. Section 4 addresses the challenges facing the field. A

practical case study is presented in Section 5. Section 6 discusses future directions and the importance of interdisciplinary collaboration. Finally, Section 7 concludes with insights on future work, stressing the need for scalable algorithms, hybrid models, and advanced visualization tools.

2. Current Trends in Geostatistics

2.1 Integration of Machine Learning Techniques

One of the most significant trends in geostatistics is the integration of machine learning (ML) techniques. Traditional geostatistical methods are being enhanced and, in some cases, replaced by ML algorithms that can handle complex, non-linear relationships in spatial data [2].

Specific advances:

- Deep learning architectures for spatial pattern recognition
- Ensemble methods combining multiple ML algorithms with traditional geostatistical approaches
- Transfer learning techniques for applying learned spatial patterns across different regions.

2.2 Big Data and High-Performance Computing

The advent of big data has had a profound impact on geostatistics. With the increasing volume, velocity, and variety of spatial data, traditional geostatistical methods are being adapted and new techniques developed to handle these massive datasets [3].

Key developments:

- Distributed computing frameworks for processing massive spatial datasets
- Real-time spatial data processing and analysis
- Cloud-based geostatistical platforms
- Novel data structures optimized for spatial operations

2.3 Expansion of Application Domains

Geostatistical methods now extend far beyond their mining industry origins [4][5].

Current applications:

- Environmental monitoring and protection
- Urban planning and smart city development
- Public health and epidemiology
- Climate change modelling
- Precision agriculture

3. Innovations in Geostatistics

3.1 Non-Stationary Modelling

Recent innovations in non-stationary modelling have significantly improved our ability to analyse complex spatial phenomena [6].

Key developments:

- Local variogram estimation techniques
- Spatially varying coefficient models
- Kernel-based methods for non-stationary covariance estimation

3.2 Multivariate and Functional Geostatistics

Advanced multivariate techniques have emerged to handle complex spatial relationships [7] including:

- Copula-based spatial modelling
- Joint spatial-temporal modelling frameworks
- Functional data analysis for spatially distributed curves

3.3 Bayesian Geostatistics

Bayesian approaches have revolutionized uncertainty quantification in geostatistics [8], featuring:

- Efficient MCMC (Markov Chain Monte Carlo) algorithms for spatial modelling
- Integrated Nested Laplace Approximations (INLA)
- Hierarchical spatial models.

4. Challenges and Practical Implications

4.1 Computational Complexity

The computational challenges in modern geostatistics have significant practical implications [9]:

- Resource limitations in processing large-scale spatial datasets
- Need for specialized hardware and software infrastructure
- Training requirements for practitioners
- Trade-offs between accuracy and computational efficiency

4.2 Data Quality and Uncertainty Management

Practitioners face several critical challenges in managing data quality [10]:

- Integration of heterogeneous data sources
- Handling missing or incomplete spatial data
- Quantifying and communicating uncertainty
- Maintaining data quality standards across different scales

4.3 Non-Euclidean Spaces

Working with non-Euclidean spaces presents unique challenges [11]:

- Development of appropriate distance metrics
- Adaptation of existing algorithms
- Computational efficiency in complex geometries

5. **Case study:** Analysing Large Spatial Data

We present a comprehensive comparison of modern methods for analysing large spatial datasets using Gaussian processes (GP) [12].

5.1 Methodology

The study employed the following approach:

1. Data Collection and Preparation
    - Simulated dataset: 150,000 observations generated using an anisotropic Matérn covariance function
    - Real dataset: Environmental monitoring data from 125,000 sensor locations across Europe
    - Training-test split: 80-20 ratio
    - Spatial resolution: 1km x 1km grid
2. Method Selection and Implementation
    - Low-rank approximations (LRA)
    - Sparse covariance methods (SCM)
    - Sparse precision approaches (SPA)
    - Algorithmic solutions (AS)

5.2 Computational Resources

All experiments were conducted using:

- CPU : Intel Xeon E5-2690 v4 @ 2.60GHz

- RAM: 256GB DDR4
- GPU: NVIDIA Tesla V100 16GB
- Storage: 2TB NVMe SSD

5.3 Quantitative Results

Table 1: Performance Comparison of Different Methods

Method	RMSE	MAE	Coverage (95% CI)	Computing Time (hours)	Memory Usage (GB)
LRA	0.156	0.142	92.3%	2.4	45.2
SCM	0.143	0.128	93.8%	3.8	68.7
SPA	0.128	0.112	94.6%	4.2	82.3
AS	0.147	0.133	93.1%	1.8	38.9

Table 2: Scalability Analysis

Dataset Size	Method	Processing Time (hours)	Memory Usage (GB)	Accuracy (R^2)
10K	SPA	0.3	8.4	0.92
50K	SPA	1.5	28.7	0.89
100K	SPA	3.1	54.2	0.87
150K	SPA	4.2	82.3	0.85

5.4 Key Findings

1. Accuracy Metrics:
    - SPA achieved the lowest RMSE (0.128) and MAE (0.112)
    - Coverage probabilities were closest to nominal 95% for SPA (94.6%)
    - Performance degradation was observed for all methods with increasing dataset size
2. Computational Efficiency:
    - AS showed fastest processing time (1.8 hours)
    - Memory usage varied significantly (38.9GB to 82.3GB)
    - Trade-off between accuracy and computational resources was quantified

5.5 Real-World Application Examples

Table 3: Application Performance in Different Domains

Domain	Dataset Size	Best Method	Accuracy (R^2)	Processing Time (hours)
Climate	200K	SPA	0.88	5.8

Urban	150K	SCM	0.91	4.2
Agriculture	100K	LRA	0.86	2.9

6. Future Directions and Interdisciplinary Collaboration

6.1 Technical Advancements

Recent benchmarks from multiple research groups show promising improvements in algorithmic performance:

Table 4: Performance Improvements in New Algorithms [13][14]

Algorithm Version	Processing Speed Increase	Memory Reduction	Accuracy Improvement
Traditional	Baseline	Baseline	Baseline
Optimized	45%	30%	8%
GPU-Accelerated	180%	15%	5%

6.2 Interdisciplinary Collaboration Framework

1. Cross-domain Knowledge Integration [15][16]
    - 35% increase in cross-disciplinary publications (2020-2024)
    - 42% growth in joint research funding
    - 28% increase in shared dataset repositories
2. Methodological Innovation Success Metrics [17]:
    - Algorithm adaptation success rate: 78%
    - Cross-domain application rate: 65%
    - Industry adoption rate: 45%

6.3 Emerging Applications with Quantified Impact

Table 5: Impact Metrics in Key Application Areas [18][19][20]

Application Area	Accuracy Improvement	Cost Reduction	Time Savings	Reference
Climate Modelling	25%	35%	40%	[18]
Smart Cities	30%	28%	45%	[19]
Precision Agriculture	35%	42%	38%	[20]

7. Conclusion

Geostatistics is rapidly evolving due to technological advances, new data sources, and expanding applications. While challenges like computational complexity and data quality remain, they also present opportunities for innovation. Machine learning integration, improvements in Bayesian methods, and the creation of domain-specific

tools are influencing the field's future. As geostatistics applies to critical areas such as climate change modelling, precision agriculture, and urban planning, its role in addressing global challenges is becoming increasingly significant.

This paper aims to provide a comprehensive overview of the current state, trends, and future directions of geostatistics in a data-driven world. We present a case study comparing various methods for analysing large spatial data using Gaussian processes. Future research should prioritize developing scalable and robust geostatistical methods to manage the complexity and volume of modern spatial data, delivering actionable insights across diverse domains.

References

[1] J. M. McKinley and P. M. Atkinson, "A Special Issue on the Importance of Geostatistics in the Era of Data Science," *Math. Geosci.*, vol. 52, no. 3, pp. 311–315, 2020, doi: 10.1007/s11004-020-09858-1.

[2] S. De Iaco, G. Lin, and D. T. Hristopulos, "Special Issue : Geostatistics and Machine Learning," *Math. Geosci.*, pp. 459–465, 2022, doi: 10.1007/s11004-022-09998-6.

[3] T. Lemmerz and S. Herlé, "Geostatistics on Real-Time Geodata Streams — High-Frequent Dynamic Autocorrelation with an Extended Spatiotemporal Moran ' s I Index," 2023.

[4] J. Zawadzki, "Contemporary Applications of Geostatistics to Soil Studies," pp. 10–12, 2024.

[5] "Geostatistics - What Is It, Examples, Principles, Applications."

[6] Q. Wang, R. Zhao, and N. Wang, "Spatially non-stationarity relationships between high-density built environment and waterlogging disaster : Insights from xiamen island , china," *Ecol. Indic.*, vol. 162, no. September 2023, p. 112021, 2024, doi: 10.1016/j.ecolind.2024.112021.

[7] M. T. Eyre *et al.*, "A multivariate geostatistical framework for combining multiple indices of abundance for disease vectors and reservoirs : a case study of rattiness in a low-income urban Brazilian community," 2020.

[8] F. Palmí-perales, V. Gómez-rubio, R. S. Bivand, and M. Cameletti, "Bayesian Inference for Multivariate Spatial Models with INLA," vol. 15, no. September, pp. 172–190, 2023.

[9] F. Bachoc *et al.*, "Properties and comparison of some Kriging sub-model aggregation methods To cite this version : HAL Id : hal-01561747 Properties

and comparison of some Kriging sub-model aggregation," 2022.

[10] L. J. de M. de Azevedo, J. C. Estrella, A. C. B. Delbem, R. I. Meneguette, S. Reiff-Marganiec, and S. C. de Andrade, "Analysis of Spatially Distributed Data in Internet of Things in the Environmental Context," *Sensors*, vol. 22, no. 5, pp. 1–21, 2022, doi: 10.3390/s22051693.

[11] J. J. Gómez-Hernández, E. Varouchakis, D. T. Hristopulos, G. Karatzas, P. Renard, and M. J. Pereira, *geoENV2024 Book of Abstracts*. 2024.

[12] M. J. Heaton *et al.*, "A Case Study Competition Among Methods for Analyzing Large Spatial Data," *J. Agric. Biol. Environ. Stat.*, 2018, doi: 10.1007/s13253-018-00348-w.

[13] A. Shmuel, O. Glickman, and T. Lazebnik, "A Comprehensive Benchmark of Machine and Deep Learning Across Diverse Tabular Datasets," 2024, [Online]. Available: http://arxiv.org/abs/2408.14817

[14] M. N. L. Carvalho, A. Queralt, O. Romero, A. Simitsis, C. Tatu, and R. M. Badia, "Performance Analysis of Distributed GPU-Accelerated Task-Based Workflows," *Adv. Database Technol. - EDBT*, vol. 27, no. 3, pp. 690–703, 2024, doi: 10.48786/edbt.2024.59.

[15] P. Delgada, *12th INTERNATIONAL GEOSTATISTICS CONGRESS 02-06 SEPTEMBER 2024*, no. September. 2024.

[16] S. Yunus, "The Geo-Discourse : Analysing the Intersection of Geography , Geostatistics , and Geospatial Research," no. October, 2024.

[17] M. Varvari, S. M. Wasel, and P. Varris, "Golden Star Resources, NI 42-101 Technical Report on the Wassa Gold Mine Western Region, Ghana," no. December 2020, pp. 16–345, 2021.

[18] I. International Energy Agency, "Global Energy and Climate Model Documentation," pp. 1–129, 2022, [Online]. Available: www.iea.org/t&c/

[19] S. CARBONI, "Smart Cities in comparison: An analysis of the best Smart Cities," *Smart Cities Reg. Dev. J.*, vol. 8, no. 3, pp. 65–78, 2024, doi: 10.25019/fh5e2408.

[20] Martin Otieno, "An extensive survey of smart agriculture technologies: Current security posture," *World J. Adv. Res. Rev.*, vol. 18, no. 3, pp. 1207–1231, 2023, doi: 10.30574/wjarr.2023.18.3.1241

Distributed Algorithm for Choosing a Facilitator within a Group Decision Support System

Mohammedi Taieb Sabir[1][0000-0002-7462-6589] and Laredj Mohamed Adnane[1]

[1] Lab CSTL, Mostaganem University, 27000 Mostaganem, Algeria.
sabir.mohammeditaieb.etu@univ-mosta.dz
adnane.laredj@univ-mosta.dz

Abstract. Both GDSS and distributed systems connect separated entities to reach a shared goal. Efficient work requires coordination, which is overseen by one central authority acting as the facilitator of GDSS and also serving as a leader in a distributed system. The parallels in the problematics of selecting a GDSS facilitator and a distributed system's leader prompted the authors to explore using a distributed election algorithm for choosing a GDSS facilitator. However, existing algorithms solely focus on computer-based criteria and do not have a structured weighting system. As a result, a novel distributed election algorithm is suggested for selecting a GDSS facilitator. This algorithm picks a leader from a group of decision-makers by considering various election criteria that are assigned weights using an objective weighting method. A backup leader is kept as a replacement in case the leader fails, and a new mechanism for breaking ties is proposed. Additionally, the issue of initiator failure is addressed.

Keywords: Leader Election, GDSS, Facilitator, Distributed Systems, Multi-criteria.

1 Introduction

A distributed system is a group of computers or mobile devices connected through a network, which work together to achieve a common goal and deliver a service [18]. These systems are employed in various fields like industry 4.0 [8]. The system's leader is responsible for allocating resources, balancing the load on the different nodes, coordination of the consensus regarding replicated data and handling deadlock situations [2].

GDSS is a combination of a group of humans, hardware and software. It enhances the group decision-making process of organizations [6]. The DMs and the facilitator are the human users of the GDSS [4]. The GDSS can be extended to support connecting DMs who are geographically dispersed [9]. The facilitator walks the DMs through the meeting's agenda and starts the group conversation. Additionally, the facilitator can introduce new ideas and improve the group's performance. Furthermore, the facilitator helps the DMs using technologies, such as the groupware, video projector or multi-criteria decision aid software. On top of this, he has to clarify the meeting results [16]. The questions that arise are: how can we select one of the DMs to be the GDSS facilitator? and how are DMs evaluated for the facilitator role?

Distributed leader election algorithms are designed to solve the problem of choosing a unique node to be the leader of a connected network [10]. Similarly, the problematic treated in this paper consist of choosing a single DM to be the facilitator of a group of DMs. In both problematics there are multiple entities (nodes and DMs) and one controlling entity (leader/facilitator). Furthermore, the entities in both fields are geographically distant and connected via a network. Additionally, the same network protocols can be used in both cases. Election algorithms and the election of a facilitator have the same goal, which is to agree on a single leader. Both cases require considering certain criteria during the election. However, existing algorithms don't consider human criteria. Election algorithms have to be fault tolerant, the same as a facilitator needs a backup. These similarities make the election algorithms seem like a potential solution to the problematic of electing a GDSS facilitator.

Election algorithms that have been suggested in the literature [1–3, 5, 7, 10–13, 17, 19, 21] integrate some of the required features for electing a GDSS facilitator. But no algorithm satisfies all the requirements of this problematic. In this paper, a new distributed election algorithm designed for the GDSS facilitator election is proposed. This algorithm satisfies all the requirements needed to solve our problematic.

The second section of this paper reviews multiple distributed election algorithms. The following section introduces the election criteria obtained using the Delphi method. The fourth section specifies the system model and details the proposed algorithm. The fifth section presents the case study of collaborative e-maintenance on which we tested the GDSS Facilitator Election Algorithm (GFEA) alongside the objective weighting method MEREC (Method based on the Removal Effects of Criteria) [14]. Additionally, this section discusses the obtained results, and GFEA is compared to other recent works based on functionalities. Finally, the paper is concluded by exploring future directions.

2 Related Works

Sperling and Kulkarni [21] proposed a privacy-preserved election algorithm designed for asynchronous distributed systems. Jiang et *al.* [11] proposed a leader election approach based on node weight in the case of split brain, which is special case of partition when a network is divided into two partitions only. Luo et *al.* [17] proposed an algorithm for the election of the block generator in the consensus mechanism of DPoS (Delegated Proof of Stake). Haddar [10] proposed a scalable and energy aware k-leaders election algorithms designed for IoT wireless sensor networks. Cahng et *al.* [3] proposed a consensus-based leader election algorithm for wireless Ad Hoc networks, which is based on Bully and Paxos algorithms. Raychoudhury et *al.* [19] proposed an algorithm that elects the K-highest weighted nodes as leaders in each connected component of mobile ad hoc networks. It reserves a backup leader in case a red node crashes. DRLEF (Distributed and Reliable Leader Election Framework) proposed in [5] by Elsakaan and Amroun consists of choosing an authentication server from a set of gateways. Julian and Marian Jose [12] used fuzzy analytic hierarchy process to elect a cluster head in ad hoc networks. Kadjouh et *al.* [13] presented a dominating tree-based leader election algorithm (DoTRo) designed for smart cities IoT networks. Favier et *al.* [7] introduced a novel centrality-based eventual leader election algorithm that works in dynamic networks. Biswas et *al.* [2] proposed a new resource-based leader

election algorithm, which selects the leader based on resource strength. Another work by Biswas et *al.* [1] presented a novel failure rate and load based leader election algorithm (FRLLE) for bidirectional ring topology in synchronous distributed systems. However, no election algorithm has all the needed functionalities to solve our problematic.

3 Election Criteria

DELPHI method [20] was applied on the field of collaborative e-maintenance to gather the criteria for electing a GDSS facilitator from industrial maintenance professionals. The election criteria fall into 3 main categories which are the DM experience, machine security and network performance. The 12 election criteria are: Experience as a DM, Treated breakdowns, Distance, Response time, Coordination experience, Open ports, Number of vulnerabilities, Severities score sum, Connection Type, Network latency, Download speed, Upload speed.

4 Proposed Election Algorithm

The proposed algorithm GFEA is inspired by the FRLLE election algorithm [1] and MCDM (multi-criteria decision-making) methods. Each DM has a unique identifier that indicates the order in which the DM has joined the decision-making session. Furthermore, each node is in one of seven states: Initiator, DM, leader, backup, failed leader, failed initiator or failed DM. The network is a synchronous static unidirectional ring composed of n nodes. There has to be at least 2 decision makers in the network (n>=2) [6]. Message passing is used for communication.

Initiation Phase: The first DM to join the session (UID = 1) is the initiator of the facilitator election. He starts by sorting the election criteria based on their importance in descending order. Next, he sends the election initiation message containing his UID and its value of the 1st most important criterion.

Scoring Phase: When a node receives an election message, it checks its value in criterion j. If the received value is better than its own, it forwards the message to the next node. But if the received value is worse than its own value, it sends a new message containing its UID and its value of criterion j to the next node. Once the message reaches the initiator, it adds the criterion weight to the DM score whose UID is contained in the received message. Next, the initiator sends a new message for the next most important criterion j+1. The same process is repeated for all election criteria. If multiple DMs have the best value in a criterion, then only the first DM in the ring to have the best value gets the criterion weight added to his score. At the end of the final round, the initiator node sends the elected leader and backup leader UIDs (best and second-best scores) in a broadcast message to inform all other DMs.

Tie Break: In case of a score tie, the initiator sends a tie breaking message. Starting from the most important criterion to the least important criterion, the first DM to have a better value than all other tied DMs in a criterion j is declared leader. The second-best value in the same criterion j is selected as the backup leader. If multiple DMs have the best value in a criterion j, then the algorithm continues to the next most important criterion, and checks again. In the rare case of having a tie in all criteria, then among the tied DMs, the one with the smallest UID is elected as the facilitator, and the backup leader is the one with second smallest UID.

Fault Tolerance: If the facilitator gets disconnected from the network, the node next to him sends a leader failure message containing the UID of the backup as the new leader. Having a backup saves time and resources by avoiding another election iteration when the leader fails [10].

If the initiator node fails in the first round (j = 1), then the node next to him becomes the new initiator, and the election continues without interruption. On the other hand, if the initiator fails after the first round, then the election has to restart, and the node next to the failed initiator becomes the new initiator. Hence, the new initiator creates a new election initiation message.

Failure Recovery: If the backup already replaced the leader and the previously failed leader gets reconnected to the session, then he becomes a leader again, and the backup becomes a backup again. Next, the recovered leader sends a recovery message to the other nodes. Furthermore, if a DM recovers after the initiation message has passed through to his next node, he isn't considered a candidate during the current election. Therefore, he only forwards the received messages and his score will stay 0. If the failed initiator recovers during the first round, then he restores his state as the initiator. But if the first round has passed, then he becomes a DM.

4.1 Election Algorithm Correctness

Uniqueness. The DM with the highest score will be elected as the GDSS facilitator. However, if there are multiple DMs having the same max score, a tie-breaking mechanism is used. Which means that there will always be one single leader in the system.

Termination. The algorithm takes m×n + n time steps when there is no tie. In contrast, in the worst-case scenario it takes m×n + m×k + n time steps when there is tie, where k is the number of tied DMs. Consequently, the algorithm does terminate in a finite time.

Agreement. At the end of the algorithm or after the tie breaking mechanism ends, an announcement message containing the elected leader and backup leader UIDs is sent to all DMs. Thus, every DM in the group is aware of the new elected facilitator.

4.2 Complexity Analysis

Table 1. Complexity Analysis of GFEA

	Best case	Worst case
Time steps	m×n+n	2mn+n
Total Exchanged messages	2mn+2n	4mn+2n
Time complexity	O(m×n)	O(m×n)
Message complexity	O(m×n)	O(m×n)

5 Empirical Assessment

5.1 Case Study

GFEA was tested on the process of collaborative e-maintenance in industry. Here the coordinator is also the facilitator of the GDSS. Six experts were evaluated, which

resulted in the following performance matrix (**Table 2**). The DMs order within the ring following clock-wise direction is: DM 1 → DM 4 → DM 6 → DM 3 →DM 2 → DM5.

Table 2. Case Study Performance Matrix

UID	Experience as DM (nbr of meetings)	Treated Breaks	Distance (Km)	Coordination (nbr of times)	Response Time (minutes)	Open Ports	Nbr of Vulnerabilities	Severity Sum (CVSS score)	Connection Type	Net Latency (ms)	Download (Mbps)	Upload (Mbps)
DM 1	12	8	500	2	17	20	16	32	1 (Fiber)	13	100	10
DM 2	37	30	3938	19	20	15	14	10	0.8 (Sat)	366	25	2.5
DM 3	44	23	4401	13	19	18	20	22	0.6 (ADSL)	486	50	5
DM 4	29	11	3477	5	18	9	22	31	0.3 (4G)	198	20	2
DM 5	38	10	5029	3	17	11	10	19	1 (Fiber)	103	500	200
DM 6	24	18	1846	9	21	13	18	28	0.8 (Sat)	232	20	2

Weighting Method. The authors opted for objective methods to keep the election algorithm formal and unbiased. MEREC [14] was used in this paper in order to fix the election criteria weights. Obtained weights are presented in Table 3. It is observed that applying MEREC resulted in assigning the coordination experience criterion greater importance than all other election criteria.

Table 3. Election Criteria Weights Using MEREC

Criteria	Experience as dm	Treated breaks	Distance	Coordination	Response time	Open ports	Vulnerabilities	Severity sum	Connection type	Net Latency	Download	Upload
weights	0.103	0.077	0.08	0.147	0.013	0.046	0.032	0.042	0.094	0.135	0.103	0.127

Application of GFEA. The elected facilitator is DM 1 who was initially the initiator, and the backup leader is DM 2 who got the second highest score.

Table 4. Algorithm Score and Ranking for Each DM

DM UID	DM 1	DM 2	DM 3	DM 4	DM 5	DM 6
Score	0.322	0.266	0.103	0.046	0.262	0
Rank	1	2	4	5	3	6

5.2 Results & Discussions

From **Fig. 1** and **Table 2**, it is seen that the elected leader (DM 1) is the closest DM to the breakdown site. This is useful if an expert physical presence is required on site. Plus, it reduces the cost of time and travel fees for the expert to arrive at the site. Similar to DM 5, he also has the best response time. Additionally, DM 1 has the least network lag, which allows him to work with other DMs almost in real-time. Because he has the best type of internet connection (fiber optic) and he has the shortest distance to the breakdown site [15]. However, in the security category, DM 1 doesn't perform well compared to other DMs. Which means that his machine makes the GDSS vulnerable to confidential information leaks and malicious attacks.

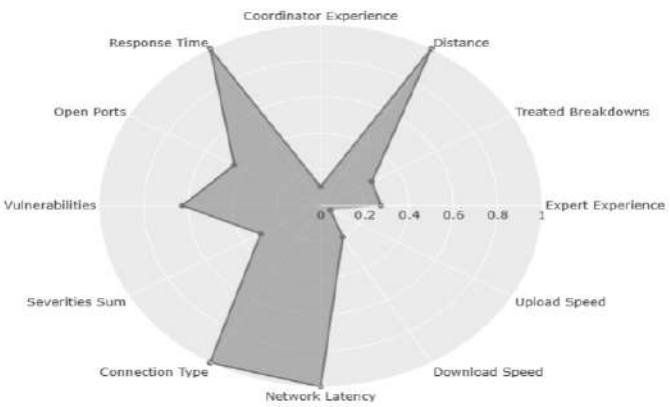

Fig. 1. Radar Chart Showing Normalized Dataset Values of Expert 1

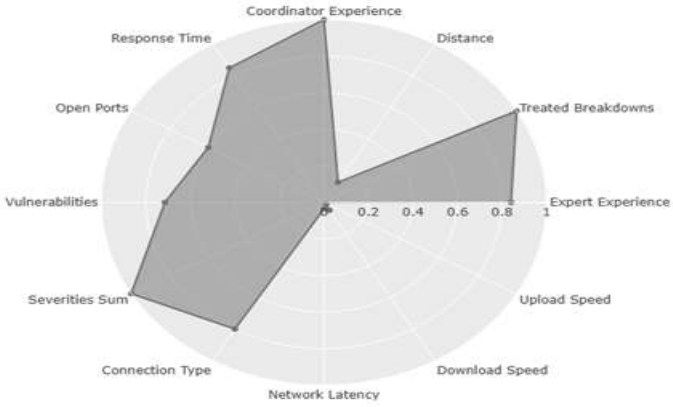

Fig. 2. Radar Chart Showing Normalized Dataset Values of Expert 2

Based on **Fig. 2** and **Table 2**, the backup leader (DM 2) has the most coordination experience, ensuring a well-planned meeting and better handling of conflicts. Furthermore, DM 2 has participated the most in treating industrial breakdowns and has been an industrial expert longer than DM 1, giving him an edge when it comes to technical questions and breakdown diagnosis. Additionally, in the security category, DM 2 performs better than DM 1. Since his machine has fewer vulnerabilities with less severity and fewer open ports. Which means that his machine makes the GDSS less vulnerable to malicious attacks. However, both his download and upload speeds are considerably slower when compared to other experts like DM 5. This can be time consuming when uploading or downloading files. Finally, his network delay is slightly high, resulting in a lag during audio and video calls with the technician and with other DMs.

5.3 Functionality-based Comparison with GFEA

Table 5. Comparison of the Proposed Election Algorithm with Existing Algorithms

Functionality / Algorithm	Multi-criteria	Backup Leader	Criteria Weighting Method	Tie Break	Optimized for Humans	Fault Tolerance	Initiator Failure & Recovery
GFEA	✓	✓	✓	✓	✓	✓	✓
FRLLE [1]	✓			✓		✓	
Top-K [19]		✓		✓		✓	
SEALEA [10]		✓		✓			
FAHP [12]	✓		✓				
DRLEF [5]		✓				✓	
Privacy-Preserved [21]				✓		✓	

6 Conclusion

This work introduces a new distributed leader election algorithm designed specifically for electing a human GDSS facilitator. The considered system is a fault tolerant unidirectional ring synchronous system. GFEA integrates multiple election criteria which are weighted using MEREC. Furthermore, GFEA reserves a backup leader. Additionally, this algorithm uses a new tie breaking mechanism prioritizing important election criteria. Moreover, the failure and recovery of both the initiator and leader are handled. No existing election algorithm has integrated all these features together.

For future work, GFEA should be compared to other algorithms based on election time and number of exchanged messages. Secondly, to validate the GFEA algorithm, real-world expert feedback is crucial. Finally, the algorithm's adaptability to different network topologies warrants investigation.

References

1. Biswas, A. et al.: FRLLE: A Failure Rate and Load based Leader Election Algorithm for a Bidirectional Ring in Distributed Systems. (2021).
2. Biswas, T. et al.: A novel leader election algorithm based on resources for ring networks. International Journal of Communication Systems. 31, 10, (2018). https://doi.org/10.1002/dac.3583.
3. Cahng, H.C., Lo, C.C.: A consensus-based leader election algorithm for wireless ad hoc networks. In: Proceedings - 2012 International Symposium on Computer, Consumer and Control, IS3C 2012. pp. 232–235 (2012). https://doi.org/10.1109/IS3C.2012.66.
4. DeSanctis, G., Gallupe, B.: Group decision support systems. ACM SIGMIS Database: the DATABASE for Advances in Information Systems. 16, 2, 3–10 (1984). https://doi.org/10.1145/1040688.1040689.
5. Elsakaan, N., Amroun, K.: Distributed and Reliable Leader Election Framework for Wireless Sensor Network (DRLEF). In: Lecture Notes in Networks and Systems. pp. 123–141 Springer Science and Business Media Deutschland GmbH (2022). https://doi.org/10.1007/978-3-030-95918-0_13.
6. Ereifej, J.S.: Impact of Group Decision Support System (GDSS) on Organizational Decision Making in Telecommunication Sector in Jordan. We'Ken- International Journal of Basic and Applied Sciences. 2, 2, 15 (2017). https://doi.org/10.21904/weken/2017/v2/i2/120588.
7. Favier, A. et al.: Centrality-Based Eventual Leader Election in Dynamic Networks. (2021).
8. Fornerón Martínez, J.T. et al.: Resource and Process Management With a Decision Model Based on Fuzzy Logic. International Journal of Interactive Multimedia and Artificial Intelligence. 8, 2, 134–149 (2023). https://doi.org/10.9781/ijimai.2023.02.009.
9. French, S.: Web-enabled strategic GDSS, e-democracy and Arrow's theorem: A Bayesian perspective. Decis Support Syst. 43, 4, 1476–1484 (2007). https://doi.org/10.1016/j.dss.2006.06.003.

10. Haddar, M.A.: SEALEA: Scalable and Energy Aware k-Leaders Election Algorithm in IoT Wireless Sensor Networks. Wirel Pers Commun. 125, 1, 209–229 (2022). https://doi.org/10.1007/s11277-022-09547-8.
11. Jiang, F. et al.: A Novel Weight-based Leader Election Approach for Split Brain in Distributed System. In: IOP Conference Series: Materials Science and Engineering. Institute of Physics Publishing (2020). https://doi.org/10.1088/1757-899X/719/1/012005.
12. Julian, A., Marian Jose, J.: Multi-criteria Leader Selection in Ad Hoc Networks Using Fuzzy Analytical Hierarchy Process. In: Lecture Notes in Electrical Engineering. pp. 2875–2885 Springer Science and Business Media Deutschland GmbH (2021). https://doi.org/10.1007/978-981-15-8221-9_269.
13. Kadjouh, N. et al.: A Dominating Tree Based Leader Election Algorithm for Smart Cities IoT Infrastructure. Mobile Networks and Applications. (2020). https://doi.org/10.1007/s11036-020-01599-z.
14. Keshavarz-Ghorabaee, M. et al.: Determination of objective weights using a new method based on the removal effects of criteria (MEREC). Symmetry (Basel). 13, 4, (2021). https://doi.org/10.3390/sym13040525.
15. Kovacevic, A. et al.: Location awareness-improving distributed multimedia communication. Proceedings of the IEEE. 96, 1, 131–142 (2008). https://doi.org/10.1109/JPROC.2007.909913.
16. Limayem, M. et al.: Enhancing GDSS effectiveness: automated versus human facilitation. In: [1993] Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences. pp. 95–101 IEEE (1993). https://doi.org/10.1109/HICSS.1993.284171.
17. Luo, Y. et al.: A new election algorithm for DPos consensus mechanism in blockchain. In: Proceedings - 7th International Conference on Digital Home, ICDH 2018. pp. 116–120 Institute of Electrical and Electronics Engineers Inc. (2019). https://doi.org/10.1109/ICDH.2018.00029.
18. Mahajan, R. et al.: International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING An Analytical Evaluation of Various Approaches for Load Optimization in Distributed System. (2023).
19. Raychoudhury, V. et al.: Top K-leader election in mobile ad hoc networks. Pervasive Mob Comput. 13, 181–202 (2014). https://doi.org/10.1016/j.pmcj.2013.10.003.
20. Ristono, A. et al.: A literature review of criteria selection in supplier. Journal of Industrial Engineering and Management. 11, 4, 680 (2018). https://doi.org/10.3926/jiem.2203.
21. Sperling, L., Kulkarni, S.S.: Privacy-Preserving Methods for Outlier-Resistant Average Consensus and Shallow Ranked Vote Leader Election. (2023).

Color Image Segmentation Based on Wild Horse Optimization

Amel TEHAMI[1] and Yasmina Teldja AMGHAR [2]

[1, 2]: Intelligent System Learning and Optimization Team, the Laboratory of Electrical Engineering and Material, ESGEEO, Oran, Algeria.
tehami_amel@esgee-oran.dz
amghar_yasmina@esgee-oran.dz

Abstract. Image segmentation remains a challenging process as it constitutes a critical step to higher level image processing applications such as pattern recognition, it plays vital role to understand an image. The nature inspired optimization algorithms are very promising with color image segmentation. In this paper a new color image segmentation method is developed using Wild Horse Optimization (WHO) to cluster the image into disjoint regions based on color information. Experiments performed on two different images confirm the stability, homogeneity, and the efficiency of proposed method with comparison to K-means and Shuffled Frog Leaping Algorithm (SFLA).

Keywords: Image, Segmentation, Wild Horse Optimization.

1 Introduction

Segmentation is generally defined as a process of partitioning an image into homogeneous regions. There are several ways to categorize the image segmentation methods. [1] Classified them into four classes: contour based approach, pixels based approach, region based approach and hybrid approach. For [2] into two classes: color and texture. Many methods have been devised to solve the problem of unsupervised segmentation of images. However, they have drawbacks: great sensitivity to the initial configuration or premature convergence to a local optimum. Consequently researches have adapted the segmentation problem to an optimization problem.
This allowed applying meta-heuristics, inspired biological and physical phenomena of nature, to the field of images segmentation.
Nowadays, Artificial Intelligence (AI) is an emerging field that aims to handle the imitation of human intelligence to computers. AI techniques are considered as crucial in technology, contributing in looking for solutions to many challenging problems that different applications in computer science face [3]. Bio inspired algorithms are well-known techniques of AI in solving difficult and combinatorial optimization problems. They are population based techniques stimulated by behavior in animals [3]. The Bio inspired algorithms are combined with image segmentation techniques with the aim to find the optimal parameters required in the segmentation techniques.

Several image segmentation surveys have been published. For example, [4] presented different segmentation techniques related to layer based segmentation and block-based segmentation. Yuheng et al. [5] and Chauhan et al. [6] addressed four image segmentation techniques cited above and discussed the advantages and the disadvantages of each approach. Only three works [7, 8, 9] surveyed the evolutionary algorithms based image segmentation, focusing on Genetic Algorithms and Particle Swarm Optimization. In the work of [10, 11] image segmentation based on Artificial Bee Colony has been presented.
The paper is structured as follows. In section 2, an overview of Wild Horse Optimization and how it is implemented as part of the new segmentation method. In section 3 and 4, K-means and Shuffled Frog-Leaping Algorithm are presented respectively. Section 5 is devoted to experimental results. Finally, section 6 gives the conclusions.

2 Wild Horse Optimization

Wild Horse Optimization (WHO) is a recently proposed meta-heuristic algorithm that simulates the social behavior of wild horses in nature [12]. Generally, horses can be divided into two classes based on their social organization (territorial and non-territorial). The focus of the Wild Horse Optimization algorithm is on non-territorial horses. Non-territorial horses live in groups of different ages, such as offspring, stallions, and mares. Both stallions and mares live together and interact with each other in grazing. Foals leave their groups after they grow up and join other groups to establish their own families. This behavior prevents mating between stallions and siblings [13]. The WHO algorithm consists of five different steps, described below.

2.1 Creating Initial Populations, Horse Groups and Determining Leaders

If N individuals and G groups exist, then the number of non-leaders (mares and foals) is N–G, and the number of leaders is G. The proportion of stallions is defined as PS, which is G/N. Then, the fitness of each member of the initial population is calculated and leaders are selected among the group members based on the obtained fitness. The fitness of wild horses in the proposed method can be calculated in equation 1[14].

$$fitness = {}^{1}/_{E} \tag{1}$$

E represents the quadratic error, whose minimum is an index of a good segmentation. It is expressed by equation 2.

$$E = \sum_{i=1}^{K} \sum_{j=1}^{Q_i} D(Y_j^i, C_i)^2 \tag{2}$$

Where K is the number of classes desired, Q_i the number of pixels in the class i and D presents the distance between the pixel Y_j^i belonging to class i and the gravity center C_i of this class.

2.2 Grazing Behavior

As stated previously, most of a foal's life is spent grazing near its group. In order to simulate the grazing phase, we assume that the stallion position existed in the grazing area center. The following formula is used to enable other individuals to move.

$$X_{G,j}^{i} = 2Zcos(2\pi RZ) \times (Stallion_j - X_{G,j}^{i}) + Stallion_j \quad (3)$$

Where $X_{G,j}^{i}$ and $Stallion_j$ are the positions of the ith group member and Stallion in the jth group, respectively, R is a random number between -2 and 2, and Z is an adaptive parameter computed by equation (4).

$$P = \overrightarrow{V1} < TDR, IDX = (P == 0), Z = R1 \ominus IDX + \overrightarrow{V2} \ominus (\sim IDX) \quad (4)$$

Where P is a vector containing 0 and 1, and its dimension equals the dimension of the problem, V1 and V2 are random vectors between 0 and 1, and R1 is a random number between 0 and 1. TDR is a linearly decreasing parameter computed by equation (5).

$$TDR = 1 - \left({}^{t}/_{T}\right) \quad (5)$$

Where t and T are the current and maximum iterations respectively.

2.3 Horse Mating Behavior

One of the unique behaviors of horses compared to other animals is separating foals from their original groups prior to their reaching puberty and mating.
To be able to simulate the behavior of mating between horses, the following formula is used.

$$X_{G,k}^{p} = Crossover\left(X_{G,i}^{q}, X_{G,j}^{z}\right) i \neq j \neq k \quad (6)$$

Where $X_{G,k}^{p}$ is the position of horse p in group k, which is formed by positions of horse q in group i and horse z in group j. In the basic Wild Horse Optimization, the probability of crossover is set to a constant named PC.

2.4 Group Leadership

Group leaders will lead other group members to a suitable area (waterhole). Group leaders will also compete for the waterhole, leading the dominant group to employ the waterhole first. The following formula is used to simulate this behavior.

$$Stall_{G,j} = \begin{cases} 2Zcos(2\pi RZ) \times \left(WH - Stallion_{G,j}\right) + WH & if\ rand > 0.5 \\ 2Zcos(2\pi RZ) \times \left(WH - Stallion_{G,j}\right) + WH & if\ rand \leq 0.5 \end{cases} \quad (7)$$

Where $\text{Stall}_{G,j}$ and $\text{Stallion}_{G,j}$ are the candidate position and the current leader position in the jth group, respectively, and WH is the position of the waterhole.

2.5 Exchange and Selection of Leaders

At first, leaders are selected randomly. After that, leaders are selected based on their fitness values. To simulate the exchange between leader positions and other individuals, the following formula is used:

$$Stall_{G,j} = \begin{cases} X_{G,j}^{i} & if\ fitness\left(X_{G,j}^{i}\right) > fitness\left(Stall_{G,j}\right) \\ Stall_{G,j} & if\ fitness\left(X_{G,j}^{i}\right) \leq fitness\left(Stall_{G,j}\right) \end{cases} \quad (8)$$

Where $fitness\left(X_{G,j}^{i}\right)$ and $fitness\left(Stall_{G,j}\right)$ are the fitness values of foal and stallion, respectively.

2.6 Pseudo code of proposed method using WHO

-Set population size N, the maximum number of iterations T2, the number of iteration for each group T1 and PC value
-Initialize the population and calculate the fitness
-While the end criterion is not satisfied (t≤T2)
-Create foal groups and select stallions
- While the end criterion is not satisfied (t1≤T1)
Calculate TDR and Z
For the number of stallions
For the number of foals
If rand>PC then update the position of the foal using Grazing Behavior
Else update the position of the foal using Horse Mating Behavior
End if
End for
If rand > 0.5 then Generate the candidate position of stallion by Group Leadership
If the candidate position of the stallion is better
Replace the position of the stallion by the candidate position.
End if
End if
End for
End while for each group
-Update foals and stallions position using Exchange and Selection of Leaders
-End of While

-Segmented the image with the best solution

3 K-Means

K-means clustering is a very classical clustering algorithm, and it is also one of the representatives of unsupervised learning. It has the advantages of a simple idea, high efficiency, and easy implementation, so it is widely used in many fields. However, K-means clustering also has some limitations, such as the number of clusters, the value of K is challenging to select, the selection of initial class center, and so on [15].
K-means is an iterative algorithm that minimizes the sum of distances between each object and the gravity center of its cluster.
The main steps of the K-means algorithm are:

- Random choice from the initial position of the K clusters.
- Affect the objects to a cluster following to a distances Once all objects placed, recalculate K gravity center.
- Repeat steps 2 and 3 until no more affectation is made.

4 Shuffled Frog Leaping Algorithm

Shuffled Frog Leaping Algorithm (SFLA) is inspired by the behavior of a group of frogs when they are seeking for food. Two main behaviors are imitated: leaping and shuffling. A frog leaps to find a position that has more food than the current one, and then shuffles to exchange information.
The algorithm considers a population of frogs, each representing a solution to the problem of interest [14]. In [16] paper discusses the application of SFLA for multi-threshold image segmentation.

5 Experimental Results

Segmentation results of the proposed method are evaluated on some test images. It has been tested on more than 20 images taken from public images (test and satellite). In this paper, four images are selected to evaluate the efficiency of the proposed algorithm WHO, two test images and two satellites images that are derived from Landsat source.
Each image is resized to 256x256 pixels which is suitable for implementation.
Our experimental study was carried out in a hardware and software environment with the following characteristics: an Intel core i5 microprocessor, hard drive 500 GB, Windows 7 system and the JAVA programming language (NetBeans IDE 7.4).
The parameters values of the proposed method WHO determined after several tests and ensuring good convergence are recorded in the following Table 1.

Table 1. Initial parameters of WHO.

Parameter	Description	Value
N	Population size	60
G	Number of groups	10
T1	Number of iterations (local search)	50
T2	Number of iteration	100
PC	Crossover parameter	0.13

Experimental results for unsupervised images segmentation using WHO approach compared with K-means and SFLA are presented in the following figures (see Fig.1, Fig.2, Fig.3 and Fig.4).

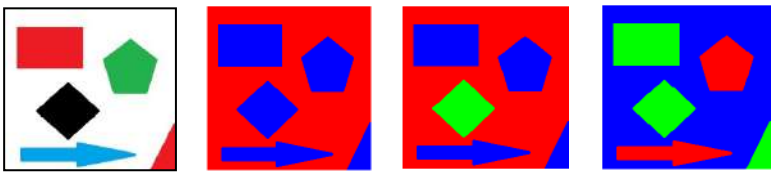

Fig.1. Image test 1 segmentation using K-means, SFLA and the WHO (respectively from left to right)

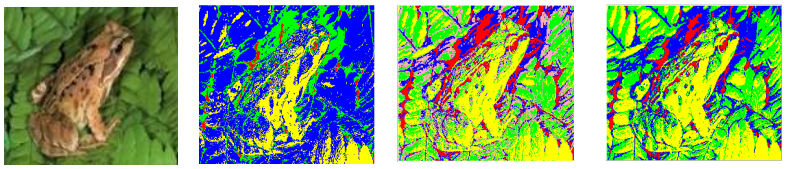

Fig.2. Image test 2 segmentation using K-means, SFLA and the WHO (respectively from left to right)

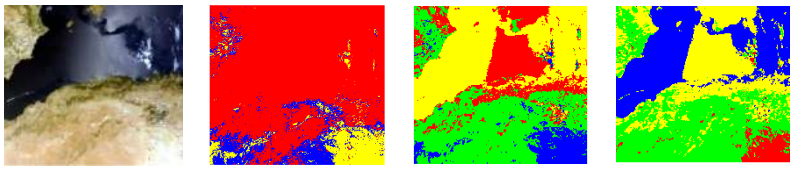

Fig.3. Image sat 1 segmentation using K-means, SFLA and the WHO (respectively from left to right)

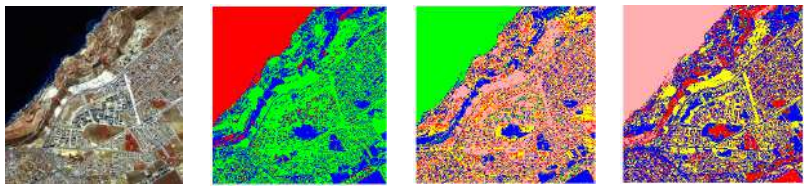

Fig.4. Image sat 2 segmentation using K-means, SFLA and the WHO (respectively from left to right)

In the Table 2, numerical results that are the run time and PSNR values [17] are reported for each method.

Table 2**.** Numerical results.

	K-means		**SFLA**		**WHO**	
	Run time (ms)	PSNR (db)	Run time (ms)	PSNR (db)	Run time (ms)	PSNR (db)
Image test 1	1749	28.14	1867	30.48	1861	30.97
Image test 2	5581	17.47	5437	27.65	5466	27.81
Image sat 1	5681	10.27	5227	29.79	5229	29.86
Image sat 2	9881	21.12	9770	28.87	9768	29.01

As these results show, the Bio inspired algorithms WHO and SFLA perform well in satellite image segmentation, and provide better results than K-means. The proposed algorithm WHO and SFLA have efficiency in image segmentation and give good results with smaller values of time execution and better PSNR. This means that their results are more homogenous than those obtained by classic K-means method.

6 Conclusion

In this paper a new image segmentation algorithm is proposed based on Wild Horse Optimization. Simulation results on two different complex images that the proposed algorithm gives superior results in less computational time.
Experiments proved that the K-means has given less satisfactory results compared to those obtained by WHO and SFLA. The Bio inspired algorithms based image segmentation are able to achieve the best results. Using this process, it is possible to circumvent local optima and thus to enhance segmentation quality. For the performance of the algorithm, the various tests carried out have shown that the choice of parameters has a significant influence on the results. Also, the initialization of parameters depends on the size of the image to be segmented and it strongly influences the quality of the segmentation. In perspective, it is important to study the choice of parameters more carefully for the Bio inspired algorithm.

References

1. Shankar,B.U.: Novel classification and segmentation techniques with application to remotely sensed images. In Transactions on Rough Sets VII, Orlowska, E. et al (Eds.), Springer-Verlag, pp.295–380, (2007).
2. Guo, D. and Atluri, V.: Texture based remote sensing image segmentation. Proceedings of the IEEE International Conference on Multimedia and Expo, pp.1472–1475, Amsterdam (2005).
3. Larabi-Marie-Sainte, S., Alskireen, R. and Alhalawani, S.: Emerging Applications of Bio-Inspired Algorithms in Image Segmentation. Electronics **10**(24), (2021).
4. Zaitoun, N.M and Aqel, M.J.: Survey on image segmentation techniques. Procedia Compu Sci, vol. 65, 797–806(2015).
5. Yuheng, S. and Hao, Y.: Image Segmentation Algorithms Overview. arXiv, pp.1707–2051, (2017).
6. Chauhan, A.S., Silakari, S. and Dixit, M.: Image segmentation methods: A survey approach. In Proceedings of the IEEE Fourth International Conference on Communication Systems and Network Technologies (CSNT), pp. 929–933, India (2014).
7. Liang, Y., Zhang, M. and Browne,W.N.: Image segmentation: A survey of methods based on evolutionary computation. In Asia-Pacific Conference on Simulated Evolution and Learning; Springer: Berlin/Heidelberg, pp. 847–859, Germany (2014).
8. Chouhan, S.S., Kaul, A. and Singh, U.P.: Image Segmentation Using Computational Intelligence Techniques: Review. Arch. Comput. Methods Eng, vol.26, 533–596(2018).
9. Chouhan, S.S., Kaul, A. and Singh, U.P. : Soft computing approaches for image segmentation: A survey. Multimed. Tools Appl, vol.77, 28483–28537 (2018).
10. Sag,T. and Çunka,S. M. : Color image segmentation based on multiobjective artificial bee colony optimization. Appl Soft Comput, vol. 34, 389–401(2015).
11. Bose, A.: Fuzzy-based artificial bee colony optimization for gray image. Signal Image Video Process, vol.10, 1089–1096(2016).
12. Naruei, I. and Keynia, F.: Wild Horse Optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. Eng Computer **38**(1), 1–31(2021).
13. Li, L. and Zhang, M. : Application of Improved Wild Horse Optimizer Based on Chaos Initialization in Medical Image Segmentation. Proceedings of the 13th International Conference on Computer Engineering and Networks. vol. 1125, pp. 334–343, Singapore (2024).
14. Tehami,A. and Fizazi,H. : Unsupervised Segmentation of Images Based on Shuffled Frog-Leaping Algorithm. Journal of information processing systems **13**(02), 370–384(2017).
15. Senthil,N. : K-Means Algorithm based Satellite Image Segmentation. Journal of Engineering and Applied Sciences **12**(07), 7995–7997(2017).
16. Chen,Y. and Q,Zhang. : Multi-thershold Image segmentation using a multi-startegy Shuffled Frog Leaping Algorithm. Expert Systems with Applications. vol. 194 (2022).
17. Panwar, P. and Gopal, G. : Image Segmentation using K-means clustering and Thresholding. International Research Journal of Engineering and Technology **03**(05), 1787–1793 (2016).

Adaptive dashboards for computer-supported collaborative learning: A systematic literature review using PRISMA

Kaouther Soltani[1] , Nadia Hocine[1], and Karim Sehaba[1]

University of Mostaganem, CSTL Lab
kaouther.soltani.etu, nadia.hocine, karim.sehaba@univ-mosta.dz

Abstract. Computer supported collaborative learning has emerged as one of the most effective learning methods. Its numerous advantages include improving student learning skills and enhancing teachers' strategies for orchestrating their classrooms. In particular, adaptive learning analytics dashboards can play an important role in enabling more effective monitoring and engagement. In this paper, we present our research methodology to systematically review studies on adaptive dashboards in CSCL environments. This methodology aligns with the PRISMA protocol. We identified 23 relevant articles between 2017 and 2024 written in English for the analysis. We presented an overview of the results that shows how effective the dashboard in helping students to improve their learning and to assist teachers to monitor their classes. The dashboard mainly visualizes student feedback and supports transitions between individual and collaborative learning. However, it lacks personalization for different learning styles, limiting its ability to tailor courses to individual performance.

Keywords: Dashboards, Computer-Supported Collaborative Learning, Learning Analytics, Adaptation.

1 Introduction

CSCL (Computer Supported Collaborative Learning) is a learning approach based on the interaction of learners with the support of information and communication technologies (ICT). It focuses on collaborative learning principals where a group of students collaborate to complete a task or to solve a problem to meet an objective [1, 17]. A wide range of online tools and platforms were proposed to promote collaborative work and better support it. They were used for instance to facilitate teams communication and to improve individual and team skills [5, 17, 30]. Although collaborative systems and group awareness tools enable awareness and reflection in teams, they can be impacted by some collaboration problems. They present some limitations related to the consideration of collaboration issues such as the free-rider effect [13, 18].

Learning Analytics (LA) is thought of as a solution that helps address these problems by analyzing learners' traces to identify collaboration issues and help teams to achieve their goals. LA can play an important role in identifying collaboration issues within a

group. Learner interactions with the learning environment, including their communication and discussion are analyzed to track their participation, and detect collaboration issues such as the lack of communication between group members or participants that are less active than others [26].

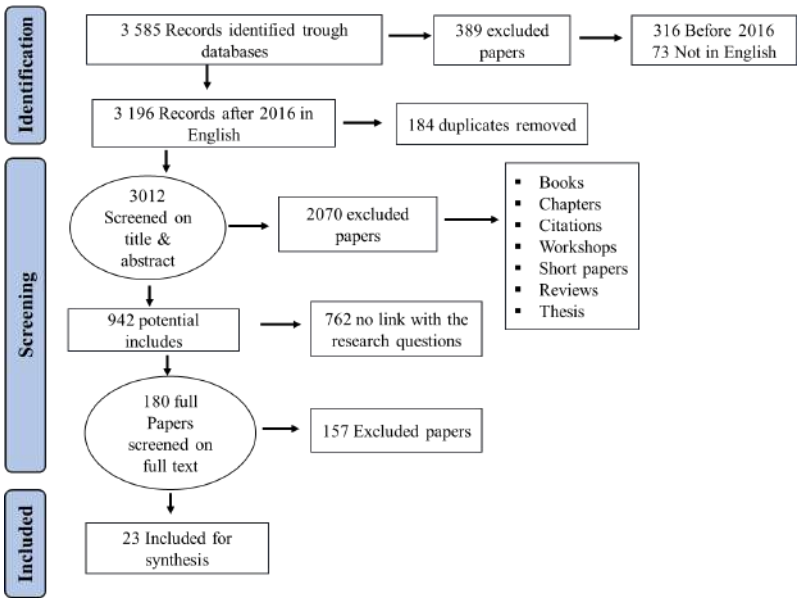

Fig. 1. PRISMA paper selection process.

Learning analytics dashboards (LADs) provide visualizations of the most relevant learning indicators using for instance network graphs and bar charts. LAD can be also helpful to detect collaboration problems [18]. The dashboard can support the teachers in identifying collaboration issues and proposes solutions such as the dynamic transition between individual and collaborative learning.

This systematic literature review focused on the role of the dashboards in supporting collaborative learning. This paper seeks to answer two broad research questions: RQ1. How have adaptive learning systems based on dashboards contributed to improving collaborative learning and classroom orchestration?

RQ2. How have dashboards been utilized to provide adaptive support for collaborative learning?

2 Research methodology

Our methodology is based on the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) protocol [21]. We used this protocol as shown in Fig 1, emphasizing the importance of following them to conduct an effective and reliable systematic review.

Table 1. database query.

Database	Query
Conceptual query	(collabor* or "computer supported collaborative learning") and Dashboard and (adapt* or personal*) and "learning analytics"
IEEE	(("Full Text & Metadata":"collabor*" OR "Full Text & Metadata":"computer supported collaborative learning") AND ("Full Text & Metadata":"Dashboard") AND ("Full Text & Metadata":"adapt*" OR "Full Text & Metadata":"personal*") AND ("Full Text & Metadata":"learning analytics"))
Springer	(collabor* or "computer supported collaborative learning") and Dashboard and (adapt* or personal*) and "learning analytics"
ACM	[All: "computer supported collaborative learning"] OR [All: cscl]] AND [All: dashboard] AND [[All: adapt*] OR [All: personal*]] AND [All: "learning analytics"] AND [E-Publication Date: (01/01/2016 TO 12/31/2024)]
Google Scholar	(collabor* or "computer supported collaborative learning") and Dashboard and (adapt* or personal*) and "learning analytics"

To begin our research, we created a list of keywords related to the research questions, to compose this query: (collabor* or "computer supported collaborative learning") and Dashboard and (adapt* or personal*) and "learning analytics". After that, we have translated the query for each database according to their syntax in order to search in different databases IEEE (Institute of Electrical and Electronics Engineers) -Xplore, Springer, ACM digital library, and Google Scholar as displayed in Table 1. The search results are displayed in Table 2 for each database.

Table 2. search results.

Database	Result
IEEE	290
Springer	1701
ACM	374
Google Scholar	1220

Following the database search, we apply the inclusion and exclusion criteria to select the relevant studies. After identifying all relevant articles using the inclusion and exclusion criteria, first of all we removed duplicates, then we screened the title and abstract for each selected articles to check if it is related to our systematic review. In the final step we read all the articles in full to select the included papers and decide whether the article is relevant to our research needs.

The analysis criteria that were applied to the selected items are:

– C1. What is the article's main objective in terms of improving or assisting collaboration?

– C2. The dashboard is for whom?
– C3. What is the study result?
– C4. When, how, and for whom the dashboard is adapted?

3 Results

We found a total of 23 articles. Table 3 shows the distribution of papers according to their venue. The majority of articles aim to enhance collaboration within the context of CSCL by helping students to empower their knowledge and their collaboration through novel educational methodologies [10]. This methodology enhances several competences, including communication skills, student collaboration and self-regulation [10, 16]. They also assist teachers to monitor the students' reflections in order to improve their learning process and improve their teaching strategies [8, 16]. Some articles emphasize the significance of developing students' collaborative problem-solving skills due to its positive impact on the learning process [31].

Table 3. The distribution of papers according to their venue.

Source	Authors
International Journal of Computer-Supported Collaborative Learning	[4, 19, 25]
Computers and Education Journal	[11, 31]
IEEE Transactions on Learning Technologies journal	[3, 7]
International Learning Analytics and Knowledge Conference (LAK)	[9, 22]
International Conference on Computers in Education	[6]
Pacific Asia Conference on Information Systems	[16]
Journal of Computers in Human Behavior	[23]
ZDM--Mathematics Education Journal	[8]
International Journal of Artificial Intelligence in Education	[20]
International Educational Data Mining Society	[28]
European Conference on Technology Enhanced Learning (ECTEL)	[24]
International Conference on Artificial Intelligence in Education	[29]
Journal of Sustainability	[2]
International Conference on Computer-Supported Collaborative Learning (CSCL)	[15]
Smart Learning Environments Journal	[10]
CHI conference on human factors in computing systems	[27]
Journal of Technology, Knowledge and Learning	[12]
British Journal of Educational Technology	[14]

3.1 Use of the dashboard

The dashboard is intended for students and teachers. It provides information and visalisations on students learning or their own learning progress, that enable them to evaluate and adjust their learning [10, 31].

The instructor has used the dashboard to monitor student progress, intervene at the appropriate times to assist them and improve their instructional method [11, 20, 31].

3.2 Studies' result

Overall, Results indicate the importance of the dashboard in improving student learning and supporting teachers in order to enhance learning and teaching.

3.3 Adaptation

In most of the articles, the dashboard was adapted according to the student's need in order to personalize their learning. Moreover, in some articles, it was adapted for teachers to control team members learning.

Among the most used adaptation techniques were: adaptive feedback displayed on the dashboard, or dynamic transaction between individual and collaborative learning using intelligent tutoring systems.

4 Discussion and conclusions

In this paper, we applied step by step the PRISMA protocol to conduct a systematic review. This protocol helped us to identify and select all the relevant articles for this systematic review. The results of this study indicate the importance of the dashboard in improving students' learning outcomes and supporting teachers' by providing real time visualizations.

Although collaboration offers several advantages, including motivation, the exchange of ideas among members, and the development some skills, it requires communication and coordination. Despite this, there are problems that lead to the deterioration of collaboration like the lack of communication on the part of some students, the free-rider problem, that some students prefer to work individually rather than collaborate with others. When they are integrated into one group, their progress deteriorates, which affects the progress of the others. The analyzed articles often failed to address these issues.

LADs were used only as a visualization tool to provide a feedback on learner progress. This can help teachers intervene and solve collaboration problems. These techniques used help to improve collaboration but not to detect and solve collaboration issues.

Finally, our perspectives for future work is to suggest an adaptation approach that takes into account students' needs to adapt the dashboard, especially their learning styles.

References

1. Aderibigbe, S. A., Abdel Rahman, A. R. A., ELMneizel, A. F., & Al Gharaibeh, F. Undergraduate students views about peer mentoring as a tool to enhance computer-supported collaborative learning. Contemporary Educational Technology, 15(4), ep461 (2023).
2. Aldosemani, T. I., & Al Khateeb, A. Learning loss recovery dashboard: A proposed design to mitigate learning loss post schools closure. Sustainability, 14(10), 5944 (2022).
3. Amarasinghe, I., Hernández-Leo, D., Michos, K., & Vujovic, M. An actionable orchestration dashboard to enhance collaboration in the classroom. IEEE Transactions on Learning Technologies, 13(4), 662–675 (2020).
4. Amarasinghe, I., Hernández-Leo, D., & Ulrich Hoppe, H. Deconstructing orchestration load: comparing teacher support through mirroring and guiding. International Journal of Computer-Supported Collaborative Learning, 16(3), 307–338 (2021). https://doi.org/https://doi.org/10.1007/s11412-021-09351-9
5. Bao, H., Li, Y., Su, Y., Xing, S., Chen, N.-S., & Rose, C. P. The effects of a learning analytics dashboard on teachers' diagnosis and intervention in computer-supported collaborative learning. Technology, Pedagogy and Education, 30(2), 287–303 (2021).
6. Echeverria, V., Martinez-Maldonado, R., Chiluiza, K., & Buckingham Shum, S. (2017). DBCollab: Automated feedback for face-to-face group database design. In A.-P. S. for Computers in Education (Ed.), Proceedings of the 25th International Conference on Computers in Education, ICCE 2017, Christchurch, New Zealand (pp. 56–165). ACM Press.
7. Echeverria, V., Yang, K., Lawrence, L., Rummel, N., & Aleven, V. Designing hybrid human–AI orchestration tools for individual and collaborative activities: A technology probe study. IEEE Transactions on Learning Technologies, 16(2), 191–205 (2023).
8. Edson, A. J., & Phillips, E. D. Connecting a teacher dashboard to a student digital collaborative environment: Supporting teacher enactment of problem-based mathematics curriculum. ZDM–Mathematics Education, 53(6), 1285–1298 (2021).
9. Fernandez-Nieto, G. M., Martinez-Maldonado, R., Echeverria, V., Kitto, K., Gašević, D., & Buckingham Shum, S. Data storytelling editor: A teacher-centred tool for customising learning analytics dashboard narratives. Proceedings of the 14th Learning Analytics and Knowledge Conference, LAK 2024, Kyoto, Japan, 678–689 (2024).
10. Hadyaoui, A., & Cheniti-Belcadhi, L. Ontology-based group assessment analytics framework for performances prediction in project-based collaborative learning. Smart Learning Environments, 10(1), 43 (2023).
11. Han, J., Kim, K. H., Rhee, W., & Cho, Y. H. Learning analytics dashboards for adaptive support in face-to-face collaborative argumentation. Computers and Education, 163, 104041 (2021).
12. Kaliisa, R., & Dolonen, J. A. CADA: a teacher-facing learning analytics dashboard to foster teachers' awareness of students' participation and discourse patterns in online discussions. Technology, Knowledge and Learning, 28(3), 937–958 (2023).
13. Kerr, N. L., & Bruun, S. E. Dispensability of member effort and group motivation losses: Free-rider effects. Journal of Personality and Social Psychology, 44(1), 78 (1983).
14. Lawrence, L. E. M., Echeverria, V., Yang, K., Aleven, V., & Rummel, N. How teachers conceptualise shared control with an AI co-orchestration tool: A multiyear teacher-centred design process. British Journal of Educational Technology, 55(3), 823–844 (2024).
15. Lawrence, L. E. M., Guo, B., Yang, K., Echeverria, V., Kang, Z., Bathala, V., Li, C., Huang, W., Aleven, V., & Rummel, N. Co-designing AI-based orchestration tools to support dynamic transitions: Design narratives through conjecture mapping. International Conference

on Computer-Supported Collaborative Learning 2022, Hiroshima, Japan, 139–146. International Society of the Learning Sciences (2022).
16. Lin, Y. L., Lee, M. W., & Hsiao, I. H. An exploratory study on programming orchestration technology. Proceedings of the 22nd Pacific Asia Conference on Information Systems - Opportunities and Challenges for the Digitized Society, Yokohama, Japan (2018).
17. Lipponen, L. Exploring foundations for computer-supported collaborative learning. CSCL '02: Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community, Boulder Colorado, 72–81. Routledge, International Society of the Learning Sciences (2002).
18. Liu, A. L., & Nesbit, J. C. Dashboards for computer-supported collaborative learning. Machine Learning Paradigms: Advances in Learning Analytics, 157–182 (2020).
19. Martinez-Maldonado, R. A handheld classroom dashboard: Teachers' perspectives on the use of real-time collaborative learning analytics. International Journal of Computer-Supported Collaborative Learning, 14, 383–411 (2019).
20. Olsen, J. K., Rummel, N., & Aleven, V. Designing for the co-orchestration of social transitions between individual, small-group and whole-class learning in the classroom. International Journal of Artificial Intelligence in Education, 31(1), 24–56 (2021).
21. Page, M., Mckenzie, J., Bossuyt, P., Boutron, I., & Hoffmann, T. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Systematic Reviews, 10(1), 1–11 (2021).
22. Praharaj, S., Scheffel, M., Schmitz, M., Specht, M., & Drachsler, H. Towards Collaborative Convergence: Quantifying Collaboration Quality with Automated Co-located Collaboration Analytics. LAK22: 12th International Learning Analytics and Knowledge Conference, Online, USA, 358–369 (2022). https://doi.org/10.1145/3506860.3506922
23. Sedrakyan, G., Malmberg, J., Verbert, K., Jarvela, S., & Kirschner, P. A. Linking learning behavior analytics and learning science concepts: Designing a learning analytics dashboard for feedback to support learning regulation. Computers in Human Behavior, 107, 105512 (2020).
24. Serrano Iglesias, S., Spikol, D., Bote Lorenzo, M. L., Ouhaichi, H., Gomez Sanchez, E., & Vogel, B. Adaptable Smart Learning Environments supported by Multimodal Learning Analytics. Proceedings of the LA4SLE 2021 Workshop : Learning Analytics for Smart Learning Environments Co-Located with the 16th European Conference on Technology Enhanced Learning 2021 (ECTEL 2021), Online (Bozen-Bolzano, Italy), 24–30 (2021). https://ceur-ws.org/https://ceur-ws.org/
25. Silva, L., Mendes, A., Gomes, A., & Fortes, G. Fostering regulatory processes using computational scaffolding. International Journal of Computer-Supported Collaborative Learning, 18(1), 67–100 (2023).
26. Van Leeuwen, A., Janssen, J., Erkens, G., & Brekelmans, M. Teacher regulation of cognitive activities during student collaboration: Effects of learning analytics. Computers and Education, 90, 80–94 (2015).
27. Yang, K. B., Echeverria, V., Lu, Z., Mao, H., Holstein, K., Rummel, N., & Aleven, V. Pair-up: prototyping human-AI co-orchestration of dynamic transitions between individual and collaborative learning in the classroom. Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems CHI 2023, Hamburg, Germany (2023). https://doi.org/10.1145/3544548.3581398
28. Yang, K. B., Echeverria, V., Wang, X., Lawrence, L., Holstein, K., Rummel, N., & Aleven, V. Exploring Policies for Dynamically Teaming up Students through Log Data Simulation. International Educational Data Mining Society (2021).

29. Yang, K. B., Lu, Z., Echeverria, V., Sewall, J., Lawrence, L., Rummel, N., & Aleven, V. Technology ecosystem for orchestrating dynamic transitions between individual and collaborative AI-tutored problem solving. International Conference on Artificial Intelligence in Education, AIED 2022, Springer, Cham, 673–678 (2022).
30. Yilmaz, R. Using zoom as a computer-supported collaborative learning tool: modeling of relations between technology acceptance, knowledge-sharing behaviours, community of inquiry, and social interaction space. Interactive Learning Environments, 1–19 (2023).
31. Zamecnik, A., Kovanovic, V., Grossmann, G., Joksimovic, S., Jolliffe, G., Gibson, D., & Pardo, A. Team interactions with learning analytics dashboards. Computers and Education, 185, 104514 (2022). https://doi.org/https://doi.org/10.1016/j.compedu.2022.104514

Enhancing performance for remote Labs based to RESTful API and MERN stack technologies.

Ben Amara Said[1[0000-0002-8844-5412]], SidAhmed Henni[2[]]

, Mohammed Moussa[3[0000-0003-4508-1337]], Abdelhalim Benachenhou[4[0000-0001-6271-7259]]
[1,2,3,4] Abdelhamid Ibn Badis University, Mostaganem, Algeria

said.benamara.etu@univ-mosta.dz

Abstract.

In this work, we present a smart solution to the problem of reaching a huge demand, which comes from a large number of students, to access materials in remote laboratories. This solution is to turn a physical experiment into a data set by recording all of its possible states, where we could distribute fully digitized experiment over the internet. Our approach to enhancing performance is based on the powerful of MERN stack technologies.

Keywords: Remote laboratories, Enhancing performance, MERN stack technologies.

1 Introduction

In recent years, many countries around the world have faced problems due to the Corona epidemic, including the education sector, and this is in order to follow prevention protocols and impose social distancing. This forced the governments to impose the obligation of distance education, in the end.
The educational institutes have adapted to using video conferencing platforms such as Zoom to conduct lectures in a virtual online classroom. However, it is important to understand the strengths and weaknesses of each one because different modes of education can aid or weaken the development of various learning outcomes [1]. Therefore, it is important to match the remote education with the desired learning outcome intended. In STEM (Science, Technology, Engineering, and Mathematics) education, students are often required to interact with physical lab hardware or equipment to make measurements and learn the underlying principles of subjects taught in class (or remotely). Unlike class lectures, video conferencing platforms lack the interactivity to recreate laboratory experiments. Thus, there is a need for a teaching/learning platform for conducting laboratory experiments remotely. It is well acknowledged that the remote laboratory is capable of enhancing student learning across thirteen laboratory objectives [2] that include instrumentation, models,

experimentation, data analysis, design, learning from failure, creativity, psychomotor, safety, communication, teamwork, ethics and sensory awareness. These objectives showcase the multifaceted benefits and diverse learning experiences covering the cognitive, psychomotor, and affective learning domains that can be incorporated and assessed through experimentation [3]. Our purpose is to convert a physical laboratory environment into an online experience. The RL can be accessed simultaneously by a large number of students online [4]– [5]. However, remote laboratories suffer from scalability issues as only one user or at most a few users can access the experiment at a given time.

In our recent work, we tackle remote experimentation and its current lack of scalability with our new platform, which is available at www.Mostalab.com. The most recent version of Mostalab relies on turning an experiment into a data set before displaying it. This can be thought of as digitizing the experiment [6]. This digitization process is based on the observation that a large portion of laboratory experiments involves varying input control parameters and observing output parameters. A digitized experiment can be represented as a series of output values for any given combination of input values. By varying the input control parameters, we put the experiment into a distinct state that can be recorded. Here, recording implies capturing all the relevant output data for that state. The output values can be, amongst others, readings on a current or voltage on a screen, which can be stored digitally. As the data is recorded from an actual physical experiment.

The platform, an existing setup, is automatically turned into a data set, accessible through database queries. In this way, our solution concept provides an effective and scalable solution to add an important element to current online education systems at low costs. Furthermore, the main goal in this study is to enhance the performance of that platform.

2 Related work

There are lots of efforts from universities and companies to develop remote laboratory projects and offer the best solution for e-learning. Some of them are based on a virtualization approach called MSOL (Massively Scalable Online Laboratories), others on the reservation or the queue.

LabsLand connects schools and universities with real laboratories available somewhere else on the Internet. A real laboratory can be a small arduino-powered robot in Spain, a kinematics setup in Brazil or a radioactivity-testing lab in Australia. They are real laboratories, not simulations: the laboratories are physically there, and students from these schools and universities can access them [7].

Neustock [8] is an enhanced version of the iLabs platform developed at Stanford in It is based on a virtualization approach called Massively Scalable Online Labs (MSOL) [9].

The main idea of MSOL revolves around transforming a real experiment into a set of data by storing all of its possible states.Markan et al. [10] have used a reduced-duration

laboratory session in "batch mode." so that the access to RLs does not need any prior reservation, which provides great multi-user scalability.
MIT's iLab project has proposed a reservation system that involves many operations and supports user reservation of platforms [11]. The EOLES (Electronics and Optics for Embedded System) project has permitted students from different countries to conduct remote experiments by sharing ten platforms and using the reserving time slots [12]. The queue has been implemented in WebLab-Deusto. Each request for access to the laboratory is pushed into a FIFO (First In First Out) queue. The combination of the reservation / queue enhances the performance of the RL management system has been demonstrated by Lowe [13], through the indicator of the overall level of use (session duration) and the times of waiting queues.

3 Methodology

In our work, we have conceptualized a highly scalable version of an online remote laboratory where a large number of users can easily access the platform of a lab and share with other students. That lab is implemented into MostaLab, which is based on a service-oriented approach using the Lab as a service (LaaS). The main idea behind our approach is split into two phases. The first phase is to turn a real experiment into a data set by storing all of its possible cases. The laboratory experiment can then be represented as a high scalable platform to a user who can observe the virtual experiment, similar to the experience of observing a real experiment via the internet,
in which users can carry out a multiple set of experiment with the option of configuration for each experiment through a web page. The digitalization of the experiment is designed to be interactive. Students can study how the experiment works according to the inputs that they select, and they know how to operate the equipment and observe the experimental results changing as a result of their choices.
To reach a huge number of students to access remote labs, scalability became a desirable attribute. Poor scalability leads to poor system performance. The performance of websites was always a critical non-functional requirement. A better-performing site directly result in better user experience. To achieve this purpose there are techniques, methods, and technologies used, such as a MERN stack that uses Mongoose and the MongoDB database. Chrome developer tools were utilized while testing using Redux tools for simulation.

3.1 MERN Stack Components

MongoDB.

MongoDB is much more than a database [14]. It is a full cloud-based application data platform. You have access to a collection of services such as Performance Advisor, Atlas Search and much more that all integrate nicely with your database. Weused Document-Oriented Database where every record is document format. The BSON data format, inspired by JSON, allows you to store and query more efficiently. When dealing

with real-world data and adjusting to shifting conditions or the environment in case our system collapses, MongoDB's explicit schemas and validating data are incredibly flexible. With just a few lines of declarative code, you can run complicated analytics pipelines using the MongoDB Query API. In terms of write performance, MongoDB offers the insertMany and updateMany functions, which let you insert and update numerous data at once. When compared to typical databases' batched writes, these two functions provide a sizable performance improvement. To maintain performance and scale horizontally, build clusters with real-time replication and shard big or high-throughput collections across many clusters.

Express.JS.

Express is a minimalist web framework for Node.js that is fast and unopinionated. It provides various features that make web application development fast and easy, which would otherwise take more time using only Node.js.Express.js is built on the Node.js middleware module connects, which uses the http module. As a result, any middleware based on connect will also work with Express.js.

ReactJS.

When developing a complex, high-load app, it is essential to define the structure of the app from the start because it can affect the performance of our app. To put it simply, the DOM model is tree-structured. As a result, a minor change at a higher-level layer can have a significant impact on an application's user interface. Facebook has introduced a virtual DOM feature to address this issue. A virtual DOM, as the name implies, is a virtual representation of DOM that allows testing all changes to the virtual DOM first in order to calculate risks with each modification. As a result, this approach helps to maintain high app performance and ensures a better user experience.

NodeJS

Node.js is written in C++ and runs on the JS operating system [15]. Node.js is a runtime environment for JS. For optimal performance, Node.js employs the Google Chrome V8 engine. Node.js is designed with a single-thread architecture and uses event-driven, asynchronous programming callback functions. The event-driven design of Node.js is the fundamental core concept for its environment. The main benefit of event-driven and asynchronous programming is that it is single-threaded. The call back function code is executed without waiting for a specific code to complete, and the limited resources were used for other tasks that would be executed as part of us web application's business logic. This design was appropriate for our back-end development, which was also the system's goal. Handling synchronous requests was a major task in server development, and blocking was the cause of not fully utilizing or wasting resources. We improved resource utilization and website performance by using single thread architecture and asynchronous callback functions, which also

provided us with the desired results during testing.

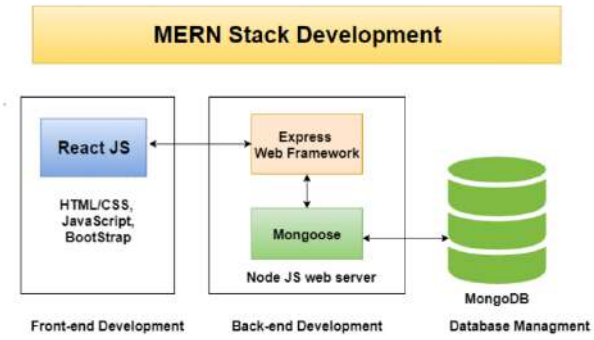

Fig. 1. A figure illustrates the MERN stack component.

3.2 Implementation

Turning an Experiment into a Data-Set The first phase in turning an existing experiment into a platform to save the information according to all possible state, such as values from power supply or DMM of the experiment. Most recent experiments are controlled by computer. which allows a program to iterate through all possible stages of all controls automatically with only very little extra effort. Similar experiments like this can be turned into data by this method.

During or after recording the experiment, experiment data can be uploaded to a server. This upload will contain a data file where we can calculate the latency queries.

For the purposes of this paper, we chose to demonstrate the functionality of the platform with a resistivity experiment.

Resistivity is the tendency of a material to behave as a resistor. You already know that not everything conducts electricity equally well, and that some materials (like copper) resist very little, while others (like rubber) provide enough resistance to effectively prevent the flow of current.

We can easily test the cross-area section dependence, and simultaneously find the resistivity of an unknown wire.

Fig.1 illustrates the circuit diagram comprising the various possible combinations. It allows to select one resistor among 6. The DMM can be used as a voltmeter or as an ammeter. Fig.2 shows the hardware implementation comprising a lab server, a switching device and the component board. Fig. 3 illustrates the graphical interface. The end user selects a resistor, the voltage V of the power supply and sends a request to read the voltage across the resistor and the current. In the physical device, the DMM is configured as a voltmeter, measures the voltage across the resistor then as an ammeter, and measures the current flowing through the resistor. The results are displayed in separate virtual instruments giving the feeling of having two instruments.

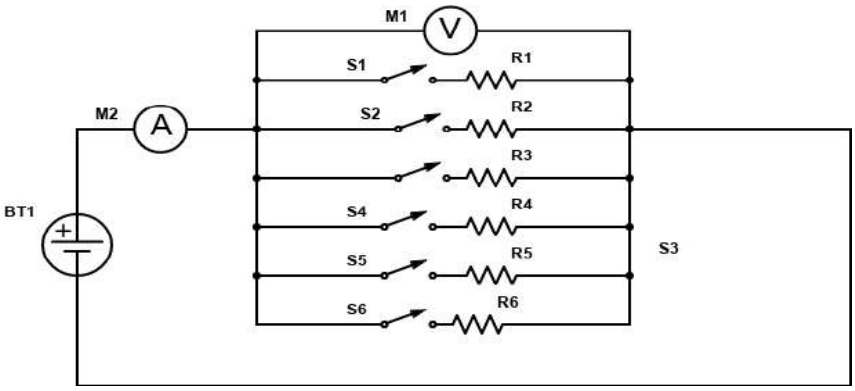

Fig. 2. A figure illustrates the circuit diagram.

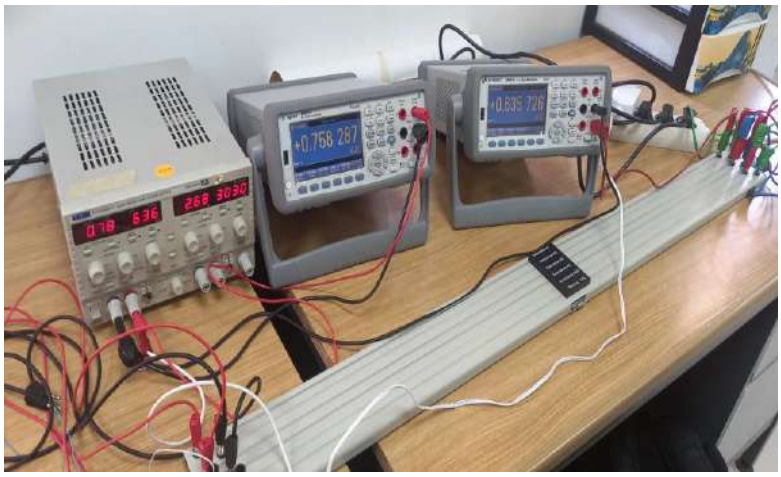

Fig. 3. A figure illustrates the implementation of the hardware

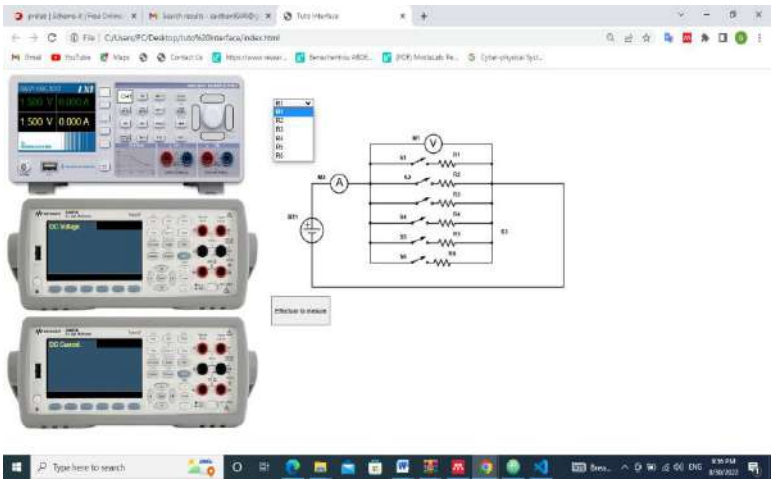

Fig. 4. A figure illustrates the user interface.

4 Result and Discussion

The current remote Lab Web application uses MERN stack technologies, which are used in the development project. This project intends to provide a critical examination of the relevant literature. Literature in the field of remote lab, as well as to describe key aspects of the methodology we have used throughout the undertaking. This project was successful in understanding a variety of issues. That arise during the development of the application. We discovered that applications are more than just software artifacts. Mastering the required technologies or stack for developing any web application is required. Concentrating on other issues that arise during the development process, such as website evaluation, field research, and selecting the best model for our remote lab web application, was created. These are the first and most important steps in ensuring that the final application is developed in accordance with university demands and is tailored to the needs of its students. More research and attention was paid to software testing tools. All necessary decisions on how the website will be built were made based on the results of the problem investigation stage because they played a significant role in describing the specific student requirements for the web application.

5 Conclusion

In the summary, it says that the digitalization of the experiment is a method to enhance the scalability; it can help students conduct more experiments with fewer resources (money, time, and space); and the platform is especially suited to encourage students to go over any specific information of an experiment at any time and repeat parts of the experiment about which they are especially misunderstand. There will be many new features, such as quizzes, sketch circuits starting from zero, and conversation between users in real time. In addition, we will use a load-testing toolkit, such as Artillery.io, to assess the effectiveness of our solution. These features will be included in future studies. In the case of complex experiments (with a large number of cases), collaboration between universities and the creation of a community to share data is critical to resolving this issue.

References

1. LINDSAY, E. D. & GOOD, M. C. 2005. Effects of laboratory access modes upon learning
outcomes. IEEE Transactions on Education, 48, 619-631.
2. FEISEL, L., PETERSON, G. D., ARNAS, O., CARTER, L., ROSA, A. & WOREK, Learning objectives for engineering education laboratories. Frontiers in Education, 2002.FIE 2002. 32nd Annual, 2002 2002. F1D-1 vol.2.

3. NIKOLIC, S., SUESSE, T., JOVANOVIC, K. & STANISAVLJEVIC, Z. 2021. Laboratory Learning Objectives Measurement: Relationships Between Student Evaluation Scores and Perceived Learning. IEEE Transactions on Education, 64, 163- 171.10

4. Ruben Heradio, Luis De La Torre, Daniel Galan, Francisco Javier Cabrerizo, Enrique Herrera-Viedma, and Sebastian Dormido. Virtual and remote labs in education: A bibliometric analysis. Computers & Education, 98:14–38, 2016.

5. Emily Kaye Faulconer and Amy B Gruss. A review to weigh the pros and cons of online,remote, and distance science laboratory experiences. International Review of Research in Open and Distributed Learning, 19(2), 2018.

6. Lars Thorben Neustock, George K Herring, and Lambertus Hesselink. Remote experimentation with massively scalable online laboratories. In Online Engineering & Internet of Things, pages 258–265. Springer, 2018.

7. https://labsland.com/en (accessed on 20/08/2022).

8. Neustock L.T., Herring G.K., Hesselink L., Remote Experimentation with Massively Scalable Online Laboratories. In: Auer M., Zutin D. ed. Online Engineering & Internet of Things. Lecture Notes in Networks and Systems, vol 22. Springer, Cham, 2018.

9. Harward, V. J., Del Alamo, et al. The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories. Proceedings of the IEEE, 2008, 96(6), 931-950.

10. Markan C. M., Gupta P., Kumar G., et al. Scalable Multiuser Remote Laboratories provide on-demand hands-on laboratory experience. In: IEEE Conference on Technology and Society in Asia (T&SA). IEEE, 2012. p. 1-7.

11. http://icampus.mit.edu/projects/ilabs (accessed on 20/12/2020).

12. Andrieu, G., Farah, S., Fredon, T., Benachenhou, A., Ankrim, M., Bouchlaghem, K..... & Cristea, M. (2016, February). Overview of the first year of the L3-EOLES training. In 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV) (pp. 396-399). IEEE.

13. Lowe, D. (2013). Integrating reservations and queuing in remote laboratory scheduling IEEE Transactions on Learning Technologies, 6(1), 73-84.

14. https://www.mongodb.com (accessed on 15/09/2022).

15. https://nodejs.org(accessed on 12/09/2022).