BOOK OF PROCEEDINGS

ADVANCES IN COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

International Conference on Computer Science and Information Technology CSIT® 2025

LUIS PRINT
SHTËPI BOTUESE | PRINTIME

BOOK OF PROCEDEENGS

1st International Scientific Conference
on Computer Science and Information Technology - CSIT® 2025

Proceedings of “Advanced in Computer Science and Information Technology”

LUIS PRINT
2025

BOOK OF PROCEDEENGS

1st International Scientific Conference
on Computer Science and Information Technology - CSIT® 2025

Proceedings of "Advanced in Computer Science and Information Technology"

Editors: Edmira XHAFERRA
Department of Information Technology/ Aleksander Moisiu University Durres, Albania

Uendi ÇERMA

Department of Information Technology/ Aleksander Moisiu University Durres, Albania

Viola SHTINO

Department of Computer Science/ Aleksander Moisiu University Durres, Albania

Eda TABAKU

Department of Computer Science/ Aleksander Moisiu University Durres, Albania

Rinela KAPÇIU

Department of Computer Science/ Aleksander Moisiu University Durres, Albania

Fatjona BUSHI

Department of Computer Science/ Aleksander Moisiu University Durres, Albania

Përgatiti për botim: Luis Print

ISBN: 9789928835147

LUIS PRINT
SHTËPI BOTUESE | PRINTIME

Shtëpi Botuese, Studio Grafike, Shtypshkronjë
Adresa: Lagjia nr.1, Rruga Taulantia pranë Universitetit "Aleksandër Moisiu", Durrës
Email: luisiprint@gmail.com
www.luisprint.com
Cel: +355 69 364 6568

Preface

We are privileged to introduce the Proceedings Book of the ***1st International Conference on Computer Science and Information Technology – CSIT® 2025***, collaboratively organised by the **University "Aleksandër Moisiu" Durrës (UAMD)** and the **South East European University (SEEU)**. The conference, scheduled for June 19–20, 2025, in Durrës, Albania, marks a pivotal advancement in fostering research, innovation, and global collaboration in the rapidly evolving fields of Computer Science and Information Technology.

The objective of CSIT® 2025 is to establish a dynamic and inclusive forum for researchers, academics, industry experts, and students to exchange knowledge, disseminate innovations, and discuss emerging trends and issues in the digital realm. In an era marked by constant technological advancements, the importance of computer science is paramount. The domains of artificial intelligence, cybersecurity, cloud computing, robotics, and sustainable IT are continually expanding, influencing nearly all facets of our lives, economies, and communities.

In recognition of the importance of accessibility and global outreach, CSIT® 2025 has been structured as a hybrid conference, accommodating both in-person and virtual participation. This style increases event inclusivity and emphasises the flexibility and interconnectedness of digital technologies. Researchers and professionals globally have convened, both digitally and in person, to showcase their work, participate in substantive conversations, and establish new connections.

The conference has garnered contributions from several fields and institutions, resulting in a comprehensive and diverse scientific agenda. The subjects addressed encompass:

- Artificial Intelligence and Machine Learning
- Data Science and Big Data Analytics
- Cybersecurity and Privacy
- Software Engineering
- Internet of Things (IoT)
- Cloud Computing and Distributed Systems
- Human-Computer Interaction
- Educational Technology and AI in Learning
- Blockchain and Decentralized Systems
- AR/VR and Immersive Technologies
- Mathematics in Data Analysis and Artificial Intelligence

The extensive breadth of these subjects illustrates the multidisciplinary nature of contemporary computer science and its crucial significance in both theoretical advancements and practical applications. This volume's contributions offer innovative ideas and techniques, demonstrating an increasing understanding of the social, ethical, and environmental aspects of technology.

This *Proceedings Book* comprises peer-reviewed articles chosen for presentation at CSIT® 2025. The Scientific Committee has meticulously assessed each submission to ensure high academic standards, originality, and relevance. The selected papers exemplify new thought, thorough analysis, and intriguing research trajectories that we anticipate will stimulate additional investigation and collaboration.

We extend our sincere gratitude to all contributors who submitted their research and generously shared their skills. Your contributions are crucial to this conference and will influence its future iterations. We extend our gratitude to our **Scientific Committee** and reviewers for their commitment and academic integrity, which guaranteed a fair and high-quality review process. Their endeavours facilitated the preservation of the academic rigour that CSIT® aspires to sustain.

We extend our heartfelt gratitude to our **keynote speakers**, session chairs, workshop organizers, and panelists. Their thoughts, leadership, and readiness to engage with varied audiences have greatly enhanced the event's intellectual depth.

The **Organising Committee** has diligently laboured to realise this aim. The committee's initiatives in logistics, planning, digital infrastructure, and communication facilitated a seamless and engaging experience for all participants. We recognise the indispensable assistance of our institutional partners and sponsors who facilitated this event.

This volume aims to function as both a documentation of the conference and a significant resource and source of inspiration for researchers, educators, and practitioners. This collection of papers offers a robust basis for contemplation and initiative, whether you aim to investigate a particular study area, generate ideas for future enquiries, or engage with other academics.

We aim to establish CSIT as a recurring, international event that supports research quality, facilitates multidisciplinary discourse, and cultivates a culture of innovation and transparency. We are assured that CSIT® 2025 marks the inception of a significant and enduring academic legacy.

Thank you for being part of this journey.

On behalf of the Organizing Committee,
CSIT® 2025

Organization

Dr. Amarildo Rista	General Chair
Prof. As. Dr. Anjea Pasku	Chair
Prof. Dr. Xhemal Zenuni	Chair

Keynote Speakers

Prof. Dr. Bekir Karlik	Expert in Artificial Intelligence and Pattern Recognition
Prof. Dr. Abdulhamit Subasi	Expert in Artificial Intelligence and Biomedical Signal Processing

Organizing Committee

Arben Reka	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Bora Myrto	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Eda Tabaku	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Edmira Xhaferra	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Elton Tata	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Fabiana Muharremi	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Fatjon Bushi	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Frida Gjermeni	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Robert Kosova	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Uendi Çerma	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës

Viola Shtiono	Member, Faculty of Information Technology, Aleksandër Moisiu University of Durrës
Arta Demiraj	secretary, Faculty of Information Technology, University Aleksander Moisiu Durres
Brunilda Hoxha	secretary, Faculty of Information Technology, University Aleksander Moisiu Durres

Scientific Committee

Adrian Besimi	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Amarildo Rista	Vice-Dean of the Faculty of Information Technology, Aleksandër Moisiu University of Durrës, Albania
Anita Agárdi	Lecturer at Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary
Artan Luma	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Attila Baksa	Lecturer at Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary
Azir Aliu	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Besnik Selimi	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Blerina Zanaj	Associate Professor at Faculty of Economics and Agribusiness, Agriculture University of Tirana, Tirana, Albania
Bogdan Tudorica	Full Professor at Faculty of Economic Sciences, Petroleum-Gas University of Ploiești, Romania
Carlo Ciulla	Associate Professor at Faculty of Economics, Technology and Innovation, Western Balkans University, Tirana, Albania
Cemil Turan	Associate Professor at Faculty of Engineering and Natural Science, Suleyman Demirel University, Almaty, Kazakistan
Debabrata Samanta	Associate Professor at Rochester Institute of Technology, Pristina, Kosovo
Dragan Pamucar	Full Professor at Faculty of Organizational Sciences, University of Belgrade, Serbia
Eda Tabaku	Lecturer at Faculty of Information Technology, Aleksandër Moisiu University of Durrës, Albania

Edlira Kalemi	Associate Professor at Faculty of Computing, Engineering and the Built Environment, Birmingham City University, England
Eglantina Kalluçi	Full Professor at Faculty of Natural Sciences, University of Tirana, Tirana Albania
Elda Cina	Associate Professor at College of Engineering and Technology American University of the Middle East, Kuwait
Elinda Kajo Meçe	Full Professor at Faculty of Information Technology, Polytechnic University of Tirana, Tirana, Albania
Elira Hoxha	Associate Professor at Department of Statistics and Applied Informatics, Faculty of Economics, University of Tirana, Tirana, Albania
Elma Zanaj	Full Professor at Faculty of Information Technology, Polytechnic University of Tirana, Tirana, Albania
Elva Leka	Lecturer at Polytechnic University of Tirana, Tirana, Albania
Emre Eroglu	Associate Professor at Faculty of Architecture and Engineering, Epoka University, Tirana, Albania
Erika Baksáné Varga	Lecturer at Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary
Erkan İmal	Full professor at CT State Community College Gateway, New Haven, CT, USA
Ermira Idrizi	Lecturer at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Ersin Elbasi	Full Professor at College of Engineering and Technology American University of the Middle East, Kuwait
Evgjeni Xhafaj	Associate Professor at Faculty of Mathematical Engineering and Physical Engineering, Polytechnic University of Tirana, Albania
Fatos Xhafa	Full Professor at Universitat Politècnica de Catalunya, Spain
Florie Ismaili	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Gabriel Xiao-Guang YUE	Full Professor at European University Cyprus,Nicosia, Cyprus
Galia Marinova	Full professor at Faculty of Electronics Engineering and Technology, Technical University of Sofia, Bulgaria
Gjergji Mulla	Associate Professor at Department of Statistics and Applied Informatics, Faculty of Economics, University of Tirana, Tirana, Albania
Ilir Keka	Associate Professor at Faculty of Computer Science, AAB College, Pristina, Kosovo

Jaumin Ajdari	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
José Angel Fernández Prieto	Full Professor at Department of Telecommunications Engineering, University of Jaén, Spain
Jose Francisco Sigut Saavedra	Associate Professor at University of La Laguna, Spain
Jusuf Qarkaxhija	Associate Professor at Faculty of Computer Science, AAB College, Pristina, Kosovo
Karima Boudaoud	Associate Professor at Université de Nice-Sophia Antipolis (UNS), France
Kreshnik Vukatana	Associate Professor at Department of Statistics and Applied Informatics, Faculty of Economics, University of Tirana, Tirana, Albania
Laszlo Kovacs	Full Professor at Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary
Laurik Helshani	Associate Professor at Faculty of Computer Science, AAB College, Pristina, Kosovo
Leonard Barolli	Full Professor at Fukuoka Institute of Technology, Japan
Lindita Mukli	Full Professor at Faculty of Information Technology, Aleksander Moisiu University of Durres, Durres, Albania
Lorena Margo	Associate Professor at Faculty of Natural Sciences and Humanities and Vice-Rector at Fan S. Noli University of Korçë, Korçë, Albania
Lorenc Ekonomi	Full Professor at Faculty of Natural Sciences and Humanities and Rector at Fan S. Noli University of Korçë, Korçë, Albania
Luis Lamani	Associate Professor at Polytechnic University of Tirana,Tirana, Albania
Majlinda Fetaji	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Markela Muça	Associate Professor at Faculty of Natural Sciences, University of Tirana, Tirana Albania
Marko Bajec	Full Professor at University of Ljubljana, Slovenia
Mehmet Emir Koksal	Full Professor at Faculty of Science, Ondokuz Mayis University, Samsun, Turkey
Mentor Hamiti	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Mirela Panait	Full Professor at Faculty of Economic Sciences, Petroleum-Gas University of Ploiesti, Romania

Mohammad Salman	Associate Professor at American University of the Middle East, Kuwait
Monica Landoni	Full Proffesor at Professor at the Faculty of Informatics, University of Italian Switzerland, Lugano, Switzerland
Nuhi Besimi	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Olivér Hornyák	Lecturer at Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary
Onur Günlü	Associate Professor at Electrical Engineering Department,Linköping University, Sweden
Özcan Asilkan	Full Professor at Higher Colleges of Technology, Abu Dhabi, United Arab Emirates
Pardeep Kumar	Associate Professor at Department of Computer Science, University of Warwick, UK
Rafail Prodani	Associate Professor at the Faculty of Natural Sciences and Humanities, Fan S. Noli University of Korça, Korça, Albania
Rinela Kapciu	Associate Professor at Faculty of Information Technology, Aleksandër Moisiu University of Durrës, Albania
Robert Kosova	Associate Professor at Faculty of Information Technology, Aleksandër Moisiu University of Durrës, Albania
Senada Bushati	Lecturer at Faculty of Information Technology, Aleksandër Moisiu University of Durrës, Albania
Szilveszter Kovács	Full Professor at Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary
Tihomir Latinovic	Full Professor at Faculty of Informational Technology, Vitez University, Bosnia and Herzegovina
Valentina Ndou	Associate Professor at the Faculty of Engineering University of Salento, Italy
Valmira Osmanaj	Associate Professor at College of Business Administration, American University of the Middle East, Kuwait
Visar Shehu	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia
Xhemal Zenuni	Full Professor at Faculty of Contemporary Sciences and Technologies, SEEU, Tetovo, North Macedonia

Table of Contents

Section 1: Artificial Intelligence and Machine Learning

1. Age Prediction based on 3D Structural MRI Images using Deep Neural Networks

Stela Lila

International Burch University, Ilidža Sarajevo 71210, BIH
Epoka University, Tirana 1000, ALB
stelalila46@gmail.com

Abstract. The use of structural magnetic resonance imaging (MRI) data has allowed for an in-depth exploration of how the brain experiences age-related neuroanatomical changes. These transformations occur both locally and network-wide as it undergoes maturation and aging. Based on these alterations, minimizing the difference between the real and the chronological age, the so-called brain age delta, necessitates accomplishing precise brain age, which can serve as a biomarker. In recent times, there has been a significant amount of research conducted in the field of age prediction utilizing data from brain MRI scans. A multitude of studies have explored utilizing machine learning algorithms like support vector machines (SVMs), random forests (RFs), and deep learning approaches employing both 2D and 3D imaging modalities. In this paper, we explore age prediction models based on familiar structural networks using convolutional neural networks (CNN) with volumetric data from 1,016 healthy subjects aged 50-98 years (Girona dataset). The model design incorporates preprocessing techniques to standardize the images, including bias correction, registration, brain extraction, and intensity normalization, ensuring consistent input for subsequent analysis. We tested the impact of different architectures on the Girona dataset. Fine-tuning only the prediction block of SFCN architecture achieved the best Mean Absolute Error (MAE) of 3.33 years and r2 coefficient = 0.6713. Overall, more refined results and an increase in prediction metric was obtained when fine-tuning the hyperparameters of the networks.

Keywords: Brain MRI, Age Prediction, Healthy Subjects, Machine Learning, Convolutional Neural Networks.

1 Introduction

Structural MRI studies have shown that the brain undergoes significant neuroanatomical changes with age, particularly in the reduction of gray matter. As humans age, these changes are influenced by a combination of natural aging processes, environmental factors, and neurodegenerative conditions. To assess brain aging, researchers have developed methods to estimate” brain age” [1], comparing an individual's biological brain state to their chronological age. This approach has potential applications in understanding cognitive decline and identifying early markers of neurological disorders.

In recent years, the concept of" brain age" has emerged as a key focus in neuroscience. A key area of exploration has been how functional connectivity in the brain changes with age. These patterns, observed through resting-state functional MRI, offer valuable insights into how brain networks evolve over time. As the brain ages, changes in connectivity can reflect underlying neural mechanisms involved in cognitive decline, memory loss, or even mental health conditions. Understanding these patterns has the potential to reveal important biomarkers for early diagnosis and treatment of disorders such as schizophrenia and Alzheimer's disease.

Recent advances in artificial intelligence, particularly deep learning, have revolutionized brain age prediction. These techniques analyze vast amounts of MRI data, identifying complex patterns and features that may not be detectable through traditional methods [2]. Deep learning models have significantly improved the accuracy of brain age estimation, offering new opportunities to detect subtle signs of aging-related changes in the brain. The potential applications of these advancements extend to early diagnosis of neurodegenerative diseases, monitoring cognitive health over time, and even personalizing interventions to promote healthy aging.

Given the need for a reliable method to accurately predict brain age and its potential application in clinical settings, the primary goal of this research is to develop a robust brain age prediction model. Our study explores two different deep learning architectures. The first is based on the model proposed by [3], with modifications to the final output layer, adapting it for binary classification. The second architecture is the SFCN model, as described by [4], which has shown promising results in brain age prediction. Based on its performance, this paper will focus exclusively on the SFCN model to further develop and optimize its predictive capabilities.

The rest of the paper is as follows: Section 2 represents the literature review coducted in the research field, section 3 presents the implemented research design including data collection and analysis, and methodology of the proposed methods. The experimental results of the algorithm's performance are presented in Section 4. The final concluding remarks together with discussion and conclusions are presented in Section 5.

2 Literature Review

MRI quantification of the brain has been extensively studied due to its clinical importance. Age prediction methods vary based on input data, such as 2D or 3D projections, brain volumes, and gray/white matter maps, as well as the dataset used. Since our focus is on brain volumes, we concentrate on models that use this input. These methods fall into two categories: machine learning and deep learning, both of which are represented in Table I.

Table 1. Comprehensive quantitative analysis across all machine learning methods.

Paper	Methodology	Dataset/s	Mean Absolute Error (in years)
Machine Learning [5]	Ensemble: SVR+DTR	PAC 2019	4.57
Machine Learning [6]	SVR	UK Biobank	3.69
Machine Learning [7]	2D CNN	PAC 2019	4.38
Machine Learning [8]	brainageR	OASIS	5.02
Deep Learning [4]	SFCN	UK Biobank PAC 2019	**2.14** **2.9**
Deep Learning [9]	3D Resnet-18	OpenBHB	3.76
Deep Learning [10]	3D Resnet-34	UK Biobank	2.55
Deep Learning [11]	3D CNN	UK Biobank	2.2
Deep Learning [12]	M-AVAE	OpenBHB	2.77

3 Materials and Methods

3.1 Girona Dataset

The dataset used in this work is Girona dataset. This dataset consists of T1-weighted and T2-weighted images of adults aged 50-98 years. It is important to mention that the individuals participating in this study do not exhibit any signs of cognitive impairment or conditions that affect the functioning or structure of their brains. Initially, the dataset contained 1022 images and after a quality vision check, we ended up working with 1016 subjects which were 610, 204, and 202 were used for training, validation, and testing respectively. In Figure 1, we can observe a graphical representation showcasing the distribution of ages. Most of the cases are centered between the interval of 60 to 70 years old. A few cases are found in the extremes of the distribution. Together with the images, an excel file containing the id of the patients, diet score, age, and sex was provided.

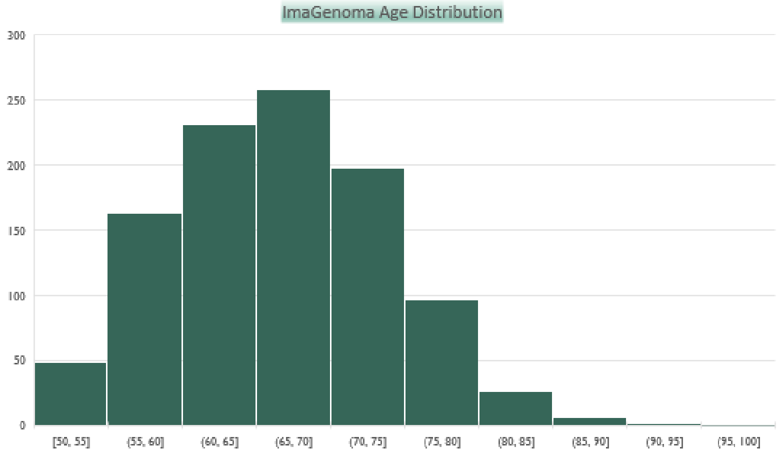


Fig. 1. Girona dataset age distribution.

Given that the images were not preprocessed, the following steps were undertaken to standardize the dataset: (i) applying bias field correction to all images, (ii) performing non-linear registration to MNI atlas, (iii) skull stripping, and (iv) tissue and subcortical structures segmentation. More details about the preprocessing steps can be found in section 3.2.

3.2 Preprocessing

Preprocessing is essential for enhancing the quality and reliability of trained models. It involves techniques like data normalization, scaling, and feature extraction to prepare raw data for optimal model performance. A primary goal is to reduce inter-subject variability, ensuring uniformity across samples, which helps models capture key patterns more effectively. Before implementing the architecture, we analyzed the raw images directly from the scanner and decided to pre- process them to improve model performance. Figure 2 shows the preprocessing steps, including bias field correction, non- deformable registration, brain extraction, and tissue and sub- cortical structure segmentation. This preprocessing was performed in parallel, divided into 40 constraints for time efficiency.

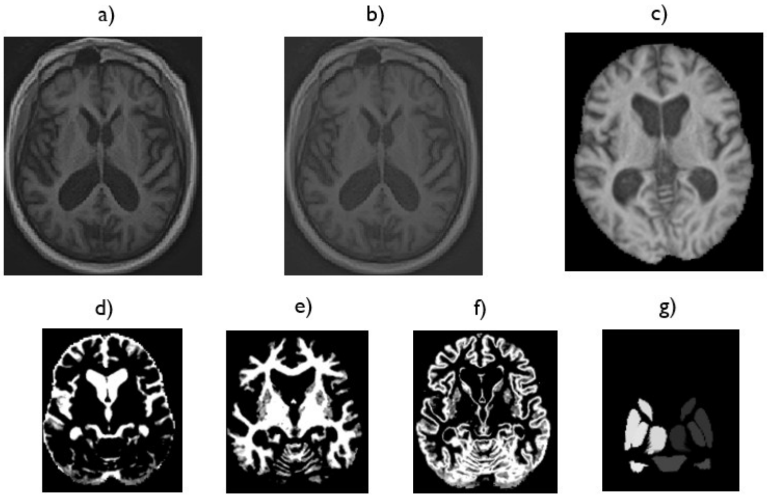


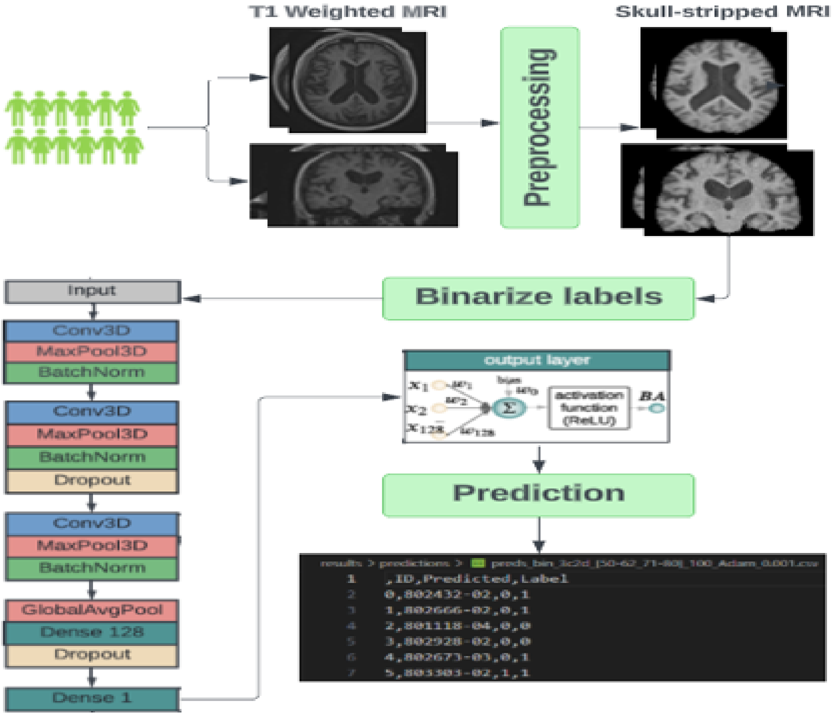
Fig. 2. Visual representation of data preprocessing: a) axial slice of the original T1-weighted, b) bias field corrected, c) skull stripped registered image; d-f) CSF, WM, GM volume and g) subcortical structures.

We applied N4 bias field correction using the method presented by [13] defined in SimpleITK library, which addresses uneven intensity bias in MRI images caused by factors like magnetic field irregularities. Non-deformable registration and skull-stripping were executed using the FSL library [14], which applies non-deformable registration in MNI space for skull stripping, the FAST algorithm for tissue segmentation, and the FIRST method for segmenting subcortical structures. For image registration, we used the FSL pairreg function, which performs affine registration while preserving the skull's scale, ensuring consistent intracranial volume across patients. Although non-linear registration is used for skull stripping, the images are only linearly transformed to the MNI template.

3.3 3c2d Binary Classification Model

We started our deep learning area using a binary classification model. In this study, the training was conducted using a small subset of cases that have extreme values for the score, resulting in the classification of the classes as "low "and "high" scores, assigned with 0 and 1 respectively. We adopted this classification approach as a start-up work because classifying data is generally considered easier than regression.

Architecture Adaption. We adapted the 3D-CNN model from [15] for binary classification, naming it "3c2d" for its three convolutional and two dense layers. The architecture includes three convolutional blocks, a global average pooling layer, and a final dense layer with two output neurons instead of one, modifying the original regression-based design. A full overview of the training flow can be seen in Figure 3.



```
,ID,Predicted,Label
0,802432-02,0,1
1,802666-02,0,1
2,801118-04,0,0
3,802928-02,0,0
4,802673-03,0,1
5,803303-02,1,1
```

Fig. 3. Working flow of the binary classification pipeline.

Training and Testing. The Girona dataset was split to 60/20/20 for training, validation, and testing, ensuring balance. We used SGD and Adam optimizers and focused on age intervals [50, 62] and [71, 80] to avoid imbalance. To fit the classification model, we binarized ages, assigning 0 to the lower bound and 1 to the upper bound when the binarize option was enabled. Different combinations of the parameters were conducted whereas the best resulting one was using MSE loss and SGD optimizer. The learning rate was set to 0.001. A detailed explanation of the results can be found in Section 4.1.

Loss Function. The loss functions implemented are Mean Square Error (MSE) and Cross-entropy (CE). Mean Square Error loss function it is defined as:

$$\text{MSE} = \frac{1}{N} * \sum_{i}^{N} (Y_i - F_{X_i}) \quad (1)$$

where:

-N is the total number of subjects
-Y_i are the true ages
-F_{X_i} represents the neural network that outputs the predicted age

The cross-entropy loss function calculates the average negative log-likelihood of the predicted age class probabilities. It penalizes the model for deviating from the true age labels, encouraging it to learn accurate age predictions. The equation for cross-entropy loss is as follows:

$$CE = -\sum_i^N t_i * \log(Ft_i) \qquad (2)$$

Evaluation Metrics. In the context of the binary classification pipeline used in age prediction accuracy, sensitivity and specificity were used as evaluation metrics. Their respective equations can be found below:

$$ACC = \frac{correct}{nr.of\ patients} \qquad (3)$$

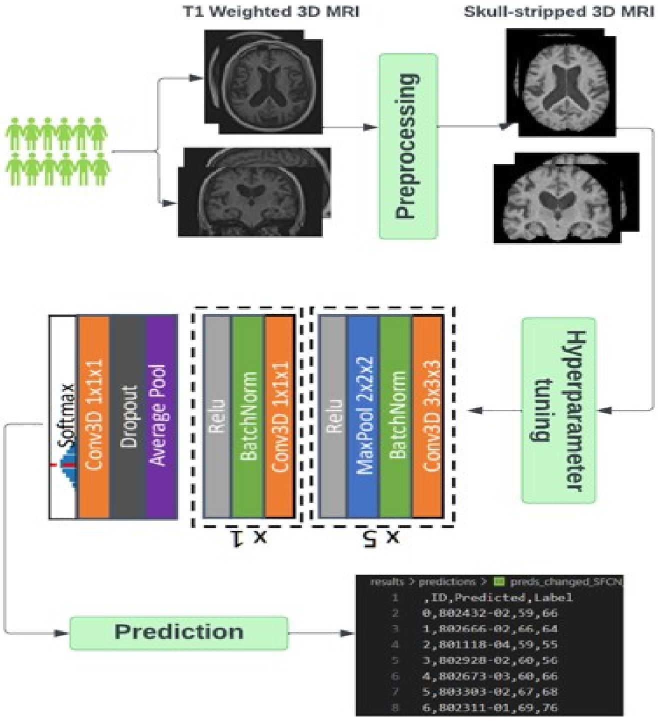
$$SENS = \frac{TP}{TP+FN} \qquad (4)$$

$$SPEC = \frac{TN}{TN+FP} \qquad (5)$$

where:
-TP: P=1 & T=1
-FN: P=0 & T=1
-TN: P=0 & T=0
-FP: P=1 & T=0

3.4 SFCN Regression Model

We implemented a deep learning pipeline for regression, following the best-performing architecture from [4] named the SFCN model. For the authors, SFCN provided the best result (MAE 2.14) in predicting age from MRI images, trained with Uk Biobank [16] data and evaluated in PAC 2019 dataset [17]. We tested this model by adapting their architecture using our dataset, the Girona dataset. The working pipeline of the model can be seen in Figure 4.



```
,ID,Predicted,Label
0,802432-02,59,66
1,802666-02,66,64
2,801118-04,59,55
3,802928-02,60,56
4,802673-03,60,66
5,803303-02,67,68
6,802311-01,69,76
```

Fig. 4. Overview of the deep learning regression pipeline.

Architecture Adaption. To be compatible with the model's requirements we have made some changes to our dataset. Our input data was originally 193x229x193 voxel size. To match the input size, the model is expecting, 2 different approaches were followed:

1) Change the spatial size to 256x256x256 padding with 0s.

2) Cropping the field of view (FOV) to 160x192x160.

Several experiments were conducted for hyper-parameter tuning. These included fine-tuning all layers with pre-saved weights, testing on the test set, fine-tuning only the classifier block, initializing weights using the Xavier technique, and training from scratch. Additionally, we fine-tuned only the batch normalization layers and experimented with freezing and unfreezing blocks. Different values for learning rate, batch size, and optimizer were tested across all experiments to assess their impact on model performance.

Training and Testing. During training, the authors used a Stochastic Gradient Descent (SGD) optimizer, as detailed by [18], to minimize the Kullback-Leibler divergence loss function between the predicted probabilities and a Gaussian distribution. In this distribution, the mean was set to the true age of each training subject, with a fixed standard deviation of 1 year for the UK Biobank dataset and 2 years for the PAC 2019 dataset. The L2 weight decay coefficient was set at 0.001, the batch size was 8, and the learning rate was initialized at 0.01, decreasing by a factor of 0.3 every 30 epochs unless specified otherwise.

Model Output and Loss Function. The loss function employed is the Kullback-Leibler divergence, or relative entropy, which measures the difference between two probability distributions. In the context of age prediction, KL divergence quantifies the dissimilarity between the predicted age distribution and the true age distribution. The equation for Kullback- Leibler divergence between two probability distributions P and Q is given by:

$$KL(P||Q) = \sum \frac{P(x) * \log(P(x))}{Q(x)} \tag{6}$$

where:
- P(x) represents the probability of age x in the true age distribution.
- Q(x) represents the probability of age x in the predicted age distribution.

The predicted probability of a subject's age falling within a one-year interval is represented by 40 digits in the output layer. To obtain the final prediction, each age bin's weighted average is calculated:

$$pred = \sum_{i}^{40} x_i * age_i \tag{7}$$

where:
-x_i stands for the probability for the i^{th} age bin
-age_i stands for the bin center in the age interval

In this approach, the age label is treated as a range rather than a single precise value, represented by a discretized Gaussian probability. Similarly, the model's output is presented as a probability distribution, as illustrated in Figure 5.

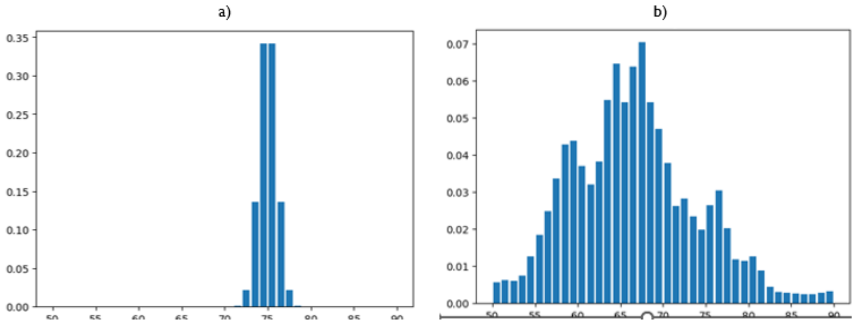


Fig.5. Example of a) soft label and b) prediction distribution for one of the subjects.

Evaluation Metrics. During the evaluation of the deep learning model, we employed the mean absolute error (MAE) and the r-squared (r^2) coefficient as metrics. The MAE quantifies the average absolute difference between the predicted ages and the actual age labels, usually represented in years. The MAE was calculated as:

$$MAE = \frac{absolute\_errors}{nr.of\ patients}$$
(8)

where absolute errors += |P − T |.

In addition, the deep learning model was assessed using the r^2 coefficient, also referred to as the coefficient of determination. This metric indicates the proportion of variance in the dependent variable (age) that is explained by the independent variables (features) in the model. A higher r^2 score implies that the model accounts for a larger share of the variance in age, signifying improved predictive performance. r2 coefficient is calculated as:

$$r2 = 1 - \frac{SSR}{SST} \tag{9}$$

where:

-SSR (Sum of Squared Residuals) is the sum of the squared differences between the true age values and the predicted ones.

-SST (Total Sum of Squares) is the sum of the squared differences between the true age values and their mean.

Figure 6a) refers to an ideal prediction matching the diagonal. Meanwhile in 6b) we can see one of the cases out of our trials.

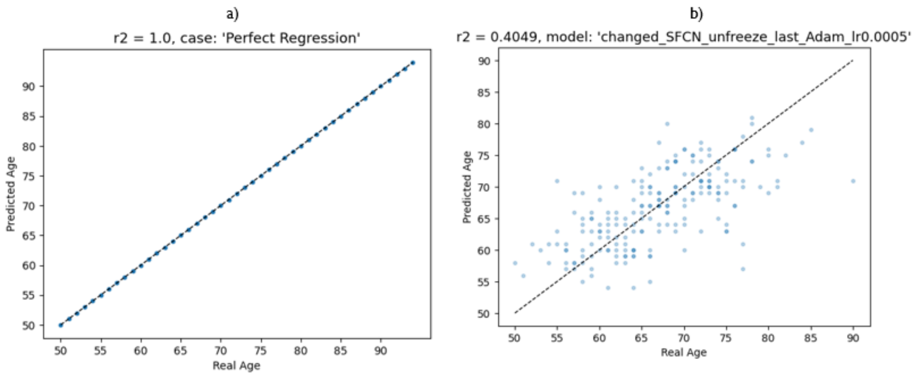


Fig. 6. Different cases of r2 coefficient.

4 Results

4.1 3c2d Model

The first experiment using the 3c2d model was choosing MSE as the loss function and learning rate = 0.01, as the authors suggested in their publication. To check the impact of the optimizer the same experiment but with a different optimizer was repeated. Taking into consideration that SGD gave better results, another experiment with a different learning rate value was conducted. For comparison purposes, CE loss was tried together with Adam optimizer instead of SGD, keeping the same value of the learning rate. A full description of the results and the corresponding parameters can be found in Table 2.

Table 2. The performance of 3c2d model on Girona dataset.

Model (Optimizer)	Epochs	LR	Loss	Batch	ACC	SENS	SPEC
3c2d (Adam)	300	0.01	MSE	8	0.48	0.55	0.34
3c2d (Adam)	100	0.001	CE	4	0.57	0.62	0.29
3c2d (SGD)	300	0.01	MSE	8	0.59	0.63	0.31
3c2d (SGD)	100	0.001	MSE	4	**0.76**	**0.79**	**0.16**

To conclude we can say that the combination of the 3c2d model architecture, SGD optimizer with a learning rate of 0.001, and MSE loss function demonstrated superior performance compared to the other results.

4.2 SFCN Model

Initial Experiments. The architecture was run using a series of experiments, exploring different parameters and methods. A diverse range of methods were utilized, focusing on the input images obtained through padding to size 256x256x256 using SpatialPad() and cropping to 160x192x169 dimensions. A series of initial experiments were run using both input data, like fine tuning all layers using pre-saving weights, inferring directly on test set, finetuning only the last block of the architecture, and training from scratch using Xavier initialization method. The impact of the input data because of padding and cropping (detailed explanation in Section 3.4, in these experiments, can be observed in Table 3.

Table 3. Comparison of the results between the input data type.

Method	SpatialPad()		Cropping	
	MAE (years)	r2	MAE (years)	r2
Finetune all layers using pre-saved weights	5.78	0.0095	5.88	0.0167
Inference on test set	6.19	0.0993	6.57	0.1800
Finetune only the classification block	7.73	0.8619	**5.24**	**0.0770**
Initialize weights using xavier and train from scratch	6.10	0.8207	5.86	0.0168

As shown, the model performed best (MAE = 5.24 years) when only the classifier block was fine- tuned. This highlights the advantage of focusing on fine-tuning the last block, allowing the model to concentrate on learning task-specific features while preserving earlier representations, thereby reducing the risk of overfitting. All experiments were conducted using the SGD optimizer, which yielded the best results for the authors. The span of the predicted values was improved and larger when using cropped data. The number of points passing through the regression line was increased, i.e. the number of correctly predicted values was higher. Therefore, different values of parameters were tried using only the cropped data and the beforementioned architecture. This includes different values of learning rate and optimizers. These results can be found in Table 4.

Table 4. Hyperparameters tuning for finetune only the classification block.

	LR	MAE	R2	Optimizer
Cropping				
	0.001	5.88	0.0167	SGD
	0.001	5.85	0.8255	Adam
	0.0001	5.24	0.0770	SGD
	0.0001	**4.52**	**0.2924**	Adam

Finetuning. Drawing inspiration from Table 3, we identified that the best-performing model was achieved by fine-tuning only the classification (last) block. Based on this finding, we conducted further investigations into the effect of different learning rates using the Adam optimizer. The results of these experiments, as shown in Figure 7, offer valuable insights into how varying learning rates influence the model's performance. The optimal performance was observed with a learning rate of 0.00001, yielding an MAE of 3.33 years and an r² score of 0.6713.

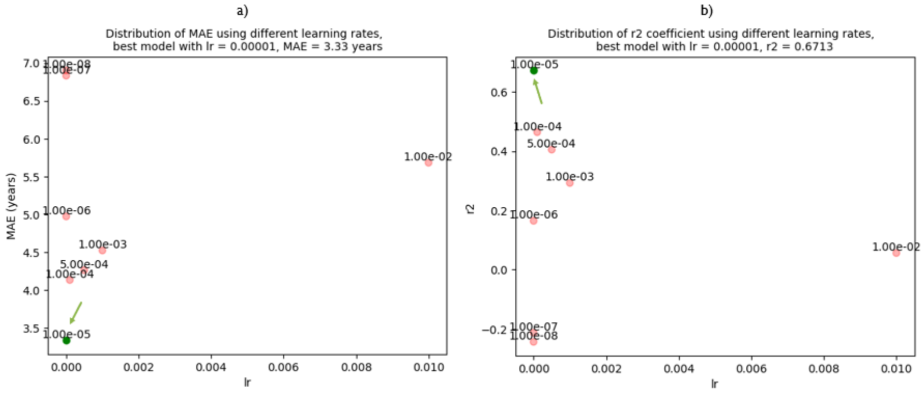


Fig. 7. Distribution of a) MAE and b) r2 using different learning rates, best model with lr = 1e-05, MAE = 3.33 years and r2 = 0.6713.

Experiments were conducted with a batch size of 1 due to MRI volume size and GPU limitations, though a batch size of 3 was tested. However, increasing the batch size worsened performance, likely because diverse samples in larger batches made it harder to extract meaningful patterns. A batch size of 1 allowed the model to focus on individual samples, reducing noise in gradient estimation and improving convergence.

Testing Other Methods. To compare the best result, we explored a recent method inspired by the work of [19], which focused on fine-tuning only the batch normalization layers. Their research demonstrated that selectively adjusting the trainable weights of the batch normalization layers could deliver performance com- parable to fine-tuning all model weights, while also promoting faster convergence. In a separate trial, we implemented a freeze-and-unfreeze strategy.

Initially, we fine-tuned only the classification block of the SFCN model for 10 epochs. After this, the feature extraction block was unfrozen and fine-tuned for an additional 3 epochs. Finally, we re-fine-tuned the classification block for another 10 epochs. The rationale behind this approach was to briefly train the feature extraction block to capture data-specific features while avoiding overfitting and then refocus the model on task-specific features by fine-tuning the classification block again. The results obtained were better with a value of 0.17 and worse with a value of 0.11 for finetuning batch normalization layers and set of freezing and unfreezing approaches respectively. These results are shown in Table 5.

Table 5. MAE and r2 for training only the batch normalization layers and freezing/unfreezing the blocks of SFCN architecture.

Method	MAE(years)	r2
Train batch normalization layers	**4.35**	**0.4128**
Combination of freeze and unfreeze of layers	4.63	0.3281

In summary, among the various trials conducted, the best-performing model was achieved by fine-tuning the last block of the architecture using a learning rate of 0.00001 and Adam as the optimizer. This model resulted in an MAE = 3.33 years and r2 = 0.6713. This configuration yielded superior results compared to other experiments. A detailed overview of the parameter tuning process and the methods employed can be found in Table 6, providing detailed insights into the performance and effectiveness of each approach.

Table 6. The performance of different variations of SFCN model on Girona dataset, epochs = 100, loss = KL-Div, patience = 10.

Method	Learning rate	Optimizer	Batch size	MAE (years)
Finetune all layers using pre-saved weights	0.001	SGD	1	5.88
Inference on test set	0.001	SGD	1	6.57
Finetune only the classification block	0.001	SGD	1	5.24
Initialize weights using Xavier and train from scratch	0.001	SGD	1	5.86
Finetune only the classification block	0.00001	Adam	1	**3.33**

Unfreeze batch normalization layers	0.00001	Adam	1	4.35
Combination of freeze and unfreeze of the layers	0.00001	Adam	1	6.63
Finetune only the classification block	0.00001	Adam	3	5.05

5 Discussion and Conclusions

The models presented in this paper focus on predicting age based on brain MRI scans, particularly emphasizing the change of brain tissues observed in healthy individuals. The modality employed for image acquisition is T1-weighted imaging. By comparing an individual’s predicted brain age to their chronological age, it becomes possible to detect early signs of accelerated or delayed brain aging, indicating potential neurodegenerative conditions or cognitive impairments. Primarily, a binary classification architecture was constructed from scratch with the goal of differencing “high “and “low “values.

By constructing the binary classification architecture from scratch, it was possible to have full control over the model’s architecture, including the choice of layers, activation functions, and connectivity patterns. The age labels were binarized before feeding to the network and the output layers were changed to 2 neurons indicating the binary classification task. Two different loss functions were used, MSE and CE. We also tested the impact of the optimizer on the model by hyperparameter tuning between Adam and SGD optimizer.

The best model resulted in using SGD as an optimizer, a learning rate of 0.001, and MSE as a loss function. Having a learning rate of 0.001 indicates relatively small steps, which can help the model converge gradually and avoid overshooting the optimal parameter values. The loss function and the optimizer also contributed effectively to capturing the discrepancies between the predicted probabilities and the true binary labels.

Secondly, another deep learning model named SFCN was implemented. Fine-tuning all layers, inferencing the test set directly, training from scratch, and fine tuning only the classification block were the first experiments. For these models, we used the parameters suggested by the authors in their paper and used 2 different input data: padding with 0 the input to 256x256x256 and cropping to 160x190x160. The best model turned out to be the one where you fine-tune the last block of the architecture.

After hyperparameter tuning, the best parameters resulted in Adam optimizer with a learning rate of 0.00001 and input data cropped to the model requirements. This resulted in the best model of our experiments. To furthermore investigate and improve the results we did two other trials based on these parameters which were unfreezing and updating the weights of the batch normalization layers only and doing a set of freezing and unfreezing of the layers of the architecture. However, none of them surpassed the performance of the best model.

While the model showcased promising and consistent outcomes, there are still several limitations that could be addressed through future research. It's worth noting that the model encounters challenges and tends to overfit in the age range of 60-70 years due to the imbalanced distribution of the dataset, with most cases falling within that interval. Furthermore, the dataset used for training the model is relatively small in size.

To improve the prediction results, enhancing the registration process and improving skull stripping techniques could potentially enhance the model's performance. In addition, including prior information related to WM and/or GM can improve the pipeline. Introducing data augmentation with appropriate transformations could lead to a more robust method. Further optimization includes fine-tuning the number of encoding/decoding blocks and convolutional layers, as well as incorporating batch normalization techniques. Exploring different architectures can also contribute to achieving improved results. Finally, incorporating multi-modal data fusion, and developing interpretable models would have great significance for better understanding and explainability.

In conclusion, the utilization of MRI-based brain age prediction has yielded promising outcomes in estimating an individual's brain age. This study incorporated two pipelines, specifically a binary classification model and a SFCN (Simple Fully Convolutional Network) model, to forecast brain age based on MRI data. The binary classification model effectively categorized individuals into two groups: younger or older, based on their brain age. Although this approach provided valuable insights regarding relative age differences, it lacked the capability to offer a continuous and more precise estimation of an individual's brain age.

In contrast, the SFCN model showcased better performance in accurately predicting brain age. Notably, the model achieved optimal results by fine-tuning only the classifier block, suggesting that the lower-level feature extraction layers of the SFCN model already captured relevant patterns and representations from the MRI data effectively. By focusing the finetuning process on the classifier block, the model can effectively adapt to specific brain age prediction tasks without sacrificing the learned representations from the lower-level layers. In summary, the integration of MRI-based brain age prediction models, especially the successful implementation of the SFCN model, not only advances our understanding of brain aging but also offers distinct clinical benefits. This technology has the potential to revolutionize clinical practice by providing valuable insights into brain health, aiding early detection, and facilitating targeted interventions to optimize brain function and promote healthy aging.

Acknowledgments

I would like to thank my professors, Xavier Llado and Arnau Oliver, for their invaluable support and continuous guidance throughout the timeline of this work. Their assistance and support have been essential in the successful completion of this research paper. I would also like to express my sincere gratitude to Dželila Mehanović for her valuable insights and encouragement, which have greatly contributed to the development of this work.

References

1. K. Franke, G. Ziegler, S. Koeppel, C. Gaser, and the Alzheimer's Disease Neuroimaging Initiative, "Estimating the age of healthy subjects from t1-weighted mri scans using kernel methods: Exploring the influence of various parameters," Neuroimage, vol. 50, no. 3, pp. 883–892, 2009.
2. B. Couvy-Duchesne, J. Faouzi, B. Martin, E. Thibeau-Sutre, A. Wild, M. Ansart, S. Durleman, D. Dormont, N. Burgos, and O. Colliot, "Ensemble learning of convolutional neural network, support vector machine, and best linear unbiased predictor for brain age prediction: Aramis contribution to the predictive analytics competition 2019 challenge," Frontiers in Psychiatry, vol. 11, p. 593336, 2020Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
3. C. Yin, P. Imms, M. Cheng, A. Amgalan, N. F. Chowdhury, R. J. Massett, N. N. Chaudhari, X. Chen, P. M. Thompson, P. Bogdan et al., "Anatomically interpretable deep learning of brain age captures domain-specific cognitive impairment," Proceedings of the National Academy of Sciences, vol. 120, no. 2, p. e2214634120, 2023.
4. H. Peng, W. Gong, C. F. Beckmann, A. Vedaldi, and S. M. Smith, "Accurate brain age prediction with lightweight deep neural networks," Medical image analysis, vol. 68, p. 101871, 2021.
5. . F. Da Costa, J. Dafflon, and W. H. Pinaya, "Brain- age prediction using shallow machine learning: predictive analytics competition 2019," Frontiers in psychiatry, vol. 11, p. 604478, 2020.
6. Baecker, L., Dafflon, J., Da Costa, P.F., Garcia-Dias, R., Vieira, S., S., Scarpazza, C., Calhoun, V.D., Sato, J.R., Mechelli, A., Pinaya, W.H. Brain age prediction: A comparison between machine learning models using region-and voxel-based morphometric data. Human brain mapping 42, 2332–2346, 2021.
7. Jonemo, J., Akbar, M.U., Kampe, R., Hamilton, J.P., Eklund, A. Efficient brain age prediction from 3d mri volumes using 2d projections. arXiv preprint arXiv:2211.05762, 2022.

8. Bacas, E., Kahhalé, I., Raamana, P. R., Pablo, J. B., Anand, A. S., & Hanson, J. L. Probing multiple algorithms to calculate brain age: Examining reliability, relations with demographics, and predictive power. *Human Brain Mapping*, 44(9), 3481–3492. https://doi.org/10.1002/hbm.26292, (2023).
9. Barbano, C.A., Dufumier, B., Duchesnay, E., Grangetto, M., Gori, P. Contrastive learning for regression in multi-site brain age prediction. arXiv preprint arXiv:2211.08326, 2022.
10. Zhang, B., Zhang, S., Feng, J., Zhang, S. Age-level bias correction in brain age prediction. NeuroImage: Clinical, 103319, 2023.
11. Zhang, X., Bai, B., Li, Y., Zhang, Y., Wang, Y., Zhang, J., Zhang, Y., & Yang, H. Improving brain age prediction with anatomical feature attention-enhanced 3D-CNN. *Computers in Biology and Medicine*, 157, 107873. https://doi.org/10.1016/j.compbiomed.2023.107873, (2024).
12. Usman, M., Rehman, A., Shahid, A., Rehman, A. U., Gho, S.-M., Lee, A., Khan, T. M., & Razzak, I. Multi-Task Adversarial Variational Autoencoder for Estimating Biological Brain Age with Multimodal Neuroimaging. *arXiv preprint arXiv:2411.10100,* (2024).
13. N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, "N4itk: im- proved n3 bias correction," IEEE transactions on medical imaging, vol. 29, no. 6, pp. 1310–1320, 2010.
14. S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney et al., "Advances in functional and structural mr image analysis and implementation as fsl," Neuroimage, vol. 23, pp. S208–S219, 2004.
15. Yin, C., Imms, P., Cheng, M., Amgalan, A., Chowdhury, N.F., Massett, R.J., Chaudhari, N.N., Chen, X., Thompson, P.M., Bogdan, P., et al. Anatomically interpretable deep learning of brain age captures domain-specific cognitive impairment. Proceedings of the National Academy of Sciences 120, e2214634120, 2023.
16. Alfaro-Almagro, F., Jenkinson, M., Bangerter, N.K., Andersson, J.L., Griffanti, L., Douaud, G., Sotiropoulos, S.N., Jbabdi, S.,Hernandez-Fernandez, M., Vallee, E., et al. Image processing and quality control for the first 10,000 brain imaging datasets from uk biobank. Neuroimage 166, 400–424, 2018.
17. Cole, J.H., Franke, K. Predicting age using neuroimaging: innovative brain ageing biomarkers. Trends in neurosciences 40, 681–690, 2017.
18. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in international conference on machine learning. PMLR, pp. 1139–1147, 2013.
19. Kanavati and M. Tsuneki, "Partial transfusion: on the expressive influence of trainable batch norm parameters for transfer learning," in Medical Imaging with Deep Learning. PMLR, pp. 338–353, 2021.

2. Enhancing Software Testing with Generative AI: A Case Study on Test Data Generation

Olivér Hornyák [1]

[1] University of Miskolc, Miskolc, Hungary
oliver.hornyak@uni-miskolc.hu

Abstract. Software testing requires diverse and high-quality data to ensure robust and secure software systems. Generating such data is particularly difficult in sensitive domains like financial fraud detection due to privacy, cost, and availability constraints. This paper investigates the use of Generative AI—specifically Conditional Tabular GANs (CTGANs)—to synthesize realistic financial transaction data for software testing purposes. We evaluate the quality of the generated data through statistical measures and assess its utility for downstream machine learning models. Our results demonstrate that CTGANs can produce statistically and structurally sound synthetic data that is similar to real-world transaction patterns. These findings highlight the growing potential of Generative AI to enhance testing efficiency and model robustness.

Keywords: Generative AI, Test Data Generation, Software Testing, Synthetic Data, Fraud Detection, Machine Learning.

1 Introduction

Software testing plays an important role in ensuring the reliability, performance, and security of modern software systems. Effective testing is based on the availability of high-quality, diverse, and representative test data. This is often expensive, time-consuming, or impractical to generate manually. Traditional approaches to test data creation cannot be accomplished with the growing complexity and scale of software applications. Sometimes, software testing requires domain knowledge to cover realistic data scenarios.

Recent advances in Generative Artificial Intelligence, particularly Large Language Models (LLM) and Generative Neural Networks, offer a promising alternative. These models have demonstrated capabilities in understanding, generating, and manipulating structured and unstructured data. Using generative AI for test data generation introduces new possibilities for automating and enhancing the software testing lifecycle.

A key factor in the effectiveness of software testing is the quality of the test data used. Test data refers to the input data used to execute test cases, and it plays a crucial role in determining how well the software is validated. High-quality test data must be:

- Representative of real-world scenarios, including both typical and edge cases.
- Diverse enough to cover a wide range of input conditions and system behaviors.
- Consistent and valid to ensure tests are meaningful and reproducible.
- Compliant with any domain-specific formats or constraints, such as data privacy regulations.

Poor-quality or insufficient test data can result in tests that miss critical bugs, falsely report errors, or fail to reflect realistic usage patterns. This can ultimately lead to software that appears robust during testing but fails under real-world conditions.

As software systems grow in complexity, often involving large data sets, interconnected components, and dynamic user interactions, the need for automated, scalable, and intelligent approaches to test data generation becomes more urgent. This is where generative AI can offer significant advantages, as it can synthesize realistic, complex, and targeted test data on scale. This paper investigates the effectiveness of generative AI in producing synthetic data for software testing, with a focus on fraud detection in financial transactions. The primary objectives are: (1) to evaluate the use of CTGAN for generating realistic financial transaction data, (2) to assess the quality of synthetic data, and (3) to explore the practical utility of this data in enhancing fraud detection systems.

2 Background and Related Work

Before the emergence of AI-driven techniques, test data generation relied primarily on traditional methods, including manual, rule-based, and random data creation strategies. However, they do not scale effectively in large or complex systems [1]. Rule-based techniques involve the use of formal specifications, business rules, or logical constraints to generate valid test data. Examples include equivalence partitioning, boundary value analysis, and combinatorial testing [2]. Model-based testing, for example, employs behavioral or state-based models of the system under test to generate corresponding test inputs [3], [4]. Random or pseudo-random test data generation involves creating input randomly within specified constraints or data types. This method is simple to implement and can help uncover unanticipated faults through broad input coverage. However, it often suffers from a lack of precision, leading to unrealistic or invalid test data [5].

Search-Based Software Testing – including genetic algorithms, simulated annealing, or hill climbing – uses a predefined search space (i.e., the range of all possible inputs) to satisfy the goal, typically to maximize code coverage, certain path execution, or find input that breaks the application [6]. Maintaining test data consistency over time remains a significant challenge [7].

These limitations motivate the exploration of more adaptive, intelligent methods such as those enabled by generative AI. Machine Learning (ML) has introduced a new level of intelligence and automation in the field of software testing. Supervised learning approaches, for instance, have been used to generate test inputs by learning from labeled datasets [8]. In scenarios where labeled data is available, such models have shown promise in improving the efficiency of test case design [9]. Reinforcement learning is another promising method, where an intelligent agent interacts with the software under test to discover optimal inputs through trial and error. The agent receives feedback in the form of rewards for achieving certain testing objectives, such as increasing code coverage or triggering specific branches [10]. Anomaly detection models can highlight rare or unexpected test behaviors, guiding the creation of edge-case data that may reveal hidden software faults [11].

Generative models have the ability to create new synthetic data sets. Generative Adversarial Network (GAN), originally proposed by [12]. It consists of two neural networks - a generator and a discriminator - which are trained simultaneously in a competitive setting. The generator's role is to produce synthetic data that looks like the real training data, while the discriminator's task is to distinguish between real and synthetic inputs. This combinative training process enables the generator to improve iteratively until the synthetic data becomes nearly identical to the real data [13].

Variational Autoencoders encode input data into a latent space and then decode samples from this space to generate new data [14]. VAEs are particularly effective for generating structured and semantically valid data, making them suitable for tasks such as generating configuration files, structured test cases, or tabular data.

Large Language Models are trained on massive corpora of text data and can generate sequences of natural language or code within certain contexts. Due to their versatility and capacity, LLMs have become increasingly popular in software testing applications. They can generate input strings, SQL queries, JSON data, or even simulate user interactions in a meaningful and context-aware manner [15]. Moreover, LLMs can be fine-tuned for specific domains, enabling the creation of tailored test data without extensive manual effort [16].

In the context of software testing, some studies have investigated the usability of synthetic data for improving test coverage, fault detection, and automation. For example, [17] investigated the use of synthetic data to test data-intensive applications and found that it significantly improved the detection of logical errors and anomalies. Similarly, [18], proposed a hybrid approach combining model-based testing and synthetic data generation to enhance software quality assurance in complex systems.

Another important area of research focuses on privacy-preserving synthetic data in software testing, see [19] or [20]. In their paper, [21] explored synthetic data generation for testing Android applications and found that synthetic user behavior traces could help uncover security vulnerabilities while avoiding exposure to real user information. In the field of fraud detection, [22] highlighted the difficulty of obtaining large-scale fraud datasets and demonstrated how synthetic data augmentation could improve the performance of classifiers for financial fraud detection. GAN-based approaches have also been applied to simulate fraudulent transaction patterns that are statistically similar to real-world fraud, enabling the training of more resilient detection systems [23]. These methods help balance class distributions and introduce realistic variation in transaction data without risking revealing sensitive financial records.

Moreover, there is a growing body of research focusing on combining synthetic data with anomaly detection algorithms to identify subtle or novel fraudulent behaviors. Such hybrid systems benefit from the diversity and variability of synthetic inputs while maintaining high precision and recall rates in fraud detection tasks [24].

Table 1. Comparison of Synthetic Data Use in Software Testing and Fraud Detection.

Aspect	Software Testing	Fraud Detection
Primary Goal	Improve test coverage, simulate realistic and edge-case inputs	Train and evaluate fraud detection models with balanced, labeled data
Key Challenges Addressed	Lack of diverse/edge-case test data Privacy constraints- Scalability of manual test data generation	Shortage of fraud samples Class imbalance Confidentiality of financial data
Common Techniques	Rule-based generation	GANs for synthetic fraud patterns
Data Types Generated	GANs and VAEs, LLMs- Model-based test generation	Data augmentation Anomaly simulation
Use of Domain Knowledge	Input strings, forms, API payloads, and GUI interaction logs	Transaction records, customer profiles, behavioral sequences
Typical Benefits	Required for format rules, constraints, and validation	Required for fraud pattern simulation and feature engineering
Common Evaluation Metrics	Broader test scenario coverage Safer privacy-compliant data- Supports CI/CD testing automation	More robust fraud classifiers Balanced datasets- Improved model generalization
Limitations	Code coverage, fault detection rate, and bug reproducibility	Precision, recall, F1-score, AUC-ROC

Prior research supports the conclusion that synthetic data - when properly generated and validated [25] - can serve as a powerful asset in software testing and fraud detection. It enables broader testing scenarios, improves model generalization, supports compliance with privacy standards, and reduces dependence on costly or restricted real-world data.

3 Generating Synthetic Financial Transactions for Fraud Detection

3.1 Problem statement

Detecting fraudulent financial transactions is a critical and highly complex task in the financial services sector. At the heart of this challenge lies the lack of high-quality, labeled data for training and evaluating fraud detection systems. Fraudulent events are rare by nature, often representing less than 0.1% of all transactions, resulting in highly imbalanced datasets. Moreover, acquiring accurately labeled fraud data is further complicated by the confidential and sensitive nature of financial transactions. Historically labeled data may become outdated, reducing its relevance and effectiveness for training current models. Maintaining up-to-date, diverse, and representative labeled datasets is both resource-intensive and operationally risky. In addition, manual labeling of fraudulent transactions is a time-consuming and error-prone process.

There are a few free public datasets available that are commonly used in academic research:

Table 2. Credit Card Fraud datasets.

Dataset	Type	No of Transactions
Credit Card (Kaggle / ULB, 2025) [26}	Real (anonymized)	284,807
IEEE-CIS (Kaggle, 2025) [27]	Real (complex)	1 million+
PaySim [28]	Simulated	~6 million
BankSim [29]	Simulated	~5,000

Given these constraints, generating synthetic financial transaction data that includes simulated fraudulent behavior presents a promising alternative. Synthetic data allows researchers and practitioners to overcome legal, ethical, and technical barriers while also enabling the creation of balanced datasets that enhance model training and evaluation.

3.2 Generative model selection

Selecting an appropriate generative model was the next step in synthesizing financial transaction data for fraud detection. Generative Adversarial Networks are widely used for synthesizing tabular and structured data due to their ability to model complex, high-dimensional distributions. On the other hand, Large Language Models have recently emerged as powerful tools for generating textual and semi-structured data, including

JSON-based transaction payloads, SQL-like records, and log-style sequences. LLMs are particularly well-suited for simulating human-readable financial narratives, chatbot logs, or detailed descriptions of transaction behavior—useful for developing and testing systems with natural language interfaces or fraud case summaries.

LLMs offer remarkable flexibility and require minimal task-specific training. They can be prompted or fine-tuned to simulate edge cases, imitate fraud patterns, or generate labeled transaction datasets in natural language or code-friendly formats. However, their output may be harder to validate statistically, especially when strict numerical fidelity or business rule compliance is needed. Furthermore, unlike GANs, LLMs may not always produce perfectly structured data without prompt engineering or post-processing.

In this paper, the Conditional Tabular Generative Adversarial Network [30] was applied. The process began by loading and merging the files from the (IEEE-CIS, 2025) dataset using the common TransactionID field. A subset of features was selected, emphasizing categorical variables such as card details, email domains, device information, and identity fields (id_12 through id_38), along with key numerical variables like TransactionAmt (transaction amount), TransactionDT (transaction datetime), and the target label isFraud. Categorical features were preprocessed by filling missing values with a placeholder and transforming string labels into numerical representations using LabelEncoder. After removing samples with missing numerical values, the cleaned dataset was used to train a CTGAN model for 50 epochs.

Following training, the model was used to generate synthetic samples equal in size to the real dataset.

3.3 Evaluating the quality of synthetic data

The quality of synthetic data in software testing depends not only on its ability to represent the important characteristics of real-world data but also on its structural correctness, diversity, and effectiveness in practical testing scenarios. In generative AI-based test data generation, evaluating quality is a multi-faceted process combining statistical analysis, domain-specific constraints, and downstream testing performance. A foundational aspect of evaluation is statistical similarity. Synthetic data should preserve the distributional properties of the original dataset without replicating it exactly. Techniques such as Jensen–Shannon divergence, Kolmogorov–Smirnov tests, and feature correlation analysis are commonly used to compare synthetic and real datasets [30]. These metrics help determine whether the generative model has learned a close representation of the source data distribution.

Equally important is structural and format validity. Generated data must conform to the schema, types, and constraints expected by the software under test. This includes valid date formats, appropriate value ranges, and correct data types. Schema validation tools are often used to automate this check, ensuring that the test system can process synthetic inputs without errors [18]. Diversity is another key indicator of quality. While synthetic data should resemble real data, it should also enhance the variety of test inputs, especially for edge cases and uncommon conditions. A lack of diversity can lead to redundant testing, whereas broad input variation improves code coverage and the likelihood of detecting faults. Researchers have proposed methods such as clustering-based diversity metrics and entropy measures to assess how well synthetic datasets cover the input space [31].

Beyond resemblance and structure, the true value of synthetic data lies in its practical utility. In software testing, this can be evaluated by observing how well the data supports bug detection, regression testing, or classifier training. For instance, if a model trained on synthetic data performs comparably to one trained on real data, then the synthetic data can be considered effective. Performance benchmarking remains one of the most reliable ways to assess real-world quality.

Privacy is also a critical concern, particularly when synthetic data is derived from sensitive or proprietary sources [32]. Even when anonymized, synthetic data must be scrutinized to ensure that it does not reveal or reproduce identifiable elements of the training data. Membership inference attacks and nearest-neighbor similarity tests are among the recommended techniques for detecting potential privacy leakage [33].

In cases where the data includes human-readable components, such as natural language inputs or error messages, human evaluation may be necessary. Expert reviewers can assess whether generated content is semantically meaningful, coherent, and contextually appropriate. While subjective, this type of evaluation adds an essential layer of quality control, especially for systems that rely on language-based inputs.

Taken together, these dimensions offer a comprehensive framework for evaluating synthetic data. A robust evaluation strategy ensures that generative AI contributes meaningfully to the software testing lifecycle without compromising data integrity, diversity, or ethical considerations.

3.4 Dataset description

For the investigation described in this paper, the IEEE-CIS 2025 dataset was used. The dataset consists of records of online transactions, partitioned into training and testing sets. Each transaction is identified by a unique TransactionID. The main transactional features are stored in the train_transaction.csv and test_transaction.csv files, while supplementary identity information is provided in train_identity.csv and test_identity.csv. These files can be joined using TransactionID, although not all transactions have corresponding identity records.

The identity data primarily captures device-related and environmental information through fields such as DeviceType, DeviceInfo, and anonymized identity attributes (id_12 to id_38). These features provide additional context for modeling user behavior and detecting anomalies associated with fraudulent activity.

The dataset consists of 590,540 transactions and includes 394 features, both numerical and categorical. It exhibits a significant class imbalance, with 20,663 fraudulent transactions (3.50%) and 569,877 non-fraudulent transactions (96.50%), resulting in an imbalance ratio of approximately 1:28.

Many categorical variables in the dataset contain a substantial amount of missing data. For example, the R_emaildomain field has over 270,000 missing entries, while the M4 variable is frequently absent and may indicate a tiered match status or specific transaction condition. Additionally, some fields had over 90% missing values.

4 Results

To assess the similarity between real and synthetic data distributions, we performed visual comparisons using histograms and Kernel Density Estimates (KDE) for two key numerical features: transaction amount and datetime. Figure 2 compares the distribution of transaction amounts between the real and synthetic datasets. As shown, both distributions are highly right-skewed, i.e., the mean is greater than the median, with a dense concentration of values below approximately 500 USD and a long tail extending toward higher amounts. The synthetic data closely mirrors the real distribution, capturing the general shape and peak density. However, minor deviations are observable, particularly in the tail regions and in the frequency of low-value transactions. These discrepancies are common in synthetic data generation and may stem from the difficulty of modeling long-tailed distributions precisely.

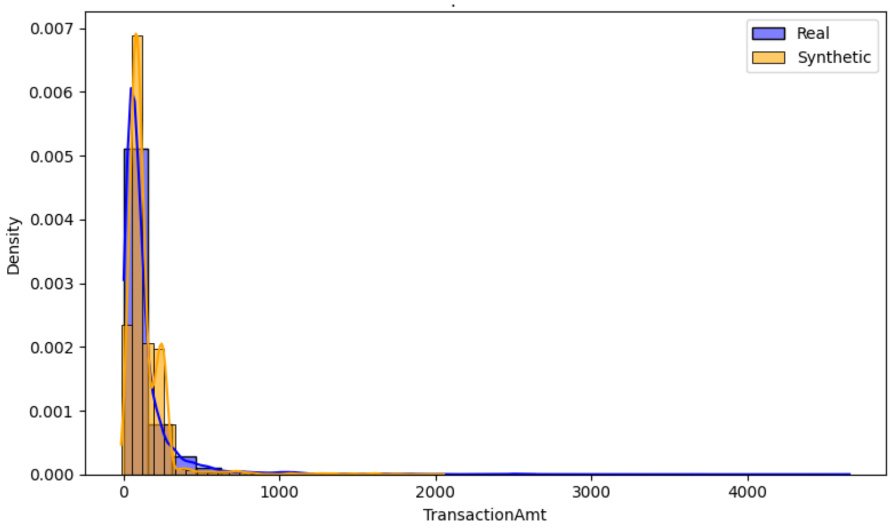


Fig. 1. Transaction Amount.

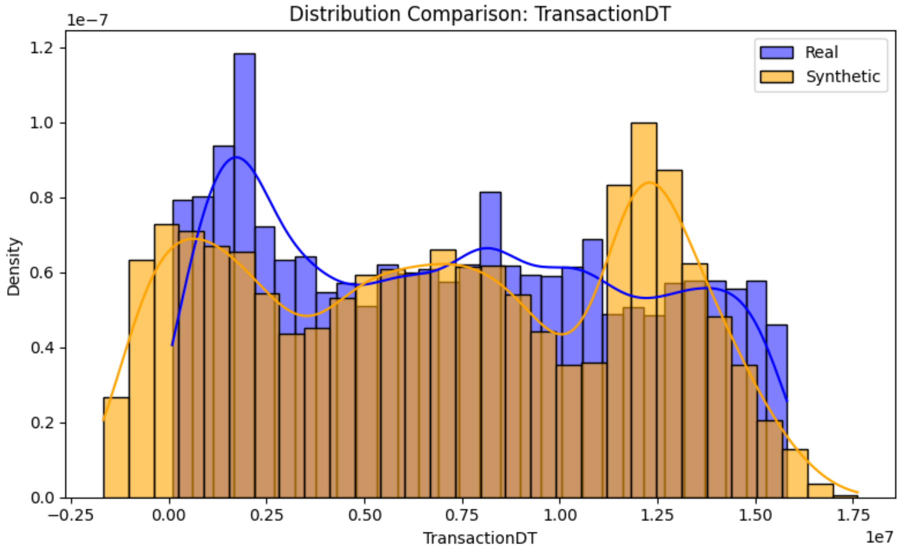


Fig. 2. Distribution comparison.

Figure 2 illustrates the distribution of the transaction's timestamp, which represents the number of seconds since a reference point. The real and synthetic data display similar global patterns, with both showing a spread over the same range and comparable multimodal behavior. Nevertheless, slight differences in the KDE curves indicate that while the CTGAN was able to capture the overall temporal structure, some local variations in the data density were not perfectly reproduced. These may correspond to underrepresented periods or noisy transaction activity in the original dataset.

In addition to visual inspections and Kolmogorov–Smirnov tests, we performed Jensen–Shannon divergence analysis, see Figure 3.

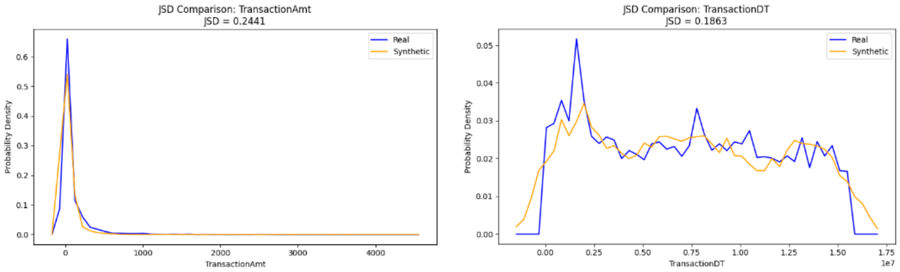


Fig. 3. Jensen–Shannon comparison of transaction amount (on the left) and transaction date (on the right).

JSD is a symmetric and bounded measure (range: 0 to 1), where lower values indicate higher similarity. Both TransactionAmt and TransactionDT produced moderate divergence scores (0.244 and 0.186, respectively), suggesting that the CTGAN model was reasonably effective in approximating the real data distributions, particularly for temporal features.

Overall, the comparisons suggest that the CTGAN was effective in learning the underlying distributions of these continuous variables, providing synthetic data that closely resembles the real data in terms of statistical structure.

5 Conclusions

This study explored the use of generative artificial intelligence for test data generation, focusing on the application of Conditional Tabular Generative Adversarial Networks to synthesize realistic financial transaction data for fraud detection. The research addressed key challenges in acquiring diverse and representative test data, particularly in domains constrained by privacy regulations and data imbalance, such as financial services. By training a CTGAN on a sample from the IEEE-CIS dataset, we demonstrated the feasibility of generating high-fidelity synthetic data that closely approximates the statistical properties of real transactions.

Visual analysis and Kolmogorov–Smirnov tests confirmed that the generated data effectively captured key distributional patterns in both transaction amount and transaction time. These findings suggest that generative AI can be a powerful tool for enhancing test coverage and model robustness, especially in scenarios where traditional data collection methods fall short. Furthermore, synthetic data offers a privacy-preserving alternative to real user data, mitigating legal and ethical concerns while enabling continuous and large-scale testing.

While promising, the approach is not without limitations. Generative models such as CTGAN require careful feature selection, preprocessing, and tuning to ensure valid and useful outputs. The quality of the data generated is also inherently tied to the quality and representativeness of the training data. Moreover, evaluating synthetic data goes beyond statistical similarity - it must also include considerations of structural validity, domain compliance, diversity, and utility in downstream testing tasks.

In conclusion, generative AI represents a paradigm shift in software testing. It enables the automated creation of diverse, context-aware, and privacy-compliant test data, offering a scalable solution to many long-standing challenges in the field. Future work may explore integrating generative data synthesis into CI/CD pipelines, extending to multi-modal datasets, or combining GANs with other AI techniques to support more adaptive and intelligent testing strategies. Future work includes more advanced multivariate similarity tests, such as Maximum Mean Discrepancy and a formal assessment of Principal Component Analysis (PCA) to project high-dimensional data and compare real versus synthetic distributions.

References

1. Afzal, W., Torkar, R., Feldt, R.: A systematic review of search-based testing for non-functional system properties. Information and Software Technology 51(6), 957–976 (2009). https://doi.org/10.1016/j.infsof.2008.12.005
2. Homès, B.: Fundamentals of Software Testing. 2nd edn. Wiley-ISTE, London (2024).
3. Aichernig, B.K., Mostowski, W., Mousavi, M.R., Tappler, M., Taromirad, M.: Model Learning and Model-Based Testing. In: Bartocci, E., Falcone, Y. (eds.) Machine Learning for Dynamic Software Analysis, LNCS, vol. 11026, pp. 49–79. Springer, Cham (2018).
4. Utting, M., Pretschner, A., Legeard, B.: A taxonomy of model-based testing approaches. Software Testing, Verification and Reliability 22(5), 297–312 (2012). https://doi.org/10.1002/stvr.456
5. Gang, L.: Genetic Algorithm and Its Application in Software Test Data Generation. In: 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), pp. 1–6. IEEE, [Location not specified] (2023).
6. Perera, A., Aleti, A., Turhan, B., Boehme, M.: An Experimental Assessment of Using Theoretical Defect Predictors to Guide Search-Based Software Testing. IEEE Trans. Softw. Eng. 49, 131–146 (2023).
7. Dobriban, E.: Consistency of invariance-based randomization tests. *Ann. Stat. 2443-2466,* (2021).
8. Srisakaokul, S., Wu, Z., Astorga, A., Alebiosu, O., Xie, T.:Multiple-Implementation Testing of Supervised Learning Software. AAAI Workshops (2016).
9. Aghababaeyan, Z., Abdellatif, M., Dadkhah, M., Briand, L.C.: DeepGD: A Multi-Objective Black-Box Test Selection Approach for Deep Neural Networks. ACM Trans. Softw. Eng. Methodol. 33, 1–29 (2023).
10. Bai, Q., Jia, Y., Chen, T.Y., Liu, Y.: Adaptive test data generation using reinforcement learning. Information and Software Technology 114, 19–31 (2019). https://doi.org/10.1016/j.infsof.2019.06.003
11. Zhao, N., Chen, J., Yu, Z., Wang, H., Li, J., Qiu, B., Xu, H., Zhang, W., Sui, K., Pei, D.: Identifying Bad Software Changes via Multimodal Anomaly Detection for Online Service Systems. In: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2021), pp. 1–13. ACM, Athens, Greece (2021).
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680. Curran Associates, Red Hook (2014). https://papers.nips.cc/paper_files/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html
13. Shin, H.C., Tenenholtz, N.A., Rogers, J.K., Schwarz, C.G., Senjem, M.L., Gunter, J.L., McMillan, C.: Medical image synthesis for data augmentation and anonymization using

generative adversarial networks. In: Simulation and Synthesis in Medical Imaging, pp. 1–11. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00536-8_1

14. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2014). https://doi.org/10.48550/arXiv.1312.6114
15. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H., Kaplan, J., et al.: Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374 (2021). https://doi.org/10.48550/arXiv.2107.03374
16. Chen, Y., Hu, Z., Zhi, C., Han, J., Deng, S., & Yin, J.: Chatunitest: A framework for LLM-based test generation. In Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering, pp. 572-576. (2024)
17. Shahriar, H., Rahman, M.A., Zulkernine, M.: Using synthetic data to improve software testing of web applications. Journal of Systems and Software 158, 110397 (2019). https://doi.org/10.1016/j.jss.2019.110397
18. Taneja, S., Dey, T.: Hybrid framework for synthetic test data generation in software testing. International Journal of Software Engineering and Knowledge Engineering 28(09), 1315–1333 (2018). https://doi.org/10.1142/S0218194018500544
19. Patki, A., Weller, J.: The Synthetic Data Vault (SDV) Project: Tools for Responsible Data Synthesis. ACM Transactions on Data Science, 3(4) (2023). https://doi.org/10.1145/3557897
20. Khan, S.I., Khan, A.B.A., Hoque, A.S.M.L.: Privacy preserved incremental record linkage. J Big Data 9, 105 (2022). https://doi.org/10.1186/s40537-022-00655-7
21. Agrawal, G., Kaur, A., Myneni, S.: A Review of Generative Models in Generating Synthetic Attack Data for Cybersecurity. *Electronics* 13(2), 322 (2024).
22. Owoade, S.J., Uzoka, A., Akerele, J.I., Ojukwu, P.U.: Automating fraud prevention in credit and debit transactions through intelligent queue systems and regression testing. *Int. J. Frontline Res. Sci. Technol.* 4(1), 45–62 (2024).
23. Fiore, U., De Santis, A., Perla, F., Zanetti, P., Palmieri, F.: Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences 479, 448–455 (2019). https://doi.org/10.1016/j.ins.2018.02.060
24. Jurgovsky, J., Granitzer, G., Ziegler, K., Calabretto, S., Portier, P.E., He-Guelton, L., Caelen, O.: Sequence classification for credit-card fraud detection. Expert Systems with Applications 100, 234–245 (2018). https://doi.org/10.1016/j.eswa.2018.01.037
25. Kiran, A., Saravana Kumar, S.: A methodology and an empirical analysis to determine the most suitable synthetic data generator. IEEE Access 12 (2024): 12209-12228.
26. Credit Card Fraud Detection. Kaggle, https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, last accessed 2025/04/13.
27. IEEE-CIS Fraud Detection Dataset. Kaggle, https://www.kaggle.com/competitions/ieee-fraud-detection, last accessed 2025/04/13.
28. Lopez-Rojas, E.A., Elmir, A., Axelsson, S.: PaySim: A financial mobile money simulator for fraud detection. In: Proceedings of the 28th European Modeling and Simulation Symposium (EMSS), pp. 249–255. Larnaca, Cyprus (2016)
29. Lopez-Rojas, E.A., Axelsson, S.: BankSim: A bank payment simulation for fraud detection research. In: Proceedings of the 26th European Modeling and Simulation Symposium (EMSS), pp. 144–152. Bordeaux, France (2014)

30. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional GAN. In: Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Red Hook (2019).
31. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional GANs. arXiv preprint arXiv:1706.02633 (2017). https://doi.org/10.48550/arXiv.1706.02633
32. Vallevik, V.B., Babic, A., Marshall, S.E., Elvatun, S., Brøgger, H.M., Alagaratnam, S., Bjørrn, E., Veeraragavan, A.K., Befring, N.R., Nygård, J.F.: Can I trust my fake data – a comprehensive quality assessment framework for synthetic tabular data in healthcare. Int. J. Med. Inform. 185, 105413 (2024).
33. Yoon, J., Jarrett, D., van der Schaar, M.: Anonymization through data synthesis using generative adversarial networks (ADS-GAN). IEEE Journal of Biomedical and Health Informatics 24(8), 2378–2388 (2020). https://doi.org/10.1109/JBHI.2020.2991086

3. Personalized Learning in the Age of AI: Adaptive Strategies for Enhancing Student Engagement and Outcomes

Greta Ahma[1] and Arbana Kadriu[2]

[1] Faculty of Contemporary Sciences and Technologies, South East European University,
Tetovo, North Macedonia
a.kadriu@seeu.edu.mk

Abstract. With the rapid advancement of artificial intelligence (AI), personalized learning has emerged as a transformative approach to education, offering adaptive strategies that cater to individual student needs. This paper presents a survey of AI-driven adaptive learning techniques aimed at enhancing student engagement and improving learning outcomes. We examine key AI methodologies, including machine learning-based recommendation systems, intelligent tutoring systems, and real-time feedback mechanisms. Additionally, we explore the role of gamification in AI-powered education, investigating how adaptive game-based strategies can foster motivation and learning. Through a review of recent studies, we identify the benefits, challenges, and future research directions in AI-driven personalized learning. The findings highlight the potential of AI to create dynamic, student-centered learning experiences while emphasizing the need for ethical considerations, data privacy, and equitable access. This survey serves as a foundation for educators and researchers to harness AI's capabilities in shaping the future of education.

Keywords: AI, personalized learning, adaptive learning, student, recommendation systems, gamification, motivation, education

1 Introduction

The rapid evolution of Artificial Intelligence (AI) is transforming educational landscapes, introducing innovative approaches to personalize learning experiences and enhance student engagement. Traditional one-size-fits-all educational models often fail to accommodate diverse learning needs, leading to disengagement and suboptimal outcomes. AI-driven personalized learning presents a promising solution by tailoring instruction to individual students' abilities, preferences, and learning styles, thereby fostering a more inclusive and effective educational environment [12]. However, the integration of AI in education also raises concerns related to data privacy, algorithmic biases, and ethical implications, necessitating a balanced approach to its adoption [11].

Traditional educational models have long struggled with issues such as rigid curricula, limited adaptability to individual learning paces, and inefficient assessment methods. These limitations contribute to disparities in learning outcomes and hinder student motivation [13]. AI-powered educational tools, including adaptive learning pathways and intelligent tutoring systems, offer a viable alternative by dynamically adjusting content delivery based on student performance and engagement metrics [19]. Moreover, generative AI technologies, such as ChatGPT, are revolutionizing academic interactions by providing real-time feedback and facilitating student-centered learning experiences [14].

This paper explores AI-driven adaptive learning strategies aimed at enhancing student engagement and academic outcomes. It examines key aspects such as machine learning-based personalization, intelligent assessment tools, and AI-driven student support systems. Additionally, it discusses the challenges and ethical considerations associated with AI integration in education, including concerns over data security, assessment integrity, and the evolving role of educators [16]. By analyzing both the potential benefits and limitations, this study aims to provide insights into the future of personalized learning in the age of AI.

As AI continues to reshape the educational paradigm, it is essential to strike a balance between leveraging technological advancements and maintaining ethical safeguards. AI-powered learning platforms, such as Duolingo's Birdbrain system, exemplify the effectiveness of adaptive learning by customizing lesson plans to learners' progress and behaviors [20]. However, ensuring that these systems uphold fairness, inclusivity, and educational integrity remains a critical challenge. This paper contributes to ongoing discussions on AI's role in education by evaluating adaptive strategies that can maximize student engagement while addressing ethical and pedagogical concerns. Through this analysis, it aims to provide a roadmap for the responsible implementation of AI-driven personalized learning.

The remaining sections are structured as follows: Section 2 explores the context and motivation for AI-driven personalized learning. Section 3 discusses the background and theoretical foundations of personalized learning and AI in education. Section 5 presents a taxonomy of adaptive strategies in personalized learning. Section 6 surveys existing works on AI models, algorithms, and adaptive strategies. Section 7 addresses challenges and open issues, including technical, ethical, and pedagogical concerns. Section 8 outlines future research directions. Section 9 concludes with key findings and contributions.

2 Background & Theoretical Foundations

2.1 Personalized Learning Principles

Personalized learning is an educational approach that tailors instruction to meet individual students’ needs, preferences, and learning paces. This approach has gained momentum in the digital age, where AI-driven tools enhance adaptive learning experiences. Education 4.0 is described as an evolution toward more learner-centered methodologies, with personalized learning being pivotal [4]. AI-powered platforms leverage student data to provide customized learning paths, improving engagement and learning outcomes [10].

Adaptive learning, a core component of personalized learning, employs real-time data analytics to adjust instructional content. The role of AI in personalizing education through assessment, evaluation, and real-time feedback is well established [13]. This adaptability fosters self-paced learning, allowing students to advance based on competency rather than rigid curricula. Furthermore, generative AI applications expand personalization by providing dynamic content generation, tutoring, and automated feedback mechanisms [9].

Adaptive learning pathways enhance personalization in online education by leveraging machine learning, natural language processing (NLP), and data mining [19]. Research demonstrates the impact of personalized learning systems in improving student engagement, satisfaction, and performance. Duolingo also exemplifies AI-driven personalized learning through its Birdbrain system, which optimizes learning experiences based on student interactions and progress, further showcasing AI’s potential in tailoring educational content [20].

2.2 Role of AI in Education

AI plays a crucial role in education by enabling adaptive learning systems, intelligent tutoring, and data-driven decision-making. Machine learning algorithms analyze student interactions, identifying strengths and weaknesses to personalize learning trajectories. AI-powered language learning systems that use natural language processing (NLP) and intelligent tutoring enhance literacy and digital competencies [10]. These technologies enable instant feedback and tailored exercises, promoting deeper understanding and skill mastery.

Generative AI, such as GPT models, further enhances personalization by generating customized learning materials and facilitating interactive learning environments. These models provide intelligent feedback, automated tutoring, and content adaptation, improving learning efficiency [15]. However, challenges such as ethical concerns, data privacy, and standardization must be addressed to maximize AI’s educational potential.

Additionally, AI-driven learning analytics support educators in refining pedagogical strategies. Castelli and Manzoni [6] explore the role of machine learning in predictive analytics, helping educators anticipate student needs and intervene proactively. AI-powered adaptive assessments ensure continuous monitoring of progress, allowing personalized recommendations for improvement [13].

Furthermore, AI-driven gamification elements enhance student engagement and motivation. As discussed by Landers and Armstrong [2], AI can tailor gamified learning experiences by dynamically adjusting challenge levels, providing real-time feedback, and offering personalized incentives. Vijayalakshmi et al. [18] demonstrate how AI and IoT-driven smart education models revolutionize digital learning by improving adaptability and student engagement through personalized instruction. Similarly, Sandu et al. [17] propose a GenAI-empowered curriculum framework, emphasizing AI's role in curriculum development and learning resource optimization in higher education.

As shown in Figure 1, the system architecture of AI in personalized learning highlights the key components and interactions, including recommendation systems, intelligent tutoring, and real-time feedback mechanisms.

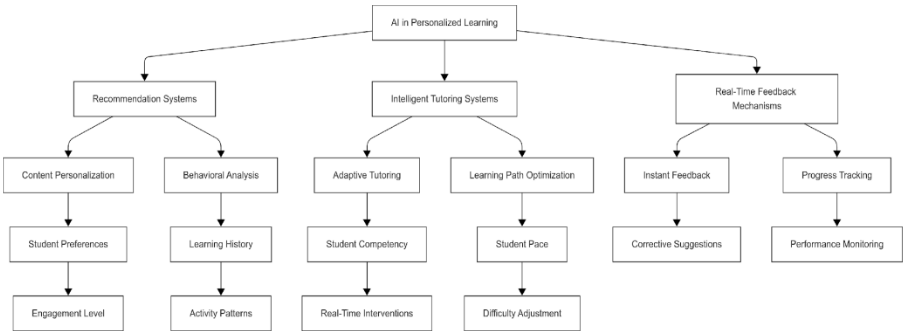


Fig. 1. System Architecture of AI in Personalized Learning: Components and Interactions.

3 Methodology

This survey employs a systematic approach to identify, analyze, and categorize relevant literature on personalized learning powered by artificial intelligence (AI) to ensure academic rigor and clarity.

3.1 Literature Selection

A comprehensive literature search was conducted across multiple academic databases, including IEEE Xplore, Scopus, and Google Scholar. The search focused on publications from 2017 to 2024 using keywords such as \textit{personalized learning}, \textit{AI in education}, \textit{adaptive learning strategies}, and \textit{student engagement}. Inclusion criteria were limited to peer-reviewed journal articles and conference papers that address AI-driven personalized learning systems and adaptive engagement techniques. Non-English language articles, non-peer-reviewed sources, and papers unrelated to educational applications of AI were excluded.

3.2 Literature Analysis

Selected papers were systematically reviewed to extract key themes and insights. This included the identification of AI techniques employed (e.g., generative models like GPT, machine learning algorithms), strategies for enhancing student engagement (such as gamification and adaptive feedback mechanisms), and considerations related to ethics and privacy in AI-enabled learning environments. A comparative evaluation was conducted to assess the effectiveness of different adaptive learning approaches and highlight emerging trends and research gaps.

3.3 Categorization

For clarity and structured synthesis, the reviewed literature was organized into three primary thematic categories: (1) AI technologies facilitating personalized learning, (2) adaptive strategies aimed at improving student engagement, and (3) challenges and ethical considerations surrounding AI adoption in education. This categorization supports a focused discussion on how AI can enhance learning outcomes and informs recommendations for future research and practice.

4 Taxonomy of Adaptive Strategies in Personalized Learning

AI-powered personalized learning employs a variety of adaptive strategies to optimize student engagement and learning outcomes. These strategies can be categorized based on three key dimensions: adaptation target, AI techniques, and learner profile data. This taxonomy provides a structured approach to understanding how AI-driven methods facilitate individualized learning experiences. The first dimension, adaptation target, focuses on what aspect of the learning process is adjusted to fit individual student needs. Content adaptation involves AI systems modifying instructional materials based on student performance and preferences, where adaptive learning platforms utilize student data to recommend personalized resources and exercises [24]. Feedback adaptation refers to AI-powered intelligent tutoring systems providing targeted feedback in real-time, helping students address weaknesses and reinforcing learning through automated guidance [22]. Pacing adaptation enables AI to personalize the speed of instruction, allowing students to progress at their own pace by analyzing learning patterns and adjusting the difficulty level and sequence of lessons [24].

The second dimension, based on AI techniques, categorizes adaptive strategies according to the artificial intelligence methodologies that drive personalized learning. Supervised learning relies on machine learning models trained on labeled educational data to predict student performance and suggest individualized learning paths ([22]. Reinforcement learning allows AI systems to dynamically adjust content difficulty and provide real-time interventions based on student interactions, optimizing engagement and retention [24]. Natural Language Processing (NLP) enhances AI-powered tutoring systems by enabling instant feedback, generating customized explanations, and facilitating language-based learning experiences [22].

The final dimension, learner profile data, examines how AI personalizes learning experiences using various data points. AI considers students' demographics and background, including educational history, language proficiency, and prior knowledge, to tailor instruction accordingly. Learning behavior and preferences are also analyzed, with AI tracking interaction patterns, engagement levels, and preferred learning modalities to enhance content recommendations [24]. Additionally, performance and progress data are continuously assessed, enabling AI to adjust difficulty levels and recommend targeted interventions [22]. By classifying adaptive strategies based on these dimensions, educators and developers can design more effective AI-powered personalized learning environments that cater to diverse student needs, ultimately improving engagement and learning outcomes.

4.1 Proposed Integrative Framework for AI-Powered Adaptive Learning

Current AI-powered adaptive learning strategies typically address individual dimensions such as adaptation targets, AI techniques, or learner profile data in isolation, thereby limiting the scope for comprehensive and context-sensitive personalization. Addressing this fragmentation, we propose an Integrative Adaptive Learning Framework (IALF) that systematically synthesizes multi-dimensional adaptation with real-time feedback mechanisms and contextual awareness to deliver dynamic, learner-centric personalization (see Fig 2).

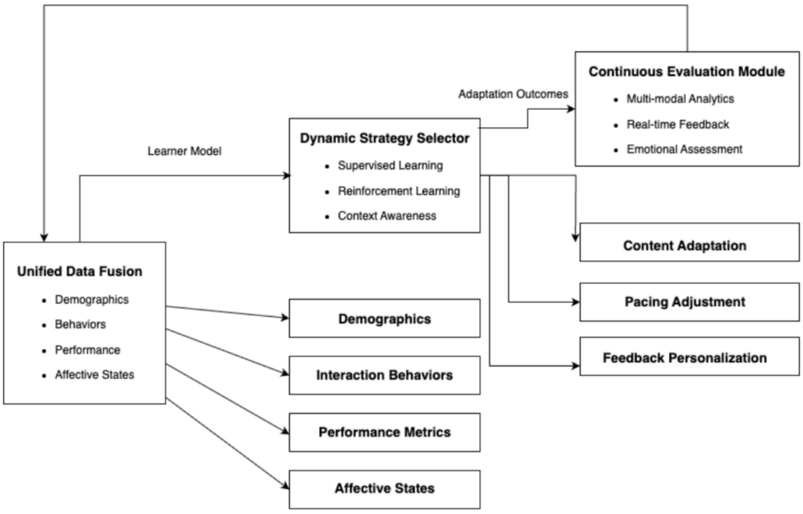


Fig. 2. IALF Framework Diagram.

The IALF is conceptualized around three interrelated components. The first is \textbf{Unified Data Fusion} — a robust mechanism to aggregate heterogeneous learner data—including demographics, interaction behaviors, performance metrics, and affective states—into an evolving learner model. This integration enables a nuanced representation of learner needs and states over time.

The second component is the \textbf{Dynamic Strategy Selector} — a meta-adaptive AI agent leveraging both supervised and reinforcement learning techniques to dynamically select and transition between adaptation targets (content, pacing, feedback) and adaptation methods. Selection is informed by contextual variables such as learner motivation, cognitive load, and environmental factors, facilitating a personalized and context-aware learning experience [3].

The third component is the \textbf{Continuous Evaluation Module} — a comprehensive assessment system that continuously monitors and evaluates the effectiveness of adaptive interventions. It employs multi-modal data analytics to inform iterative optimization of the adaptive process, including the assessment of learners' emotional states and adjustment of micro-break activities to enhance learning performance and engagement [5].

This modular framework is designed for implementation within AI platforms that integrate natural language processing, sensor data interpretation, and advanced machine learning methods to support adaptive pacing, content customization, and feedback personalization tailored to moment-to-moment learner requirements.

Pilot Study Design for Framework Validation. To empirically evaluate the efficacy of IALF, a prototype system will be developed and deployed in a controlled educational environment with a representative learner cohort. The system will collect multi-modal engagement and performance data, as well as self-reported affective measures, to inform the dynamic strategy selector's real-time adaptation decisions. Evaluation metrics will encompass learner engagement, knowledge acquisition, and user satisfaction, analyzed through both quantitative measures and qualitative feedback.

5 Survey of Existing Works

5.1 AI Models and Algorithms in Personalized Learning

Various AI models and algorithms play a crucial role in enhancing personalized learning experiences by enabling systems to adapt to the unique needs of each learner. These approaches range from traditional rule-based systems, which follow predefined logic, to more sophisticated techniques such as machine learning and deep learning, which can analyze vast datasets and make predictions for dynamic adaptation.

Each model brings its own strengths, from increasing engagement through gamification and reinforcement learning to improving the precision of content recommendations using recommender systems. These technologies facilitate real-time feedback, allowing for timely interventions that promote learner autonomy and growth. Table 1 provides an overview of key AI models and their applications in personalized learning environments, showcasing their diverse capabilities and the significant potential they hold for shaping the future of education.

Table 1. AI Models and Algorithms in Personalized Learning.

AI Model/Algorithm	Description	References
Rule-Based Systems	Rule-based systems apply predefined sets of rules to determine learning pathways. These systems can adjust content based on student input, though their flexibility is limited compared to machine learning models.	Vijayalakshmi et al. [18] discuss an AI SVM method that adapts to different student needs using rules based on data from IoT devices, improving engagement, satisfaction, and performance.
Machine Learning	Machine learning algorithms, including both supervised and unsupervised learning, predict students' performance, personalize	Al Balushi and Al Harthi [19] explore machine learning algorithms, such as data mining and NLP, to develop

	content, and optimize learning. These methods enable systems to learn from data patterns and adapt continuously.	personalized learning pathways in online education platforms. AI’s role in personalizing learning pathways is highlighted, leveraging algorithms to tailor content and resources [24].
Reinforcement Learning	Reinforcement learning enables adaptive learning by rewarding desired outcomes and adjusting strategies based on feedback. This method is particularly effective in creating dynamic, engaging learning experiences.	Duolingo's Birdbrain model [20] utilizes reinforcement learning to personalize lesson difficulty and adapt content based on real-time user engagement. Randieri [23] emphasizes the potential of reinforcement learning in AI systems to enhance personalized learning outcomes.
Deep Learning	Deep learning techniques use complex neural networks to handle large volumes of data, uncovering intricate patterns that support highly personalized and predictive learning models. Deep learning is increasingly applied to develop intelligent tutoring systems.	Duolingo’s Birdbrain system [20] is a prime example of deep learning in action, analyzing user performance to predict optimal lesson difficulty and ensuring personalized language learning. Vijayalakshmi et al. [18] employ AI-driven deep learning algorithms for personalized educational experiences

5.2 Adaptive Strategies for Student Engagement

Personalized learning experiences driven by AI have transformed student engagement by dynamically adjusting content, feedback, and recommendations to fit individual learning needs. Three key adaptive strategies—gamification, recommender systems, and real-time feedback mechanisms—enhance motivation, participation, and learning outcomes, as shown in Table 2.

Table 2. Adaptive Strategies for Student Engagement.

Gamification	Recommender Systems	Real-time Feedback Mechanisms
Gamification	The integration of game mechanics such as points, leaderboards, and interactive challenges fosters student engagement. AI-driven gamification systems analyze learner behavior to personalize game elements, ensuring optimal motivation and retention.	[20], [10].
Recommender Systems	AI-based recommendation algorithms analyze past learning behaviors and preferences to suggest personalized content. These systems improve learning efficiency by ensuring students receive materials aligned with their current proficiency and interests.	[14],[19].
Real-time Feedback Mechanisms	AI-powered tools provide instant feedback on student performance, helping learners correct mistakes and improve comprehension in real-time. These mechanisms enhance adaptive learning by allowing timely interventions and support.	[1], [9] .

Gamification, recommender systems, and real-time feedback collectively contribute to a more engaging and effective learning experience. As AI in education continues to evolve, these adaptive strategies will play a crucial role in shaping personalized and interactive learning environments.

5.3 Systems and Platforms

With the advancement of AI in education, several platforms have emerged that leverage adaptive learning technologies to enhance student engagement and outcomes. Among them, Knewton, Duolingo, and Coursera employ AI-driven personalization to tailor learning experiences.

Table 3. Overview of AI-Driven Learning Platforms.

Platform	Methodology	Application	Pros	Cons
Duolingo	Machine Learning (Birdbrain AI), Gamification, Spaced Repetition	Language Learning	Personalized difficulty adaptation, High engagement via gamification, AI-driven feedback	Limited scope beyond languages, Over-reliance on gamification
Knewton	Adaptive Learning Engine, Predictive Analytics	K-12 and Higher Education	Real-time content adaptation, Data-driven recommendatio ns, Customization for individual students	Requires substantial data input, Implementation complexity
Coursera	AI-Powered Personalization, Skill Tracking, Recommender System	Online Higher Education	Course recommendatio ns based on learning patterns, Real-time feedback, Scalability	Lack of deep individualized interaction, Dependency on self-discipline

The comparison of these platforms highlights the role of AI in education by improving engagement and learning outcomes. Duolingo effectively integrates gamification and AI to maintain motivation, while Knewton provides highly adaptive curriculum recommendations based on student performance. Coursera, on the other hand, leverages AI to offer scalable, personalized course suggestions.

These systems demonstrate how AI can bridge learning gaps, yet they also present challenges such as data dependency and implementation costs. Future research should explore hybrid models that combine the strengths of multiple platforms to optimize adaptive learning strategies [9].

6 Challenges and Open Issues

The integration of Artificial Intelligence (AI) into personalized learning offers transformative potential, but it also brings forth several challenges that must be addressed to ensure effective and ethical implementation. These challenges span technical, ethical, and pedagogical concerns, as well as gaps in current research that need to be addressed for AI to reach its full potential in enhancing education.

One of the foremost technical challenges in AI-driven personalized learning is scalability, data sparsity, and the cold-start problem. AI systems rely on vast amounts of data to create personalized learning experiences, yet collecting comprehensive data across diverse educational environments is often difficult. A significant hurdle is the cold-start problem, where AI models struggle to make accurate recommendations for new learners or content due to insufficient historical data [7]. This issue is particularly evident in adaptive learning platforms like Duolingo and Claned, which require extensive data to tailor content effectively. As the number of users increases, managing and analyzing this data becomes more complex. Furthermore, scalability is crucial, as AI systems must process large datasets and serve numerous users simultaneously without compromising performance [19]. Another pressing concern is the lack of interpretability in AI models, which can hinder educators from understanding AI-driven decisions, thereby impacting trust and adoption [15].

Beyond technical concerns, ethical issues present another major challenge. AI-driven personalized learning depends on vast amounts of student data, raising privacy concerns regarding data collection, storage, and potential misuse [12]. Unauthorized access or mishandling of this data could lead to breaches of student privacy. Additionally, bias in AI algorithms can reinforce existing educational inequalities, as AI models trained on non-representative datasets may produce biased outcomes, disproportionately affecting marginalized student groups [22]. Ensuring fairness in AI-driven education is crucial, necessitating efforts to develop systems that equitably serve all learners, regardless of background or learning style [14]. Moreover, the potential for AI to exacerbate the digital divide, requiring students to have access to advanced technology and internet connectivity, underscores the importance of equitable implementation [23].

From a pedagogical perspective, the increasing reliance on AI in education raises concerns about its impact on human instructors and student learning experiences. While AI can enhance personalized learning, it may also reduce opportunities for social learning and emotional support, which are essential components of education [21]. The automation of grading and feedback, for example, could diminish the nuanced, empathetic guidance that educators provide to students [16]. Additionally, AI-driven platforms such as Duolingo's Birdbrain focus on tailoring content to students' existing knowledge, which, while beneficial, may limit exposure to broader educational objectives and interdisciplinary learning opportunities. Over-reliance on AI-based recommendations could also stifle creativity and critical thinking if students are not encouraged to explore beyond algorithmically suggested content [8]. Furthermore, AI tools may inadvertently widen educational disparities, particularly in under-resourced areas where access to advanced technology is limited, thus restricting equal learning opportunities [17].

Despite these challenges, AI in personalized learning shows great promise, yet significant research gaps remain. Much of the existing research focuses on isolated applications of AI, rather than exploring holistic integration across curricula and teaching methodologies. There is also a lack of empirical evidence regarding the long-term effectiveness of AI-driven personalization in diverse educational settings and for students of different age groups. Future studies should evaluate AI's impact on learning outcomes across various contexts to establish best practices [19]. Additionally, more research is needed on the ethical implications of AI, particularly in developing guidelines that ensure responsible and inclusive AI deployment [23]. Another crucial area for future research is improving AI interpretability, allowing educators to comprehend AI decision-making processes and make informed choices about their implementation [15]. Moreover, exploring AI literacy programs for both educators and students will be essential in fostering responsible and informed engagement with AI technologies [9].

By addressing these technical, ethical, and pedagogical challenges, and by bridging existing research gaps, AI can be more effectively integrated into personalized learning. Ensuring responsible development and deployment of AI-driven educational tools will be key to harnessing their full potential while mitigating risks, ultimately enhancing the learning experience for all students.

7 Future Research Directions

As the field of AI-driven personalized learning continues to evolve, several key research directions must be pursued to enhance the effectiveness, equity, and scalability of AI applications in education. Despite the considerable progress made, challenges such as data privacy, algorithmic fairness, and pedagogical integrity must be addressed to ensure that AI truly transforms educational experiences in a meaningful way.

One of the significant challenges in AI-driven personalized learning is the lack of interpretability in AI models. The "black-box" nature of many AI systems makes it difficult for educators to understand how decisions are made, which can undermine trust in these technologies [15]. Future research should focus on developing explainable AI (XAI) models that provide transparent decision-making processes, making it easier for educators to use AI tools confidently and ethically. This will also help address concerns related to algorithmic bias, ensuring that AI systems serve all students equitably [14].

The integration of AI in education raises critical ethical questions surrounding data privacy, algorithmic bias, and fairness. As AI-driven learning systems rely on vast amounts of personal student data, protecting privacy is paramount [12]. Additionally, AI models may inadvertently perpetuate biases, particularly when trained on non-representative data, leading to inequitable learning outcomes [22]. Future research should examine the development of ethical frameworks for AI use in education, with an emphasis on ensuring that these systems are designed to be fair, inclusive, and transparent.

Furthermore, research on ensuring equitable access to AI-powered educational tools, particularly in under-resourced regions, is critical to avoid exacerbating the digital divide [17]. While short-term studies on AI in education have shown promising results, there is a lack of empirical evidence on the long-term effects of AI personalization on student outcomes across different educational contexts [19]. Future research should aim to evaluate how AI-driven personalized learning affects various aspects of student development, including cognitive skills, critical thinking, and emotional well-being, over extended periods. Additionally, research should focus on the effectiveness of AI in diverse educational settings and for different age groups to identify best practices for implementing AI systems across the educational spectrum [13].

AI's potential in education extends beyond content personalization. However, there is a growing concern that over-reliance on AI could marginalize the role of educators and reduce the social and emotional aspects of learning [16]. Future research should explore how AI can complement traditional teaching methods rather than replace them, focusing on hybrid models that integrate AI-driven personalized learning with human interaction. Research into the optimal balance between AI and human instruction will be vital in creating educational environments that leverage the strengths of both [21]. The use of AI in gamification has shown promise in enhancing student engagement and motivation [18]. However, much remains to be explored regarding how AI-driven gamification can be further optimized to cater to individual learning preferences and improve academic outcomes. Future studies should investigate the impact of personalized gamified experiences on long-term student engagement and the development of specific competencies, such as problem-solving and collaboration [2]. Moreover, there is a need for research into how AI can dynamically adjust game mechanics in real-time to maintain an optimal challenge level, ensuring sustained motivation and learning [20].

As AI becomes more integrated into educational systems, both educators and students must develop a foundational understanding of AI technologies and their ethical implications [9]. Research into the development of AI literacy programs that equip educators with the skills to effectively use AI tools in the classroom is essential. Additionally, AI literacy initiatives aimed at students can help them engage responsibly with these technologies, ensuring that they are equipped to navigate the AI-driven educational landscape [14].

8 Conclusion

This paper has explored the transformative role of AI in personalized learning, highlighting its potential to significantly enhance student engagement and learning outcomes through adaptive strategies. AI-powered systems, such as machine learning-based recommendation engines, intelligent tutoring, and gamification, provide dynamic, student-centered learning experiences that cater to individual needs, preferences, and learning paces. Despite these advancements, challenges related to data privacy, algorithmic biases, and ethical considerations must be addressed to ensure equitable and effective integration of AI in education.

Moreover, while AI's impact on personalization is evident, more empirical research is required to evaluate its long-term effectiveness across diverse educational contexts. There is also a critical need for frameworks that promote ethical AI usage, safeguard data privacy, and ensure inclusivity. Future research should explore hybrid models that combine AI technologies with traditional teaching methods to optimize student outcomes. Ultimately, AI's potential in education lies in its ability to provide tailored, adaptive learning experiences, but its implementation must be done responsibly to support all learners equitably.

References

1. W. K. Monib, A. Qazi, L. De Silva and M. M. Mahmud, "Exploring Learners' Experiences with ChatGPT in Personalized Learning," 2024 6th International Workshop on Artificial Intelligence and Education (WAIE), Tokyo, Japan, 2024, pp. 66–70, doi: 10.1109/WAIE63876.2024.00019.
2. Armstrong, M. B. and Landers, R. N., "An evaluation of gamified training: Using narrative to improve reactions and learning," *Simulation & Gaming*, vol. 48, no. 1, p. 104687811770374, 2017, DOI: 10.1177/1046878117703749.
3. Amin, S., Alarood, A. A., Mashwani, W. K., Alzahrani, A., and Alzahrani, A. O., "Smart E-Learning Framework for Personalized Adaptive Learning and Sequential Path Recommendations Using Reinforcement Learning," *IEEE Access*, vol. 11, pp. 89769–89790, 15 Aug. 2023, DOI: 10.1109/ACCESS.2023.3305584.

4. C. A. Bonfield, M. Salter, A. Longmuir, M. Benson and C. Adachi, "Transformation or evolution?: Education 4.0, teaching and learning in the digital age," *Education 4.0*, vol. 23752696, 2020, DOI: 10.1080/23752696.2020.1816847.
5. Darejeh, A., Moghadam, T. S., Delaramifar, M., and Mashayekh, S., "A Framework for AI-Powered Decision Making in Developing Adaptive e-Learning Systems to Impact Learners' Emotional Responses," in *Proc. 2024 11th International and 17th National Conference on E-Learning and E-Teaching (ICeLeT)*, 27–29 Feb. 2024, DOI: [Add DOI if available].
6. M. Castelli and L. Manzoni, "Special issue: Generative models in artificial intelligence and their applications," *Applied Sciences*, vol. 12, no. 9, p. 4127, 2022, DOI: 10.3390/app12094127.
7. R. Costa-Mendes, T. Oliveira, M. Castelli and F. Cruz-Jesus, "A machine learning approximation of the 2015 Portuguese high school student grades: A hybrid approach," *Education and Information Technologies*, vol. 26, pp. 1527–1547, 2021, DOI: 10.1007/s10639-020-10331-0.
8. G. Cooper, "Examining science education in ChatGPT: An exploratory study of generative artificial intelligence," *Journal of Science Education and Technology*, vol. 32, pp. 444–452, 2023, DOI: 10.1007/s10956-023-10052-2.
9. I. Khan, A. R. Ahmad, N. Jabeur and M. N. Mahdi, "An artificial intelligence approach to monitor student performance and devise preventive measures," *Smart Learning Environments*, vol. 8, article 17, 2021, DOI: 10.1186/s40561-021-00166-9.
10. K. Saddhono, R. Suhita, W. Istanti, A. Kusmiatun, D. Kusumaningsih and I. K. Sukmono, "AI-Powered Language Learning: Enhancing Literacy in the Digital Age," 2024 4th International Conference on Advancement in Electronics & Communication Engineering (AECE), Ghaziabad, India, 2024, pp. 856–861, doi: 10.1109/AECE62803.2024.10911149.
11. J. Borenstein and A. Howard, "Emerging challenges in AI and the need for AI ethics education," *AI and Ethics*, vol. 1, pp. 61–65, 2021, Available at: https://doi.org/10.1007/s43681-020-00010-x.
12. C. A. Eden, O. N. Chisom and I. S. Adeniyi, "Integrating AI in education: Opportunities, challenges, and ethical considerations," *Magna Scientia Advanced Research and Reviews*, vol. 10, no. 2, pp. 006–013, 2024, Available at: https://doi.org/10.30574/msarr.2024.10.2.0039.
13. D. Mehta, N. Chatterji, A. K. Gupta, P. D. Yadav, S. Dash and P. Verma, "The Application of Personalization, MachineVoice, Credibility, Assessement and Evaluation in Artificial Intelligence of Higher Education," 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM), Noida, India, 2024, pp. 1–5, doi: 10.1109/ICIPTM59628.2024.10563539.
14. Z. Ahmed et al., "The Generative AI Landscape in Education: Mapping the Terrain of Opportunities, Challenges, and Student Perception," *IEEE Access*, vol. 12, pp. 147023–147050, 2024, doi: 10.1109/ACCESS.2024.3461874.
15. K. Jafarzade, "The Role of GPT Models in Education: Challenges and Solutions," 2023 5th International Conference on Problems of Cybernetics and Informatics (PCI), Baku, Azerbaijan, 2023, pp. 1–3, doi: 10.1109/PCI60110.2023.10325940.
16. H. Shahri, M. Emad, N. Ibrahim, R. N. B. Rais and Y. Al-Fayoumi, "Elevating Education through AI Tutor: Utilizing GPT-4 for Personalized Learning," 2024 15th Annual Undergraduate Research Conference on Applied Computing (URC), Dubai, United Arab Emirates, 2024, pp. 1–5, doi: 10.1109/URC62276.2024.10604578.
17. R. Sandu, E. Gide, S. Karim and P. Singh, "A Framework for GenAI-Empowered Curriculum and Learning Resources: A Case Study from an Australian Higher Education,"

2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET), Paris, France, 2024, pp. 1–8, doi: 10.1109/ITHET61869.2024.10837623.

18. S. Vijayalakshmi, B. Madhavi, J. N., G. S. Bansode, N. Sharma and S. K. G., "Smart Education With IoT And AI: Revolutionizing Learning In The Digital Age," 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2024, pp. 1282–1286, doi: 10.1109/ICDT61202.2024.10489741.
19. M. Y. Al Balushi and A. S. Al Harthi, "Adaptive Learning Pathways: Advancing Personalization in Online Education Platforms," 2024 2nd International Conference on Computing and Data Analytics (ICCDA), Shinas, Oman, 2024, pp. 1–6, doi: 10.1109/ICCDA64887.2024.10867273.
20. Duolingo, "How Duolingo's AI learns what you need to learn," *IEEE Spectrum*, 2023, Available at: https://spectrum.ieee.org/duolingo.
21. K. Bicknell and C. Brust, "Learning how to help you learn: Introducing Birdbrain!" *Duolingo Blog*, 2020, Available at: https://blog.duolingo.com/learning-how-to-help-you-learn-introducing-birdbrain.
22. Claned, "The Role of AI in Personalized Learning," Claned, 2024, Available at: https://claned.com/the-role-of-ai-in-personalized-learning/.
23. C. Randieri, "Personalized Learning And AI: Revolutionizing Education," *Forbes Technology Council*, 2024, Available at: https://www.forbes.com/councils/forbestechcouncil/2024/07/22/personalized-learning-and-ai-revolutionizing-education/.
24. Meehirr, K., "How AI is personalizing education for every student," eLearning Industry, 2023, Available at: https://elearningindustry.com/how-ai-is-personalizing-education-for-every-student

4. Changes in Online Shopping Behavior During Pandemics: A Linear Regression Approach

Luan Sinanaj[1]

[1] "Aleksander Moisiu" University, Department of Information Technology, Durres, Albania
luansinanaj@uamd.edu.al

Abstract. Life is being day by day meaningless without the use of technology and internet. It has become an important part of daily life and we can say that the use of technology and the Internet during the pandemic has been a short solution to many problems, such as paying bills, learning, teaching, buying products online, etc.

In this study the purpose is to analyze the impact of pandemics on online purchases. The method used in this research paper is quantitative. Firstly, it was built a structured questionnaire to collect data and all this process was made possible via emails and social networks. From this step it was generated a sample of 164 consumers.

The results from the inferential analysis of this study showed that the: (1) People can protect their health from diseases by making online purchases more than physically during a pandemic; (2) Online shopping during the pandemic has affected to the health protection from diseases; (3) Buying more products online than physically during the pandemic is affected by the time of receiving the product, the difficulty of returning the product and the risk of not getting what I paid for.

Keywords: Online Purchase, Online Shopping, Pandemic, Linear Regression.

1 Introduction

The use of technology and the internet is increased in every aspect of life. It is clear that in recent decades the development of technology and the use of the Internet has had a significant and evident increase, an almost exponential increase. Moreover, we can say that the use of technology and the Internet proved to be particularly important in the pandemic time. The impact of the pandemic on the use of technology and the Internet would be in many different sectors such as remote work, teaching, learning but also in online shopping [1].

The purpose of this study is to analyze online shopping and its significance in the pandemic conditions. It is important to protect our health and the health of others during a pandemic. One way to protect our health in times of pandemic is to avoid physical contact with people as much as possible, and online shopping has been a way to avoid as much physical contact between people as possible. This study will investigate consumer behavior in online shopping, specifically, whether health protection has affected the purchase of more products online than physically, or whether online shopping has affected the protection of health during pandemic time. Another element that will be investigated is whether the time of receiving the product, the difficulty of returning the product and the risk of not receiving the product which you paid for has affected the purchase of more products online than physically, and another more element that will be a result from this study is which products people prefer to buy more online.

The research questions and hypothesis raised of this study are:

- RQ1: Has health protection affected the purchase of more products online than physically during a pandemic?
- $H1_0$: The health protection **has not affected** the purchase of more products online than physically during a pandemic.
- RQ2: Has online shopping during the pandemic affected to the protection of health?
- $H2_0$: Online shopping during the pandemic **has not affected** to the health protection.
- RQ3: Did the factors (time, return and risk) affect the purchase of more products online than physically during the pandemic time?
- $H3_0$: Buying more products online than physically during pandemic **has not been affected** by time of receiving the product, the difficulty of returning the product and the risk of not getting what I paid for.

The structure of this research study is organized into 5 sections. In section 2 is explained the research methodology. Section 3 includes a review of the literature on online shopping. In section 4, we have the results of the study as well as the inferential analysis. In section 5 we have the conclusions which are followed by references.

2 Literature Review

The literature review is based on the research questions and hypotheses raised for this study. This section reflects the definitions of online shopping and the studies that have been done for online shopping especially during the pandemic to see the impact that the pandemic time has had on online shopping.

Last advancements of the technology acceptance theory deepen our understanding of consumers engaging in online shopping. Technology Acceptance Model or TAM (Davis, 1989) assumed that the primary motivators to technology use would be its usefulness and ease of access. In the scope of this study, these dimensions relate to the convenience as well as the risks involved in shopping online, particularly during crisis moments like the pandemic situation [2].

Moreover, Ajzen (1991) introduced Theory of Planned Behavior (TPB), which incorporates attitudes, subjective norms, perceived behavioral control alongside the mixed intentions to predict preceded behaviors. TPB is quite useful in explaining the impact social influence and personal control over different purchasing options have on one's online shopping activities during a health crisis. These theories together assist in answering the research questions and forming coherent hypotheses with existing frameworks [3].

Assigning the models referenced above has aided in analyzing the shifts in consumer behavior with respect to technology or social constraints for studies Theodorou et al. (2023) [4], Leong & Chaichi (2021) [5], and Zhang & Chen (2023) [6]. Nevertheless, little use of TAM and TPB can be found in the context of Eastern Europe or the Balkans and, specifically, Albania case. This study addresses the gap by implementing a designed inferential approach to these overlooked regions.

Constructed from these theoretical assumptions, the proposed hypotheses for this research follow the behavioral patterns but apply them to the consideration of the pandemic. An example would be perceived risks and delivery-related issues as barriers to behavioral control, while health protection can be reframed as a form of usefulness from TAM. The integration of these hypotheses with TAM and TPB enhances the validity of the model and bridges the gap between consumer behavior and theoretical frameworks.

By definition, Online Shopping, according to author Myriam Ertz, is any kind of purchase activity made via the Internet [7]. However, according to authors Mahmoud Amer and Jorge Marx Gómez, online shopping is the process in which consumers purchase services or products over the Internet [8]. From a slightly earlier study by authors Jackson, Rowlands & Miller, Online Shopping was seen as a method of purchasing through devices like smartphones or computers using the internet [9].

According to another study, it is explained that the use of online shopping has increased with the use of the internet, although, a good part of consumers uses the information collected online to make physical purchases in the store [10]. Overall, the trend of online shopping has grown rapidly also due to the development and easier access to the internet, and in fact according to research by the University of California, online shopping is the most popular activity along with the use of email and Internet browsing [11].

According to the research conducted in May 2020, the pandemic time has had an impact on online sales. This research shows that consumers bought more online because of pandemic time and some of them confirmed that they started shopping online for the first time during the pandemic. The research concludes that during the pandemic consumers turned to online shopping for food products, daily necessities or other products. Also, it is worth noting that at the same time, in some sectors such as travel and airlines, declined especially during April 2020 due to pandemic time [12].

The research findings of the authors Hoang Viet Nguyen, Hiep Xuan Tran, Le Van Huy, Xuan Nhi Nguyen, Minh Thanh Do & Ninh Nguyen show that the intention of buying at consumers is less influenced by positive emotions and pleasant feelings related to such a behavior because most consumers are concerned about the pandemic situation [13].

From another research of Bayad Jamal Ali for online shopping it results that there has been an increase in the last years due to the pandemic [14].

3 Methodology

This study was carried out in Albania, focusing on high-frequency internet users, which include students, university personnel, and academic experts. Albania is particularly useful for providing information concerning consumer behavior due to its status as an economy in transition with developing digital systems.

The study adopted a quantitative research approach. Data were obtained through a pre-established survey, which is a fundamental tool in quantitative research. Quantitative research focuses on the collection of data through various means, including surveys, as well as from existing datasets, and further analyzes the data using statistical, mathematical or numerical techniques. This approach is also characterized by the possibility to carry out descriptive statistical analysis, tests of significance, predictive modeling for populations, and wide generalization of findings to larger populations as stated in [15], [16], and [17].

The questionnaire was designed based on several primary studies with validated tools by using their findings to develop customized models. Specifically, we drew on the standard models provided by Patrick Hille, Gianfranco Walsh and Mark Cleveland [18] as well as Ismail Erkan and Chris Evans [19]. Some of the study's contextual variables were altered in order to adjust for the scope of the research. The documents that serve as sources for building the questionnaire and defining the variables are [20], [21], and [22].

Participants were recruited through email corresponding to two universities which included professors, students, and administrative personnel, alongside social media and e-learning platforms. Over 800 invitations were distributed, ultimately 164 participants agreed to complete the survey with 53.7% identifying as female and 46.3% as male. No financial or material incentives were provided.

The approach taken was based upon models from literature. For example, some concerns were raised with regards to not getting services they had paid for which was framed as perceived risk and delivery time difficulty and return ease were measured with items from previous studies using a Likert scale.

Anonymity was ensured which reduced response bias and enhanced overall data reliability. The data were cleaned and analyzed with SPSS. Reliability was assessed by internal consistency and resulted in a strong value of 0.892 using Cronbach's alpha. Regression testing of the proposed assumptions was conducted through bivariate and multivariate linear regression analysis.

This approach in combination produces counterbalancing contextual relevance while rigorously answering the research questions, improving our understanding on the impacts of behavioral and logistical challenges on online shopping in times of crisis in emerging economies.

Limitations

This study had its limitations. The first is digital coverage. Second, the disadvantage of this study was that in different areas people are not able to access the internet or order products because delivery service is not possible. Therefore, we cannot generalize and implement our study for these areas.

4 Results and Discussions

The outcomes derived from the linear regression models confirm the presence of distinct behavioral trends in online shopping during the pandemic, which correspond to the expectations set out in the Technology Acceptance Model (TAM) and the Theory of Planned Behavior (TPB). For instance, the significant positive relationship noted between health protection and the frequency of online purchases (Hypothesis 1) suggests that individuals regard online shopping, accurately, as a convenient measure to mitigate physical contact and thus safeguard their health. This reflects the usefulness aspect of TAM.

Moreover, the validation of Hypothesis 2 concerning the influence of purchase intention variables (PDP2) on health protection perception demonstrates how behavioral intent, shaped by the efficacy and convenience offered by the online domain, directly impacts perceived outcomes, supporting TPB's claim that attitudes and control beliefs serve as foundational factors driving behavior.

How logistical issues like the time needed to receive the products as well as the potential for non-delivery risk greatly influences online buying decisions is presented in the third hypothesis. These considerations can be viewed as constraining factors in TPB's perceived behavioral control, in which case, external factors constrain the likelihood of engaging in a certain activity. Here, lower delivery time and greater trust of fulfillment garner more control and encourage frequent shopping, acting as positive influences.

Moreover, the fact that the challenge related to the return of products was not statistically significant may suggest that, during times of crisis, consumers are likely to accept small inconveniences if essential needs such as health and promptness are met. This seems to bring about a change in spending habits due to some external factors, such as the restrictions forced during the pandemic.

In combination, these results not only confirm the proposed hypotheses, but also offer an explanation on the behavioral models which reinforces the theoretical contribution of the study. The practical outcomes address e-commerce services by clearly improving delivery dependability as well as managing consumer confidence, which will increase online participation especially during times of crisis.

Study limitations include possible sampling bias since most participants were associated with a university and had stable internet connection. Also, these findings may not be generalized to rural populations with limited technology. Future work should focus on cross demographic or geographic sample comparisons, including the examination of behavioral endurance post-pandemic.

Descriptive Analysis

From a descriptive analysis of the questionnaire completed by 164 individuals, it resulted that:

1) their age was 58.5% between 18-24 years old; 25% between 25-34 years old; 12.2% between 35-44; and 4.3% over 45 years old.
2) Educational level degree was: 12.2% High school; 50% Bachelor; 31.1% Master and 6.7% PhD.

3) Regarding the question of the questionnaire: "How often do you use the Internet every day?", The respondents answered: 2.4% less than 1 hour; 9.7% from 1 to 2 hours; 37% from 3 to 4 hours; 28.5% from 5 to 6 hours; and 22.4% more than 6 hours per day. From a simple analysis of the data shows that about 51% of respondents use the Internet for more than 5 hours a day or 87% of them use it more than 3 hours a day.

4) Regarding the question of what products they usually buy more on the Internet, the respondents answered in the order: 64.2% Clothing; 43.6% Electronics; 40.6% Personal Care Products; 27.3% Books; 23.6% Home appliances; and 14.5% Food products.

5) The following table presents the variables with abbreviations that have been used in inferential analysis to test hypotheses:

Table 1. All variables used in questionnaire for this study.

	Abbreviation	Variables
Online Purchase Intention (PDP, Purchase During Pandemic)	PDP1	I would use the Internet for purchasing a product
	PDP2	Using the Internet for purchasing a product is something I would do
	PDP3	I could see myself using the Internet to buy a product
	OSHPHC19	Online Shopping protects our health during pandemic.
	BMOTPDC19	Buying more products online than physically during pandemic
	HPFC19	Health protection during pandemic
	TRP	Time to receive the product
	DRP	Difficulty in returning products/items

	RNGWP	Risk of not getting what I paid for

As mentioned above in the methodology, the three variables PDP1, PDP2 and PDP3 are for measuring online purchase intention during a pandemic. Table 2 below, shows the frequencies in percent retrieved from SPSS for online purchase intention (PDP1, PDP2 and PDP3) variables:

Table 2. Frequencies of variables PDP1, PDP2 and PDP3.

Variables		PDP1	PDP2	PDP3	Total
Valid	Never	9.1	6.7	11	26.8
	Rarely	14.0	17.1	15.2	46.3
	Sometimes	31.1	32.3	27.4	90.8
	Often	23.2	24.4	22.6	70.2
	Very Often	22.6	19.5	23.8	65.9
	Total	100.0	100.0	100.0	

As shown in Table 2 for the variables PDP1, PDP2 and PDP3, the "Never" answer of the respondents is respectively 9.1%, 6.7% and 11%; the "Rarely" answer of the respondents is respectively 14%, 17.1% and 15.2%; the "Sometimes" answer of the respondents is respectively 31.1%, 32.3% and 27.4%; the "Often" answer of the respondents is respectively 23.2%, 24.4% and 22.6%; and, the “Very Often” response of the respondents is respectively 22.6%, 19.5% and 23.8%. These results show that a high percentage of respondents used the internet to make online purchases during the pandemic period.

Table 3, shows the frequencies in percent retrieved from SPSS for “*Online Shopping protects our health during pandemic”* variable:

Table 3. Frequencies of variable OSHPHC19.

OSHPHC19		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	17	10.4	10.4	10.4
	Disagree	17	10.4	10.4	20.7
	Neutral	43	26.2	26.2	47.0
	Agree	48	29.3	29.3	76.2
	Strongly Agree	39	23.8	23.8	100.0
	Total	164	100.0	100.0	

As shown in the table above for the variable OSHPHC19 (*Online Shopping protects our health during pandemic*), 10.4% of respondents answered "Strongly Disagree", 10.4% answered "Disagree", 26.2% answered "Neutral", 29.3 % answered "Agree", and 23.8% answered "Strongly Agree". Interesting from the results is that about 53.1%

of respondents agree with the fact that online shopping protects our health during pandemic, while about 20.8% of respondents disagree with this fact. The difference between those who agree and those who disagree is 32.3%, the rest of the respondents are neutral.

Table 4, shows the frequencies in percent retrieved from SPSS for "*Buying more products online than physically during pandemic*" variable:

Table 4. Frequencies of variable BMOTPDC19.

BMOTPDC19		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly Disagree	23	14.0	14.0	14.0
	Disagree	27	16.5	16.5	30.5
	Neutral	37	22.6	22.6	53.0
	Agree	52	31.7	31.7	84.8
	Strongly Agree	25	15.2	15.2	100.0
	Total	164	100.0	100.0	

As shown in the table above for the variable BMOTPDC19 (*Buying more products online than physically during pandemic*), 14% of respondents answered "Strongly Disagree", 16.5% answered "Disagree", 22.6% answered "Neutral", 31.7 % answered "Agree", and 15.2% answered "Strongly Agree". Interesting from the results is that about 49% of respondents agree with the fact that buying more products online than physically during pandemic time, while about 31% of respondents disagree with this fact. The difference between those who agree and those who disagree is 18%, the rest of the respondents are neutral.

In reviewing the frequencies of the variables in the following tables, it is measured how important they are with a Likert scale from 1 to 5.

Table 5, shows the frequencies in percent retrieved from SPSS for "*Health protection during pandemic*" variable:

Table 5. Frequencies of variable HPFC19.

HPFC19		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Very unimportant	18	11.0	11.0	11.0
	Unimportant	16	9.8	9.8	20.7
	Neutral	27	16.5	16.5	37.2
	Important	38	23.2	23.2	60.4
	Very important	65	39.6	39.6	100.0
	Total	164	100.0	100.0	

As shown in the table above for the variable HPFC19, for the question of how important is *Health protection during pandemic*, 11% of respondents answered "Very unimportant", 9.8% answered "Unimportant", 16.5% answered "Neutral", 23.2% answered "Important", and 39.6% answered "Very important". The results show that about 62.8% of respondents think that the protection of health during pandemic is important, while about 20.8% think that it is not important. The difference between those who think it is important and those who think it is not important is 42%, this difference is relatively high, the rest of the respondents are neutral.

The following table gives the percentage frequencies for the three independent variables TRP, DRP and RNGWP, for which the respondents are asked respectively how important it is for them "*Time to receive the product*", "*Difficulty in returning products*" and "*Risk of not getting what I paid for*".

Table 6. Frequencies of variables TRP, DRP and RNGWP.

	Variables	TRP	DRP	RNGWP	Total
Valid	Very unimportant	11.0	7.9	5.5	24.4
	Unimportant	6.7	12.2	7.3	26.2
	Neutral	15.9	12.8	18.9	47.6
	Important	18.9	32.3	14.6	65.8
	Very important	47.6	34.8	53.7	136.1
	Total	100.0	100.0	100.0	

As shown in Table 6 for the variables TRP, DRP and RNGWP, the “Very Unimportant” response of the respondents is respectively 11%, 7.9% and 5.5%; the "Unimportant" response of the respondents is respectively 6.7%, 12.2% and 7.3%; the "Neutral" response of the respondents is respectively 15.9%, 12.8% and 18.9%; the "Important" response of the respondents is respectively 18.9%, 32.3% and 14.6%; and the “Very important” response of the respondents is respectively 47.6%, 34.8% and 53.7%.

The results show that about 66.5% of respondents think that TRP is important, while about 17.7% think it is not important. The difference between them is significant 48.8%, the rest of the respondents are neutral.

Regarding the DRP variable, the results show that about 67.1% of respondents think that DRP is important, while about 20.1% think that it is not important. The difference between them is 47% and the rest of the respondents are neutral.

And the results of the last variable RNGWP show that about 68.3% of respondents think that RNGWP is important, while about 12.8% think it is not important. The difference between them is very high 55.5%, the rest of the respondents are neutral.

From a first analysis of the frequencies of the variables in Table 6, it is noticed that with a high percentage "Time to receive the product", "Difficulty in returning products" and "Risk of not getting what I paid for" are important for respondents.
The following table presents the Correlation Matrix retrieved from SPSS program of all variables in this research.

Table 7. Matrix of Correlations of all variables.

	PDP1	PDP2	PDP3	OSHPHC19	BMOTPDC19	HPFC19	TRP	DRP	RNGWP
PDP1	1	.876	.772	.490	.505	.285	.365	.295	.346
PDP2	.876	1	.777	.514	.512	.288	.329	.298	.294
PDP3	.772	.777	1	.396	.567	.208	.315	.247	.254
OSHPHC19	.490	.514	.396	1	.499	.589	.465	.450	.467
BMOTPDC19	.505	.512	.567	.499	1	.311	.397	.316	.413
HPFC19	.285	.288	.208	.589	.311	1	.691	.655	.666
TRP	.365	.329	.315	.465	.397	.691	1	.783	.772
DRP	.295	.298	.247	.450	.316	.655	.783	1	.816
RNGWP	.346	.294	.254	.467	.413	.666	.772	.816	1

**. Correlation is significant at the 0.01 level (2-tailed).

The following graph presents the Heatmaps of the Correlation Matrix which was realized with *DisplayR* program [23].

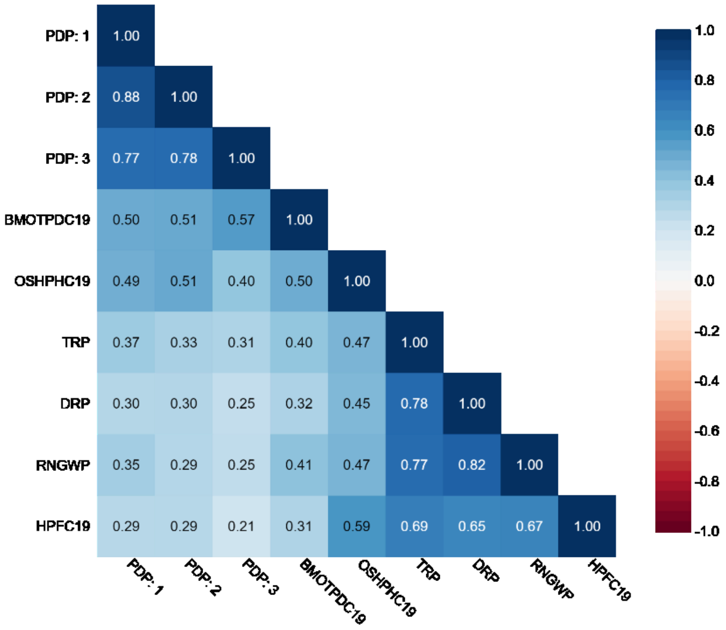


Fig. 1. Heatmaps of a Correlation Matrix elaborated with DispalyR.

Inferential Analysis: Test of Hypothesis

Before performing the inferential analysis, the reliability of the questionnaire was checked through the SPSS program, where it turned out that the Cronbach Alpha coefficient had a value of 0.892, an almost excellent value. The following table shows the result:

Table 8: Reliability Statistics.

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.891	.892	9

Bivariate and multivariate linear regression was used to test the hypotheses.

Regarding the first hypothesis: "*$H1_0$: The health protection* ***has not affected*** *the purchase of more products online than physically during a pandemic*" is used bivariate linear regression. Independent variable is HPFC19 and dependent variable is BMOTPDC19. The results are presented in Table 9 and Figure 2 below:

Table 9. Regression outputs between HPFC19 and BMOTPDC19.

Coefficients ^{a}

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations
	B	Std. Error	Beta			
1 (Constant)	2.098	.276		7.606	.000	
HPFC19	.291	.070	.311	4.169	.000	.311

a. Dependent Variable: BMOTPDC19

The significant value in the second row is 0.00, this number is less than 0.05 therefore $H1_0$ of the non-existing the relation between the variables is rejected (falls down).

Based on the linear regression equation: $Y = a + bX + e$, where Y is for the dependent variable and X is for the independent variable, the model of the linear regression equation for our case is:

$BMOTPDC19 = 2.098 + 0.291 * HPFC19 + e$,

where, a = 2.098 is the regression constant and b = 0.291 is the regression coefficient. From the equation model, we can say that there is a positive correlation between the independent variable HPFC19 and the dependent variable BMOTPDC19, where the increase by one unit of the independent variable affects the increase of 0.291 times (or 29.1%) in the dependent variable. From the analysis we come to the conclusion that the raised H1 hypothesis is proved. This conclusion is also reinforced by the following linear regression graph of our model:

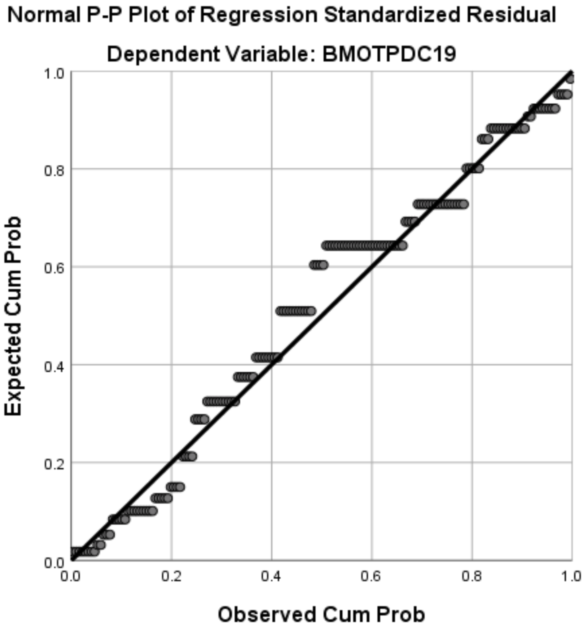


Fig. 2. Linear regression graph for the equation of our model (H1).

Regarding the second hypothesis: "*$H2_0$: Online shopping during the pandemic* ***has not affected*** *to the health protection.*" has been used the multiple linear regression. Independent variables are PDP1, PDP2, PDP3 and dependent variable is OSHPHC19. The results are presented in Table 10 and Figure 3 below:

Table 10. Regression outputs between PDP1, PDP2, PDP3 and OSHPHC19.

Coefficients ^{a}

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations
	B	Std. Error	Beta			
1 (Constant)	1.595	.261		6.120	.000	
PDP1	.192	.149	.190	1.294	.198	.490
PDP2	.416	.159	.389	2.624	.010	.514
PDP3	-.051	.109	-.053	-.469	.640	.396

a. Dependent Variable: OSHPHC19

The significant values in the *Sig.* column for the variables PDP1, PDP2, PDP3 are respectively .198, .010 and .640. The significant value of PDP2 variable is less than 0.05 therefore $H2_0$ of the non-existing the relation between the variables is rejected (falls down).

Based on the multiple linear regression equation:

$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + e,$$

where Y is for the dependent variable and X_1, X_2, X_3 are for the independent variables, the model of the linear multiple regression equation for our case is:

$$OSHPHC19 = 1.595 + 0.192 * PDP1 + 0.416 * PDP2 - 0.051 * PDP3 + e,$$

where, a=1.595 is the regression constant and b_1=0.192, b_2=0.416 and b_3=-0.051 are the regression coefficients.

In our case, in the linear multiple regression model, there is a positive correlation between the independent variable PDP2 and the dependent variable OSHPHC19, where the increase by one unit of the independent variable affects the increase of 0.416 times (or 41.6%) in the dependent variable. In this case it turned out that the independent variable PDP2 is statistically very important for this model because the increase by one of its units, increases by 41.6% the dependent variable OSHPHC19. From inferential analysis we conclude that the hypothesis raised H2 is proved. This conclusion is reinforced by the following graph of multiple linear regression of our model:

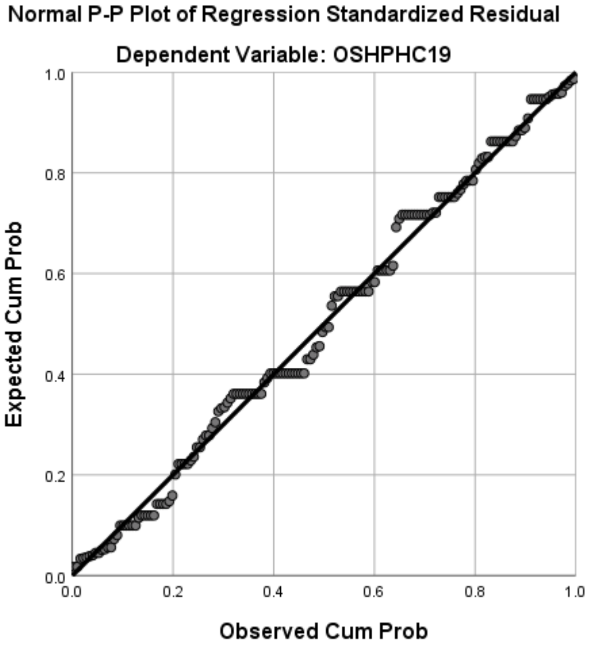


Fig. 3. Multiple linear regression graph of our model (H2).

Regarding the third hypothesis: "*$H3_0$: Buying more products online than physically during pandemic **has not been affected** by time of receiving the product, the difficulty of returning the product and the risk of not getting what I paid for.*", in this case same as the previous case, has been used the multiple linear regression. Independent variables are TRP, DRP, RNGWP and dependent variable is BMOTPDC19. The results are presented in Table 11 and Figure 4 below:

Table 11. Regression outputs between TRP, DRP, RNGWP and BMOTPDC19.

		Coefficients [a]					
		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations
Model		B	Std. Error	Beta			
1	(Constant)	1.419	.314		4.521	.000	
	TRP	.245	.114	.263	2.137	.034	.397
	DRP	-.184	.136	-.183	-1.357	.177	.316
	RNGWP	.372	.137	.359	2.719	.007	.413

a. Dependent Variable: BMOTPDC19

The significant values in the *Sig.* column for the variables TRP, DRP, RNGWP are respectively .034, .177, and .007. The significant value for the TRP variable is 0.034 and for the RNGWP variable is 0.007, these numbers are less than 0.05 therefore the $H3_0$ of the non-existing the relation between the variables is rejected (falls down). Based on the multiple linear regression equation (same at the previous case in testing second hypothesis), in our case the model of the linear multiple regression equation is:

$$BMOTPDC19 = 1.419 + 0.245 * TRP - 0.184 * DRP + 0.372 * RNGWP,$$

where, a=1.495 is the regression constant and b_1=0.245, b_2=-0.184 and b_3=0.372 are the regression coefficients.

In this case there are two positive correlations between the independent variables TRP, RNGWP and the dependent variable BMOTPDC19, where the increase by one unit of the independent variable TRP affects the increase of 0.245 times (or 24.5%) in the dependent variable BMOTPDC19, and the increase with a unit of the independent variable RNGWP affects by increasing 0.372 times (or 37.2%) the dependent variable BMOTPDC19. Thus, in relation to the linear multiple regression model, it turned out that the independent variables TRP and RNGWP are statistically significant for this model where the unit increase of these variables increases by 24.5% and by 37.2% the dependent variable BMOTPDC19. This means that from the inferential analysis the raised hypothesis H3 is proved. This conclusion is reinforced by the following graph of the multiple linear regression of our model:

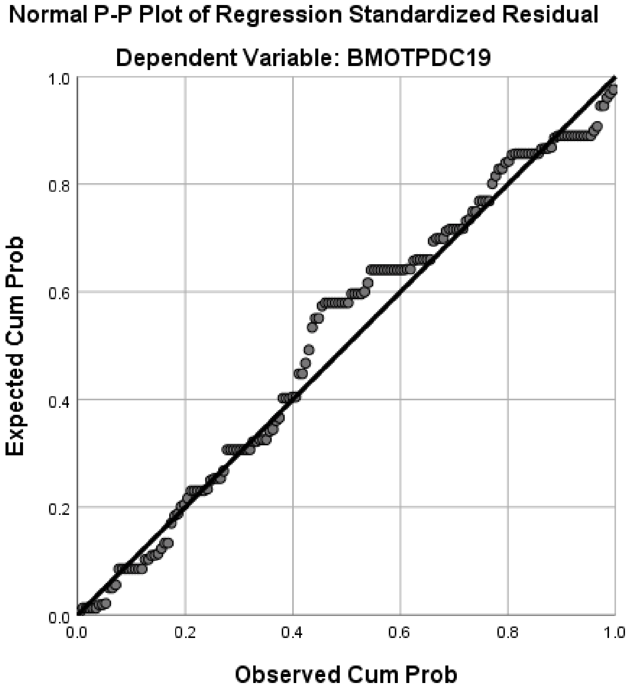


Fig. 4. Multiple linear regression graph of our model (H3).

5 Conclusions

The main objective of this paper was to analyze the online shopping behavior of individuals during the pandemic, with a particular focus on a transitional economy like Albania. The findings emphasize a significant increase in online purchasing, primarily driven by health restrictions and the necessity to avoid physical shopping environments.

The empirical data gathered through a structured questionnaire enabled the application of inferential analysis to test three hypotheses related to consumer perceptions. The analysis confirmed that these hypotheses are supported and that the corresponding research questions were successfully answered, in specifically:

1. People can protect their health by making online purchases more than physically during a pandemic period.
2. Online shopping during the pandemic has contributed to health protection.

3. Buying more products online than physically during the pandemic is affected by delivery time, return difficulty, and the risk of not receiving what was paid for.

All three of the above points support the confirmation of the hypotheses raised in this study. Furthermore, the data show that most respondents consider online shopping more comfortable and safer, and they recommend it during health crisis conditions such as a pandemic.

Regression analysis supported the impacts of critical factors such as perceived risk, website dependability, return policies, and delivery lags into overall evaluation. These findings further add to available literature while presenting more evidence in the context of an emerging economy.

In terms of gender, women showed a greater level of concern regarding delivery lags and the safety of online transactions, while younger respondents showed more trust towards online systems. These demographic differences provide actionable recommendations for marketers.

Trust, together with the use of technologies, maintains the possibility for shaping consumer behavior and is critical even after the crisis. As startling as it may seem, these factors, which surfaced as necessary during the crisis, will continue to influence consumer behavior even after the pandemic.

There is no denying of the fact that e-commerce has already reached new heights in the likes of Albania, but there is a gap which is closing with other developed economies. With that said, further growth requires funding when it comes to infrastructure, logistics, internet, and further education on digital tools.

This research focuses on assessing the drastic impacts which the pandemic has brought upon consumers regarding shopping digitally. To keep up with evolving consumer demands, continuous investment in digital transformation will be critical.

References

1. R. De, N. Pandey and A. Pal, "Impact of digital surge during Covid-19 pandemic: A viewpoint on research and practice," International Journal of Information Management, vol. 55, no. 102171, 2020. https://doi.org/10.1016/j.ijinfomgt.2020.102171
2. F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, University of Minnesota, 1989. DOI: 10.2307/249008, pp. 319–340.
3. I. Ajzen, "The theory of planned behavior," *Organizational Behavior and Human Decision Processes*, vol. 50, no. 2, Elsevier, 1991. DOI: 10.1016/0749-5978(91)90020-T, pp. 179–211.
4. A. Theodorou, L. Hatzithomas, T. Fotiadis, A. Diamantidis, and A. Gasteratos, "The Impact of the COVID-19 Pandemic on Online Consumer Behavior: Applying the

Theory of Planned Behavior," *Sustainability*, vol. 15, no. 3, MDPI, 2023. DOI: 10.3390/su15032545.

5. M. K. Leong and K. Chaichi, "The Adoption of Technology Acceptance Model (TAM) and Trust in Influencing Online Purchase Intention During the COVID-19 Pandemic: Empirical Evidence from Malaysia," *International Journal of Academic Research in Business and Social Sciences*, vol. 11, no. 8, Human Resource Management Academic Research Society (HRMARS), 2021. DOI: 10.6007/IJARBSS/v11-i8/10422, pp. 468–478.
6. R. Zhang and M. Chen, "Predicting Online Shopping Intention: The Theory of Planned Behavior and Live E-Commerce," in *SHS Web of Conferences*, vol. 155, EDP Sciences, 2023. DOI: 10.1051/shsconf/202315502008.
7. M. Ertz, "Shipment Tracking, Delivery Speed, and Product Presentation as Antecedents of Repurchase Intention: Predictors of Online Shopping Repurchase Intention.," in *Handbook of Research on the Platform Economy and the Evolution of E-Commerce*, LaboNFC, University of Quebec at Chicoutimi, Canada., IGI Global, 2021. DOI: 10.4018/978-1-7998-7545-1, pp. 231-250.
8. M. Amer and J. M. Gómez, "Measuring B2C Quality of Electronic Service: Towards A Common Consensus.," in *E-Business Development and Management in the Global Economy.*, Western Illinois University, USA, IGI Global, 2010. DOI: 10.4018/978-1-61520-611-7.ch014, pp. 135-143.
9. D. Miller, P. Jackson and N. Thrift, Shopping, Place and Identity, London, New York: Routledge. *Taylor & Francis Group*., 1998.
10. S. M. Forsythe and B. Shi, "Consumer patronage and risk perceptions in Internet shopping," Journal of Business Researc, vol. 56, no. 11, pp. 867-875, 2003. https://doi.org/10.1016/S0148-2963(01)00273-9.
11. UCLA Center for Communication Policy, "Surveying the Digital Future," Center for the Digital Future at USC Annenberg, Los Angeles, CA 90064, 2003.
12. yStats GmbH & Co. KG, "COVID-19 Impact on Global E-Commerce & Online Payments - 2020," Research and Markets, Dublin, 2020.
13. H. X. T. L. V. H. X. N. N. M. T. D. &. N. N. Hoang Viet Nguyen, "Online Book Shopping in Vietnam: The Impact of the COVID-19 Pandemic Situation," Publishing Research Quarterly. Springer., vol. 36, no. 437–445, 2020. https://doi.org/10.1007/s12109-020-09732-2.
14. B. J. Ali, "Impact of COVID-19 on Consumer Buying Behavior Toward Online Shopping in Iraq," Economic Studies Journal. SSRN. ELSEIVER, vol. 18, no. 42, pp. 267-280, 2020.
15. D. Stockemer, in Quantitative Methods for the Social Sciences. A Practical Introduction with Examples in SPSS and Stata., Gewerbestrasse 11, 6330 Cham, Switzerland, Springer, 2019, pp. 8-10; 18-20. https://doi.org/10.1007/978-3-319-99118-4.
16. D. Mujis, "Doing Quantitative Research in Education with SPSS.," London, Thousand Oaks, New Delhi, SAGE Publications, 2012, pp. 1-26.
17. G. Marczyk, D. DeMatteo and D. Festinger, "Essentials of Research Design and Methodology," New Jersey, John Wiley & Sons, Inc., 2005, pp. 49-50.

18. H. Patrick, W. Gianfranco and C. Mark, "Consumer Fear of Online Identity Theft: Scale Development and Validation," Journal of Interactive Marketing. ScienceDirect., vol. 30, pp. 1-19, 2015. https://doi.org/10.1016/j.intmar.2014.10.001.
19. I. E. &. C. Evans, "Social media or shopping websites? The influence of eWOM on consumers' online purchase intentions.," Journal of Marketing Communications. Taylor & Francis Group., 2016. http://dx.doi.org/10.1080/13527266.2016.1184706.
20. SurveyMonkey, "Online shopping attitudes survey template," Momentive, [Online]. Available:https://www.surveymonkey.com/mp/online-shopping-attitudes-survey-template/. [Accessed December 2024].
21. FreeOnlineSurveys, "Online Shopping Experience Survey," Problem Free Ltd. 47 Newfoundland Way, Portishead, Bristol. BS20 7FP, United Kingdom., [Online]. Available: https://freeonlinesurveys.com/. [Accessed December 2024].
22. M. Kashif, Aziz-Ur-Rehman and M. K. Javed, "COVID-19 IMPACT ON ONLINE SHOPPING," International Journal of Medical Science in Clinical Research andReview, vol. 3, no. 4, pp. 325-330, 2020. https://ijmscrr.in/index.php/ijmscrr/article/view/92/69.
23. DisplayR, "Displayr | Analysis and Reporting Software for Survey Data," DisplayR, [Online]. Available: https://www.displayr.com/. [Accessed December 2024].

5. Fake News Detection Using Machine Learning and the Training of Models in Python

Blerina Çeliku[1] and Rafail Prodani[2] and Vasil Bardhi[3]

[1,2] Fan S. Noli University, Korçë, Albania
[3] Tirana University, Tiranë, Albania

bceliku@unkorce.edu.al

Abstract. The detection of fake news is an emerging field of research that requires a good knowledge of approaches to identify the meaning of concepts like false information and using various digital tools, platforms and training models to detect that kind of information. There is a literature review about fake news and their detection, machine learning algorithms and classifiers used to identify the fake news, exploring various aspects of machine learning focusing on creating and using training models, computer vision and natural language processing. In this study we used five publicly available datasets — LIAR, ISOT, FakeNewsNet, COVID-19 Fake News Dataset, and the Kaggle Fake and Real News Dataset — which were preprocessed in Python using standard NLP techniques, developed and trained models with TensorFlow as the main tool integrated with Python libraries, Google Colab and Flask. This paper is focused on the training of Recurrent Neural Network and Convolutional Neural Network in Tensorflow that give some specific testing performance metrics. According to these metrics, the Bidirectional Long Short-Term Memory is chosen to be used for implementation and trained for fake news evaluation. The present research aims to evaluate different machine learning networks and to develop accurate models for identifying fake news and classifying information. The accuracy is measured during training and evaluations and illustrated with graphics that show the good performance of the chosen model for detection.

Keywords: Dataset, Machine Learning, TensorFlow.

1 Introduction

Fake news poses a serious threat to our society. It can influence public perceptions of important issues, encouraging discords, and undermine trust in democratic institutions. Fake news refers to intentionally false or misleading information presented as if it were true. Paskin (2018: 254) defines fake news as “particular news articles that originate either on mainstream media (online or offline) or social media and have no factual basis, but are presented as facts and not satire” [1]. The goal of fake news is often to deceive people, generate clicks or advertising revenue, or influence public opinion [2]. Moreover, fake news can have serious real-life consequences. For example, misinformation about public health can lead to wrong medical decisions, while fake news about politics can influence election outcomes. A fine example that shows the negative and dangerous effects of fake news is the Pizzagate shooting [7]. Social media platforms such as Facebook, Instagram and Twitter use several approaches to identify fake news. Facebook gives some instructions and provides some useful tips on identifying false claims using machine learning algorithms and placing potential fake news articles lower in the news feed [3]. Instagram redirects the user when searching to a special message providing verified information sources, and Twitter ensures that searches result in credible articles. Clickbait-style headlines, misleading mentions, and manipulative comment threads are common vectors that contribute to the widespread dissemination of false information [4], [5], [6].

Machine learning offers promising tools for distinguishing fake news from real ones, with its ability to analyze large amounts of data and identify patterns. Some papers analyse and identify the optimal machine learning algorithm for classifying articles as real or fake news [14]. According to [15], fake news is increasing with the passage of time and there are a few strategies to detect this kind of news. According to [19], various machine learning models have demonstrated competitive performance in fake news detection, particularly when trained on high-quality, labelled datasets.

While this study employs established deep learning models such as RNN, CNN, and BiLSTM, its contribution goes beyond algorithmic experimentation. The focus is on building a complete, functional pipeline that reflects the real-world application of machine learning in fake news detection. This includes sourcing and preprocessing data from 5 publicly available datasets, training and evaluating models, and deploying the most effective one — BiLSTM — through a user-friendly web interface developed with Flask. Unlike prior research that often stops at performance evaluation, this work highlights the importance of usability, accessibility, and practical integration. By bridging technical implementation with real-time user interaction, it demonstrates how machine learning models can be translated into deployable tools. Such an approach has relevance not only for applied research but also for educational settings, where full-stack ML projects offer valuable hands-on learning opportunities.

1.1 Machine Learning

Machine learning algorithms can play a crucial role in identifying fake news. This is done by using various training datasets to refine and improve the algorithms. These datasets help researchers develop new machine learning methods and techniques tailored to spotting misinformation. As explained in [8], machine learning is a branch of artificial intelligence that focuses on building algorithms and models that allow systems to learn from experience and improve over time without being explicitly programmed. These algorithms analyze data, detect patterns, and use that understanding to make predictions or informed decisions. The learning process relies on statistical methods to train models so they can generalize from past data and work effectively with new, unseen information. Machine learning typically falls into two main categories: supervised learning and unsupervised learning [9].

Supervised Learning. In supervised learning, models are trained on labeled data — that is, data where the input features are already matched with the correct output labels. The goal is to teach the model to predict outcomes for new, similar data based on the patterns it has learned.

Unsupervised Learning. This approach deals with data that hasn't been labeled. Here, the model tries to uncover hidden patterns, relationships, or structures within the data without prior knowledge of the correct answers.
Semi-supervised Learning. Sitting between the two extremes, semi-supervised learning uses a mix of labeled and unlabeled data. This combination helps improve model accuracy while reducing the effort and cost required to label large datasets.

Deep Learning. Deep learning is a subfield of machine learning inspired by the structure and functioning of the human brain, particularly through artificial neural networks with many layers. These models can automatically learn data representations, enabling them to identify complex patterns and make highly accurate predictions. Deep learning techniques have proven especially useful in fake news detection, particularly when analyzing the linguistic style and structure of news content [10].

Natural Language Processing (NLP). As stated in [11], NLP refers to a set of computational techniques that allow machines to process, analyze, and understand human language. NLP covers a broad range of tasks, including text classification, information extraction, sentiment analysis, machine translation, and question answering. It combines statistical models, machine learning, and linguistic tools to understand both the meaning and structure of language.

NLP plays a central role in fake news detection by analyzing the syntactic and semantic features of news articles. The development of models like BERT has significantly advanced the field, allowing algorithms to grasp the context and deeper nuances of language. These advancements have enhanced tasks such as sentiment analysis, named entity recognition, and translation. According to [12], research into BERT has also explored how the model learns linguistic information during training, shedding light on its ability to understand syntax and structure.

In this study, we trained neural networks and applied tokenization techniques using relevant datasets, allowing our models to better analyze textual data and detect misinformation.

1.2 Data types in Fake News Detection

In [13], data types used in fake news detection on social media are as follows: content-based data, which refer to the textual and visual content of the news, including news headlines, text, images, and videos; social context data, which are social media data associated with the news, including user profiles, follower-friend relationships, activity logs, and comments; and, network structure data, which name the structural information of the social media network, including user-user relationships and interactions. Different machine learning approaches are better suited to specific data types, and understanding these distinctions is crucial for selecting appropriate techniques in fake news detection. In this study, we trained deep neural network models using structured datasets, which were divided into training (70%), validation (15%), and testing (15%) subsets. Stratified sampling was applied to ensure balanced class distribution across splits. The training set was used for learning patterns and relationships between input features and target labels, the validation set guided model tuning and overfitting detection, and the test set evaluated final model generalization. Each model's performance was assessed using key indicators such as accuracy, loss, and sensitivity to overfitting and underfitting.

2 Research Method and Tools

We have been using datasets that are available on Kaggle and TensorFlow, working with different training models. These models were created through Google Colab in Python programming language and available software libraries, including Tensorflow, Keras, SciKit-Learn, Numpy, Pandas, and more [16][17][18]. The first model, RNN is based on the architecture Long Short-Term Memory and is used to detect sarcasm in text. The second trained model, CNN is used to classify images on Fashion-MNIST dataset. For a practical implementation of machine learning, is developed an application using Flask framework and the model Bidirectional LSTM (BiLSTM).
Beyond model development, this study will focus on the data processing process, model architecture selection, training methods, and model evaluation methods. In addition, common challenges such as overfitting and underfitting, as well as practical issues related to the use of ML will be addressed.

3 Analysis of Training Models in TensorFlow

Machine learning algorithms are widely used nowadays in various AI-powered applications. TensorFlow is a platform that makes it easy to create machine learning models that can run in any environment. Data preparation is a critical step in building machine learning (ML) models with TensorFlow. It involves a series of techniques to transform raw data into a format suitable for training ML models. TensorFlow Keras is a high-level API written in Python that provides a convenient and consistent interface for building and training machine learning models using TensorFlow in an integrated way.

3.1 RNN

TensorFlow Keras includes a powerful preprocessing library that offers a range of useful tools for preparing data for machine learning. One of the key tools is the Tokenizer, which converts words into numerical tokens — a crucial step when working with text data.

When it comes to handling sequential data like text or time series, Recurrent Neural Networks (RNNs) are a strong choice. RNNs are designed to take the order of data into account, maintaining a hidden state that keeps track of information from previous steps in the sequence. This makes them particularly effective for tasks such as language modeling, machine translation, and sentiment analysis.

Long Short-Term Memory (LSTM) networks are a specialized type of recurrent neural network (RNN) designed to address the limitations of standard RNNs in learning long-term dependencies. They incorporate gating mechanisms that regulate the flow of information, enabling the network to retain or discard information as needed and effectively mitigating the vanishing gradient problem during training. LSTMs have a built-in mechanism that helps them remember or forget information as needed, allowing them to capture the broader context of words within a sentence.

In our sarcasm detection model, we used an LSTM network that processes the text in both directions — forward and backward — to better understand the meaning of each word based on its surrounding context. After training the model, we achieved strong performance on the test data with the following results:

Test Loss: 0.2016

Test Accuracy: 92.68%

This means that the model was able to correctly detect sarcasm in tweets with an accuracy of 92.68%, and the relatively low loss value indicates that it made few significant errors during prediction.

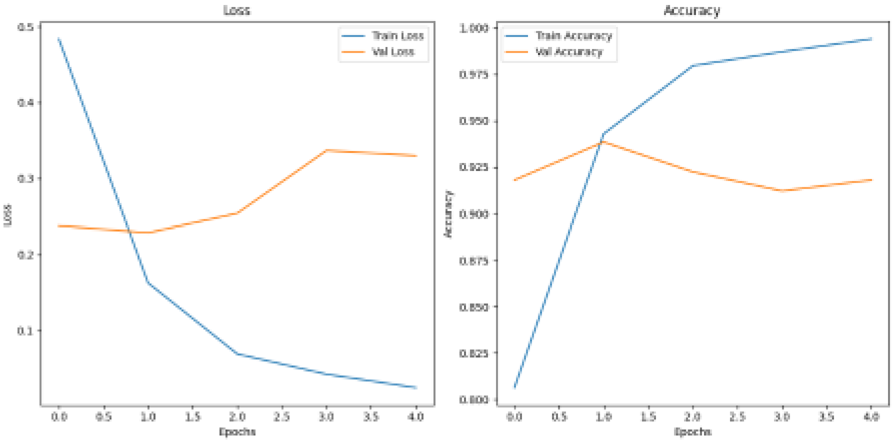


Fig. 1. RNN performance for sarcasm prediction.

3.2 CNN

Convolutional Neural Networks (CNNs) are designed to process data with a grid-like topology, such as images, by recognizing and learning local spatial patterns through convolutional operations. Convolutional filters extract hierarchical features from the input, which are passed through multiple convolutional and pooling layers to progressively reduce spatial dimensions while retaining essential information. CNNs have proven highly effective in computer vision tasks including image classification, object detection, and segmentation. However, their architecture does not inherently capture sequential dependencies, as each input element is processed independently **without temporal or contextual awareness.**

In this study, the CNN model was trained and evaluated on the MNIST dataset, a benchmark collection of grayscale images representing handwritten digits. The model architecture included multiple convolutional and pooling layers for feature extraction and dimensionality reduction, followed by fully connected layers for classification.

The model achieved high accuracy on the training set, confirming its capability to learn salient visual features. Nonetheless, a decline in validation performance across epochs indicated signs of overfitting, suggesting that the model may have started to **memorize training-specific patterns instead of learning generalizable features** applicable to unseen data.

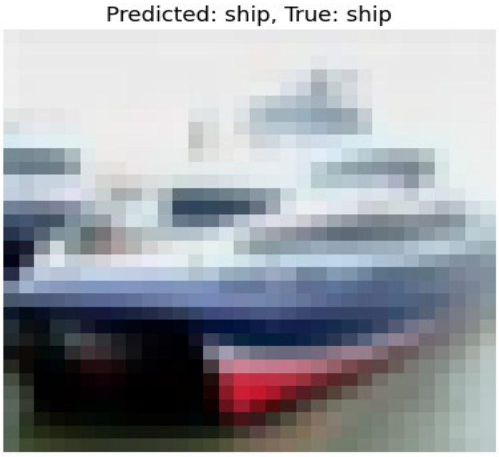


Fig. 2. CNN performance for image prediction.

In contrast, evaluation of the model on the testing data showed an accuracy of 91.09%, confirming that the model has learned to generalize to some extent. Overfitting is a common problem in deep learning and occurs when the model fits the training data very well but does not generalize well to new data.

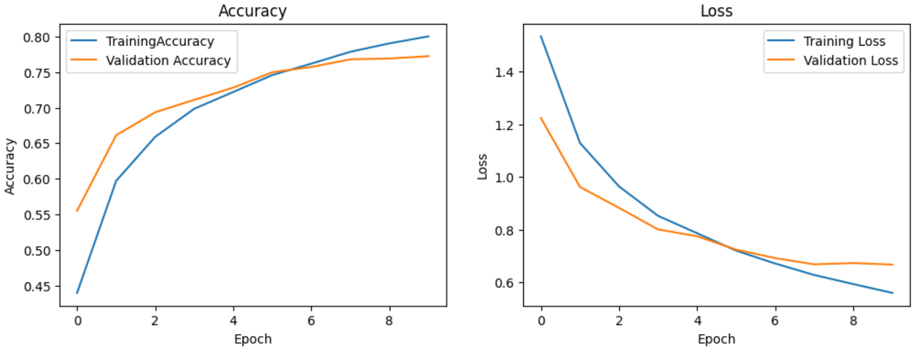


Fig. 3. CNN performance metrics for image classification.

4 Detection of Fake News

4.1 BiLSTM

To tackle the challenge of fake news detection, we implemented a neural network model called Bidirectional Long Short-Term Memory (BiLSTM). **This architecture was selected for its demonstrated effectiveness in modelling sequential text data and capturing complex contextual relationships**, particularly in capturing the nuances of language and understanding word meanings within context. **Unlike standard models, BiLSTM processes sequences in both forward and backward directions, enabling it to learn richer contextual information surrounding each token. This dual perspective supports a more accurate interpretation of sentence- or document-level meaning.**

BiLSTM networks have been shown to outperform unidirectional models in a wide range of natural language processing tasks by effectively capturing both preceding and succeeding word contexts in textual sequences. This bidirectional structure makes BiLSTM particularly suitable for tasks such as sentiment analysis, topic classification, and fake news detection. **Moreover, its adaptable architecture supports a wide range of tuning options, allowing researchers to experiment with configurations and optimizations that enhance task-specific performance.**

Although the BiLSTM model achieved high validation accuracy, we observed signs of overfitting during training. The training accuracy approached nearly 100%, while validation accuracy remained slightly lower and plateaued after a few epochs. This suggests the model may have memorized training examples rather than learning generalizable patterns. To address this, we applied regularization techniques such as dropout with a 0.2 probability, and implemented early stopping based on validation loss. Further experimentation with cross-validation and alternative architectures is planned in the future work to enhance model generalization.

4.2 Fake News Detector in Flask and web implementation

Flask is a widely used Python web framework known for its simplicity, flexibility, and ease of use. It's particularly well-suited for small to medium-sized projects and rapid prototyping. While Flask is minimalist by design, it still offers everything needed to build robust and customizable web applications.

In our fake news detection project, Flask was used to develop a straightforward user interface (UI) where users can enter a news article and receive a prediction about whether the content is likely to be real or fake. The implementation involves the following key steps:

1. **Loading the Model:** The trained TensorFlow/Keras model (fake_news_model.h5) and the associated tokenizer (tokenizer.json) are loaded into the Flask application at startup.

2. **Processing User Input:** When a user submits a news article, the text is processed using the tokenizer to convert it into a numerical format suitable for the model.

3. **Making Predictions:** The processed input is then passed to the trained model, which analyzes the text and returns a prediction indicating the likelihood of the article being fake or real.

4. **Displaying the Results:** The prediction outcome is shown on the web interface, providing the user with a clear response about the nature of the news they submitted.

This approach allows for real-time interaction with the model through a clean and accessible interface, making it easier for users to test and understand how fake news detection works.

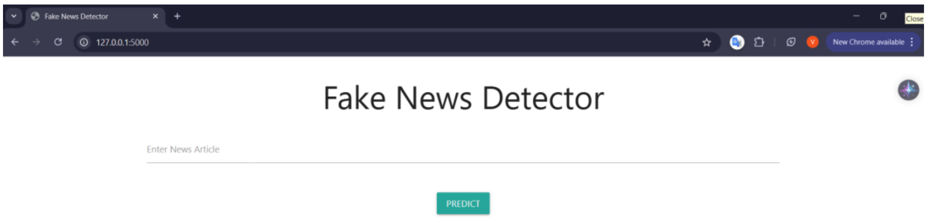
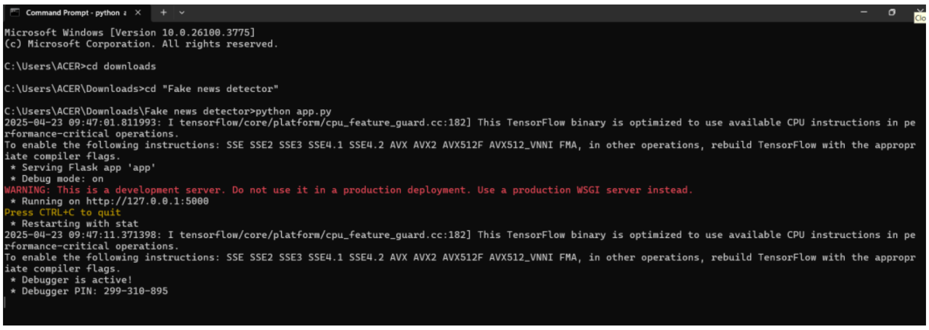


Fig.4. Fake news Web detector in Flask.



```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>cd downloads

C:\Users\ACER\Downloads>cd "Fake news detector"

C:\Users\ACER\Downloads\Fake news detector>python app.py
2025-04-23 09:47:01.811993: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropr
iate compiler flags.
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
2025-04-23 09:47:11.371398: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropr
iate compiler flags.
 * Debugger is active!
 * Debugger PIN: 299-310-895
```

Fig.5. Fake news detect.

4.3 BiLSTM Model Training Results

The BiLSTM model developed for fake news detection delivered strong performance on the evaluation of 5 datasets.

Table 1. Validation Accuracy for 5 training models and 5 epochs.

Data-set (Validation accuracy)	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
LIAR	87.5%	89.5%	91.2%	93.5%	97.5%
ISOT	88.2%	93.4%	95.2%	95.8%	97.6%
FakeNewsNet	91.2%	93.2%	93.8%	96.5%	99%
COVID-19 Fake News	84.5%	88.2%	89.5%	91.4%	92.6%
Real News Dataset	91%	92.5%	93.7%	94.5%	97.6%

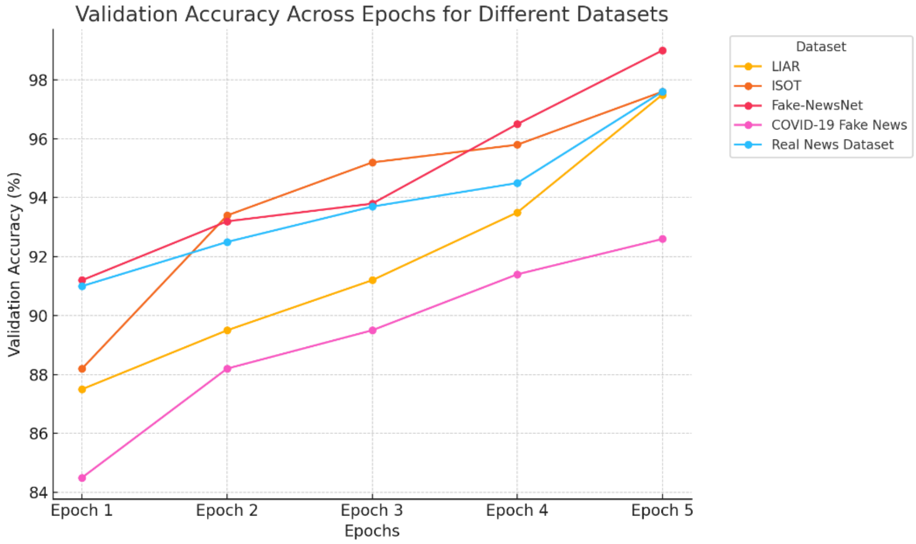


Fig.6. Validation Accuracy across Epochs for different data-sets.

The BiLSTM model achieved the corresponding average validation accuracy for each dataset:
On the LIAR dataset, the model achieved an average validation accuracy of 91.84% over five epochs. The ISOT dataset yielded an average accuracy of 94.04%, suggesting consistent and strong model performance. The FakeNewsNet dataset produced the highest average accuracy at 94.74%, highlighting its compatibility with the model. For the COVID-19 Fake News dataset, the model averaged 89.24%, reflecting the relative difficulty of this domain. On the Real News Dataset, the model maintained a strong average accuracy of 93.86%, indicating effective learning.

The total average validation accuracy of 92.74% across the five datasets and five epochs indicates that the BiLSTM model demonstrates strong and consistent performance in correctly identifying unseen data. This high level of accuracy suggests that the model has effectively learned meaningful patterns from the training data and is able to generalize well across different types of datasets, despite their varying complexities. The steady improvement in accuracy over the epochs reflects a well-structured training process that minimizes overfitting while enhancing predictive capabilities. While the overall performance is robust and suitable for many practical applications, there is still some margin for error, which means that in critical or high-stakes scenarios, additional validation or complementary methods might be necessary to ensure reliability. Overall, these results highlight the effectiveness of the models in the task of fake news detection and related classification problems.

Below we represent the model loss for 5 data-sets among 5 training epochs:

Table 2. Model loss during training in 5 epochs.

Data-set	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
LIAR	12.5%	10.5%	8.8%	6.5%	2.5%
ISOT	11.8%	6.6%	4.8%	4.2%	2.4%
FakeNewsNet	8.8%	6.8%	6.2%	3.5%	1.0%
COVID-19 Fake News	15.5%	11.8%	10.5%	8.6%	7.4%
Real News Dataset	9.0%	7.5%	6.3%	5.5%	2.4%

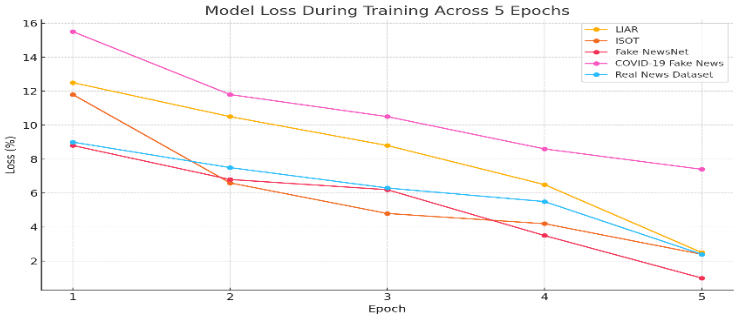


Fig.7. BilSTM Model loss during training in 5 epochs.

During the training of the model across five epochs, the average loss varied by dataset, reflecting different levels of difficulty in learning from the respective data sources. The LIAR dataset had an average loss of 8.16%, indicating moderate training performance. This suggests that the model was able to learn but perhaps struggled with the short, statement-level nature of the LIAR data, which often lacks rich contextual information. The ISOT dataset showed an improved performance, with an average loss of 5.96%. This lower loss suggests that the model adapted more efficiently to the news articles in this dataset, likely due to clearer distinctions between fake and real news. The Fake-NewsNet dataset yielded one of the lowest average losses at 5.26%, reflecting relatively strong performance. This could be due to the structured nature of the dataset and the availability of social context features that aid in classification. The COVID-19 Fake News dataset recorded the highest average loss at 10.76%. This indicates that the model faced challenges distinguishing fake news in the highly dynamic and nuanced COVID-19 information landscape, which may include rapidly changing facts and overlapping narratives. The Real News Dataset had an average loss of 6.14%, suggesting stable model learning, with a performance comparable to that of ISOT and better than LIAR and COVID-19 datasets.

Across all datasets, the average total loss was 7.26%, calculated by averaging the five dataset averages. This value gives a general indication of the model's overall training performance across diverse types of news data.

5. Conclusion and Future Work

Empirical results validate the effectiveness of BiLSTM for binary classification of news content, with performance robust across varying sample distributions. Overall, the developed model offers a promising starting point for fake news detection. Further improvements can be made by addressing overfitting and exploring other model architectures.

Evaluating the performance of a fake news detection model is an essential step in understanding how well it works in practice. To this end, several evaluation methods are used that provide a complete picture of the model's ability to distinguish fake news from real news. The calculated metrics are: accuracy, precision, recall, F1 score and confusion matrix.

Based on the evaluation results, our BiLSTM model has shown good performance in all the metrics mentioned above. The high accuracy, precision, and recall suggest that the model is capable of identifying fake news accurately and minimizing misclassifications. The confusion matrix provides a more detailed look at the types of errors the model makes, helping us identify areas where it can be further improved. The BiLSTM model, trained on a significant amount of news data, achieved high accuracy in identifying fake news. This implies that neural networks, capable of learning complex language patterns, can be successfully used to filter out disinformation.

Careful processing of text data, including tokenization, text cleaning, and text padding, is essential to model performance. It ensures that the model focuses on relevant language features and avoids inconsistencies that can affect accuracy.

The results of our study suggest that the model can be further improved using more advanced NLP techniques. To further improve the model’s performance, we can explore more sophisticated model architectures, experiment with different tuning techniques, and incorporate other data sources. The Flask application can be extended to include additional features, such as explanations for model predictions, news source analysis, or integration with social media platforms.

References

1. *Paskin D. Real or fake news: who knows? J. Soc. Media Soc. 2018;7(2):252–273.* [Google Scholar]
2. Rannard, B.G. How Fake News Plagued 2017. BBC News, 31 December 2017. Available online: https://www.bbc.com/news/world-42487425 (accessed on 10 September 2023).
3. *Sparks, H., Frishberg, and H.: Facebook gives step-by-step instructions on how to spot fake news (2020). https://nypost.com/2020/03/26/facebook-gives-step-by-step-instructions-on-how-to-spot-fake-news/*
4. *Albright J. Welcome to the era of fake news. Media Commun. 2017; 5(2):87. doi: 10.17645/mac.v5i2.977.* [DOI] [Google Scholar][Ref list]
5. *Shao C, Ciampaglia GL, Varol O, Yang K, Flammini A, Menczer F. The spread of low-credibility content by social bots. Nat. Commun. 2018;9(1):4787. doi: 10.1038/s41467-018-06930-7.* [DOI] [PMC free article] [PubMed] [Google Scholar][Ref list]
6. *Chen, Y., Conroy, N.J., Rubin, V.L.: Misleading online content: recognizing clickbait as false news? In: Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection - WMDD 2015, Seattle, Washington, USA, pp. 15–19. ACM Press (2015a). 10.1145/2823465.2823467*[Ref list]
7. Haag, M. (2020, June 27). Gunman in 'Pizzagate' Shooting Is Sentenced to 4 Years in Prison. https://www.Nytimes.Com/#publisher.https://www.nytimes.com/2017/06/22/us/pizzagate-attack-sentence.html
8. Bishop, C.M. Pattern Recognition and Machine Learning; Springer: Berlin/Heidelberg, Germany, 2006.
9. Hastie, T.; Tibshirani, R.; Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
10. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016.
11. Jurafsky, D. Speech & Language Processing; Pearson Education India: Chennai, India, 2000.
12. Jawahar, G.; Sagot, B.; Seddah, D. What Does BERT Learn about the Structure of Language? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
13. Shu, K.; Sliva, A.;Wang, S.; Tang, J.; Liu, H. Fake News Detection on Social Media. SIGKDD Explor. 19, 22–36. (2017).
14. Jumana J., Pratap A., Tijo N., Mony M., Fake News Detection using Python and Machine Learning, 5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Procedia Computer Science, Elsevier (2024).
15. Zhou, X., Zafarani, R., Shu, K., & Liu, H. Fake News: Fundamental theories, detection strategies and challenges. In WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining (pp. 836-837). (WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining). Association for Computing Machinery, Inc.(2019).
16. TensorFlow Platform, https://www.tensorflow.org/

17. Numpy Library, https://numpy.org/
18. Pandas Library, https://pandas.pydata.org/
19. Huang J., Detecting Fake News with Machine Learning, J. Phys.: Conf. Ser. 1693 012158, (2020)

6. Music Genre Classification using Machine Learning Algorithms

Valton Kamberaj[1], Ersan Hamdiu [2], Jusuf Qarkaxhija [3], Shkëlqim Miftari [4], Engin Melekoglu[5]

[1, 2, 3, 4, 5] AAB College, Pristina, Kosovo
valtonikamberaj@gmail.com

Abstract. The connection between technology and the piece of art in our case of music is increasing more and more. Music has its important parts which with the help of technology manage to develop faster, more accurately and better the parts where in the past composers and musicians had a very difficult time. The musical genre plays a major role in the music of any nationality. In this paper we make a classification of the genre based on Albanian artists and their songs. For this work we use three classification algorithms which are: SVM, KNN and ANN. Once we are done with the coding and part of the classification, we derive the results by comparing which algorithm has the highest accuracy. Dataset is available online with some Albanian artists and their songs.

Keywords: Genre, Classification, SVM, KNN, ANN, Music.

1 Introduction

At first glance, computer science and music may appear to be entirely unrelated disciplines. However, there is a strong connection between them, especially in terms of the support that technology—both hardware and software—has provided and continues to provide in solving complex problems and streamlining various tasks. Music plays a significant role in people's daily lives. It serves as a vital element that unites individuals and reflects the identity of communities, often indicated by the types of music they predominantly listen to**Invalid source specified.**.

The advancement of computer science and technology has also contributed significantly to the field of music, particularly in the classification of musical genres. Programming, algorithms, and other computational methods play a crucial role in supporting this process by enabling the creation of efficient and accurate systems for categorizing music and understanding its relationship with humans **Invalid source specified.**. A large number of musicians and emerging artists release new music on a daily basis, often experimenting with various styles or blending elements from diverse global genres in an effort to introduce innovation into the music market. In contemporary music culture, a noticeable trend is the dominance of commercialized content, where the artistic value of many songs is diminished to the point that genre classification becomes increasingly difficult. This tendency is largely driven by the pursuit of quick commercial success, frequently at the expense of musical depth and authenticity **Invalid source specified.**.

The classification of music genres has been a continuous focus for numerous scholars across various studies, with some developing algorithms and methodological approaches tailored specifically to the musical traditions, genres, and cultural values of their respective countries.

The rationale behind employing genre classification within a specific country is to preserve national cultural values by categorizing local songs within the appropriate domestic genres. This approach helps prevent the blending of indigenous music with foreign or cross-genre influences, thereby safeguarding its authenticity and cultural identity from being absorbed into external stylistic frameworks.

The paper will classify the genre with three main algorithms to measure the accuracy of each and make comparisons between them. In the second part we will see the

part of the revised papers from which we are based to do the work, the third part includes the methodology used which clearly explains the three algorithms and their use.

The fourth part includes the results achieved graphs and tables, while the paper will conclude with the conclusions in the fifth part.

To verify the system and the results of this paper we have built 3 research questions which are the main objectives as follows:

To verify the system and the results of this paper we have built 3 research questions which are the main objectives as follows:

RQ1: Which of the algorithms will be more accurate?

RQ2: Will the accuracy be over 90% of any of the algorithms?

2 Related Work

In recent years, the music industry has experienced significant transformations that have influenced both the creative process and modes of expression. Beyond its unifying power, music serves as a medium through which diverse cultures can be explored and understood. Consequently, the classification of musical genres plays a vital role not only in contextualizing these cultural and social dimensions but also in systematically addressing the varied preferences and expectations of listeners.

In their study [4], the authors proposed a novel CNN-based model by conducting a comparative analysis with existing models developed for genre classification. Utilizing the GTZAN dataset, they introduced a framework that integrates multiple input models alongside audio mel-spectrograms fed into the convolutional neural network. The approach achieved a high classification accuracy of 91%, enabling effective genre identification. However, it is important to note that certain genres, such as country and rock, exhibited overlapping characteristics that led to misclassification. In contrast, genres like traditional and blues were more distinctly represented and prioritized within the model, resulting in higher classification accuracy for those categories.

In their paper [5], the authors assert that music genre classification is a crucial and practical field within Music Information Retrieval (MIR). The use of deep learning techniques has become increasingly prevalent for genre classification, driven by two primary factors highlighted by the authors: the first is to eliminate the need for manual feature extraction from audio signals, and the second is that deep learning models' hierarchical structure aligns well with the stratified nature of musical composition, both in time and frequency domains. The authors discuss how various studies and projects employ Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or a combination of both for genre classification. However, a challenge in these approaches lies in learning dependencies between distant points within a sequence. To overcome this limitation, the authors propose a Classification Transformer based on Natural Language Processing (NLP) techniques. This model aims to better capture relationships between different audio frames, leading to improved performance in music genre classification.

In their work [6], the authors explore methods for genre classification based on the textual content of music. They argue that musical pieces can be described as a collection of texts, with the sound being influenced by the timbre. A subset of sound textures is often selected to represent the overall track. The authors demonstrate how the choice of these textures within a musical piece impacts the automatic classification of its genre. According to the study, texture selection is performed using the K-Means algorithm, which is designed to identify distinct textures within different sections of the audio. The authors conclude that the texture selection model presented in the study significantly improves the performance of genre classification.

In their paper [7], the authors highlight the growing importance of developing models to learn the similarity between music tracks based on audio media files, a task that is increasingly relevant in the entertainment industry. They propose a novel genre classification model based on metric learning, with the primary goal of learning a personalized metric tailored to each client. This model employs a structured prediction approach on a set of MP3 audio files, which represent various music genres aligned with the individual preferences of the user. The acoustic information extraction in this study is achieved using Mel-Frequency Cepstral Coefficients (MFCCs), followed by dimensionality reduction through Principal Component Analysis (PCA). The validity of the proposed model is rigorously tested through multiple experiments, comparing the results with basic algorithms such as K-means and Soft Margin.

In their study [8], the authors focus on music genre classification for Nigerian traditional music. To develop their model, they introduce a new dataset called the ORIN dataset, which contains 478 traditional Nigerian songs across five different genres. Timbre quality and rhythm are extracted from 30-second segments of each song using Python. The songs in the ORIN dataset were then trained using four different classifiers: k-Nearest Neighbor, Support Vector Machine, Extreme Gradient Boosting, and Random Forest. According to the authors, the XGBoost classifier yielded the best results, achieving an accuracy of 81.94% and a recall of 84.57%. The model was further evaluated through manual analysis, where additional investigations on individual genres revealed similarities in timbre properties across several genres.

In their study [9], the authors explore genre classification through the use of genre labels, presenting a novel approach for music genre classification. They experiment with Brazilian songs, making a significant contribution to the music classification field in that country. The authors regard this type of classification as a challenge within Natural Language Processing (NLP). The dataset used consists of 138,368 songs, distributed across 14 genres. To perform the classification, they employed Support Vector Machine (SVM), Random Forest, and Short Term Double Direction Memory Network (STDDMN), along with various embedded word techniques. According to the results presented in the paper, this approach outperforms other models, with an average improvement of 0.48.

In the study by [10], the authors demonstrate the use of speech features (POS) extracted from song lyrics, combined with three different classification techniques: k-Closest-Neighbor, Naive Bayes, and Support Vector Machines (SVM). They applied this approach to classify 600 English songs, which were categorized by both genre and mood. Additionally, two statistical measures, document frequency and inverse document frequency, were utilized as features for genre classification. The model achieved an accuracy of 70% in classifying the songs based on these features.

In the study presented in [11], the authors discuss the increasing application of Convolutional Neural Networks (CNNs) across various fields, particularly in the multimedia industry, with a focus on music genre classification. The authors combine CNN with Recurrent Neural Network (RNN) architecture to enhance genre classification accuracy. Mel-Frequency Cepstrum (MFC) coefficients were used as sound sample vectors, and Librosa was employed to convert audio files into MFC representations, aiming to create a model that approximates human auditory perception. The results achieved with this approach, using CNN in combination with MFC, yielded an accuracy of 43%, with the authors expressing optimism that this method can be further refined and improved in future work.

In the study by the authors [12], they explore the use of deep learning for music genre classification. Their contribution stems from the growing need to categorize music, especially given the vast increase in the number of songs globally and the incorporation of various new values in the music industry. The technique employed in this study is word2vec, specifically using the skip-gram model to identify songs with similar contexts for recommendation purposes.

In the next paper [13] did the research to find a better algorithm which makes the classification of the musical genre compared to the previous ones. By doing a lot of testing and experimenting and comparing some algorithms, the main reason was to find the greatest accuracy. Some of the compared models are trained with mel-spectogram while some only with song spectrogram. From all the tested models, the results have concluded that the convolutional neural network has given the greatest accuracy in the classification of the music genre compared to other models.

3 Methodology

Quantitative methods will be used for this study. The data with which we will work with them will be text of different songs which will be ready for the next stages. Classification based on textual data was done using 3 algorithms which are quite familiar to perform this work. After finishing the work from the 3 algorithms at the end to compare the accuracy values of each concluding in the one which is more efficient. The final product will enable through it to be filtered and to find the melodic line in which genre they belong.

In the first phase we will collect the lyrics of Albanian songs using a crawler which will navigate through the largest websites where these lyrics are. For each musical genre we will collect the lyrics using this method. Crawler will be built using the Python script "Scrapy" which is a library that retrieves data from the web. From the websites we will be able to get the title of the song, the lyrics and the name of the artist. The main genres of our music will be taken from pages of styles, which lists the songs in different categories for each artist. The genres that will be selected will be 4-5 different and will be the ones that unify us from other world music.

The next stage involves the preparation of the received data, in our case the lyrics by clearing it, and eliminating all punctuation marks. The resulting text is then presented as a vector to provide an embedded word pattern. Using the classic SVM and Random Forest learning algorithm, for each song a representative in vector form must be generated, calculating each vector and each word on their average. Creating a model such as BLSTM used by **Invalid source specified.**, makes the presentation of these vectors. Then the last stage is combining the SVM algorithm, the Random Forest and the created algorithm (e.x BLSTM) to divide the texts into genres. Algorithms that we will use and compare for classification are: SVM, KNN and ANN.

3.1 SVM Model

One of the main models which is used in model training the model to analyze data and identify models is Support Vecrot Machines (SVM). The results that this algorithm gives are generally satisfactory. This model is used to solve problems related to regression and classification, but especially for the latter. The SVM performs the classification process by selecting the possible hyperplan and then applies a procedure to make the partition boundaries near the hyperplan maximum that provides the best accuracy **Invalid source specified. Invalid source specified.**.

$$xiw + b \geq +1, kur\ yi = +1 \qquad (1)$$

$$\mathrm{xiw + b \leq -1, kur\ yi = -1} \qquad (2)$$

In this context, x and w represent vectors, and b is the bias term. The hyperplane is defined as $wT{\cdot}x$=0.

In Support Vector Machines (SVM), the optimal hyperplane is the one that maximizes the margin between the classes. To achieve this, the margin can be increased by minimizing ||w||, which is formulated as follows:

$$minw, b = 21 \parallel w \parallel 2 \qquad (3)$$

In cases where the situation is nonlinear, the data may not be classified accurately. To assess the extent of classification errors, the slack variable ζi is introduced, where i=1,2,3,…,n, and is related to the regularization parameter C. When C is large, the error rate tends to increase, whereas a smaller C can potentially reduce the error rate. This relationship is described by the following equation (3):

$$w, b, \zeta min = 21 \parallel w \parallel 2 + Ci = 0 \sum n \zeta i \qquad (4)$$

In Support Vector Machines (SVM), kernel functions play a crucial role in defining the width of the hyperplane margin. To enhance the predictive performance of the algorithm, various methods can be applied, one of which involves modifying the kernel function itself. The kernel trick is a mathematical technique used to transform non-linearly separable data into a higher-dimensional space, where the data becomes linearly separable. This transformation allows SVM to construct an optimal hyperplane in the new feature space, thereby improving classification accuracy.

$$K(m, n) = \big(f(m), f(n)\big) \qquad (5)$$

In this context, K(m,n) denotes the kernel function, where m and n are input vectors from a certain dimensional space, and f is a mapping function that transforms data into a higher-dimensional space.

Based on the mathematical formulation of the kernel trick, several types of kernel functions are commonly used: the Linear Kernel, the Radial Basis Function (RBF) Kernel, and the Polynomial Kernel. The Linear Kernel operates similarly to the standard dot product between two vectors and is defined as:

$$K(x1, x2) = x1 \cdot x2 \qquad (6)$$

The polynomial kernel function is like a Linear kernel but it has a polynomial degree associated with it.

$$K(x1, x2)$$
$$= (x1 \cdot x2c)n \qquad (7)$$

Another form of the kernel is, Radial Bias Function and it is represented in (8):

$$K(x1, x2)$$
$$= exp(-\gamma \parallel x1 - x2 \parallel 2) \qquad (8)$$

where $\|x1 - x2\|$ is a Euclidian distance and γ describes the decision region. In SVM, Margin (M) and Misclassifcation (MCR) are directly proportional. If we increase the margin size of the hyperplane then the misclassification rate will be increased. Cost parameter "C" helps the SVM to find out the margin for the hyperplane. To reduce the misclassification rate, we should decrease the margin size so that all the vectors will be classified into theirs the corresponding space. The relationships between "MCR", "M" and "C" was represented in (9) and (10) [14], [15].

$$MCR \propto M \qquad (9)$$

$$M \propto$$
$$C1 \qquad (10)$$

3.2 KNN Algorithm

We must first present the distance measurements that will be used when two items are similar before describing how the kNN method works. This distance aids us in calculating the degree of similarity. The distances utilized in this approach by Evelyn Fix and Joseph Hodges, the algorithm's creators in 1951, were three upline distances.

1. Manhattan didianstance
2. Minkowski distance

$$d(x, y) = i$$
$$= 1\sum Nxi2 - yi2 \qquad (11)$$

The K-Nearest Neighbor algorithm is a distance-based classification system, meaning that the closer two locations are, the more similar they are. The kNN method treats columns as a single dimension. The steps of the kNN algorithm are as follows:

1. For each value of k, adjust a kNN classifier.
2. Make predictions based on that model.
3. Using model predictions, calculate and evaluate a performance gauge.
4. Compare the outcomes and choose the one with the least amount of inaccuracy **Invalid source specified.**.

3.3 ANN Model

Neural networks are a type of Artificial Intelligence (AI) that is built on the human brain's inspiration**Invalid source specified.**. A program can learn from examples and construct an internal set of rules for classifying diverse inputs using neural network techniques. This set of neurons or units is responsible for all of a neural network's processes [17]. Because each neuron is its own communication device, its operation is quite straightforward. One unit's function is to simply receive data from other units and generate an output value based on the inputs it gets, which it then distributes to other units. Artificial neural networks are made up of layers of neurons that process data via dynamic state responses to external inputs **Invalid source specified.**. A supervised learning example is an artificial neural network **Invalid source specified.**. Artificial neural networks (ANNs) have the ability to predict new observations based on previous ones. For classification, clustering, feature mining, prediction, and pattern recognition, neural networks are used. The Multilayer Perceptron (MLP), which uses a nonlinear activation function to produce outputs, is one of the most widely used Neural Networks **Invalid source specified.**. The activation function includes a sigmoid function in the hidden layer (f(x) = 1 / (1 + exp (-x)) and a linear function in the output layer (fj(x) = p i=1wijxi, where xi's are predictor variables and wij's are input weights). The MLP's functional form can be expressed as:

$$y_k = f\left(\sum_{i=1}^{N} w_{ji} x_i + b_j \right) \qquad (12)$$

where xi is the previous layer's i-th nodal value, yj is the current layer's j-th nodal value, bj is the bias of the present layer's j-th node, wji is a weight connecting xi and yj, N is the previous layer's number of nodes, and f is the present layer's activation function [20].

4 Results and Discussions

Since we proposed three methods for classifying and predicting different genres of music let’s look at a graphical representation of the correlations between variables:

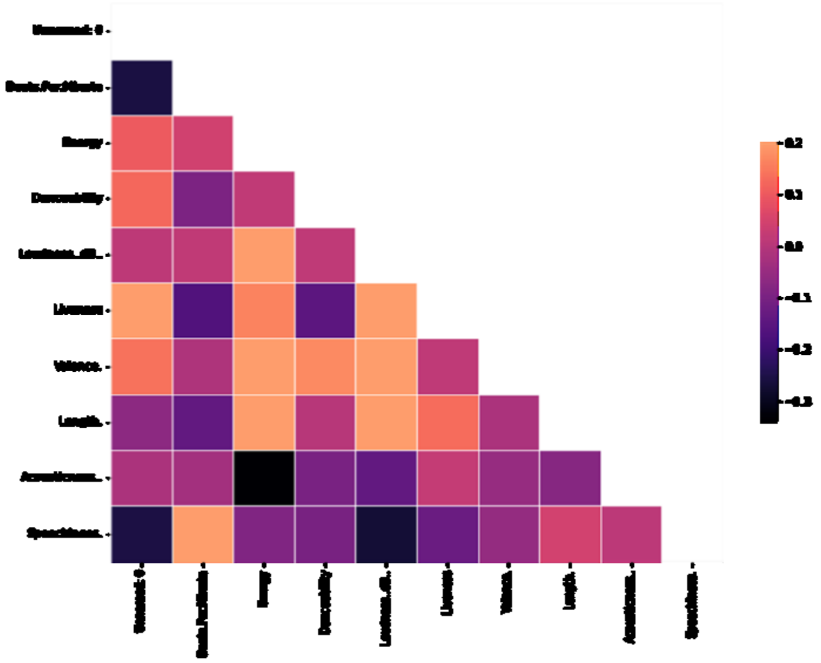


Fig. 4. Numerical features correlation (Pearson's).

There is a very strong relationship between some of the variables which shows that the independent variables affect each other. Thus, in order to have an even higher accuracy, those musical genres that are correlated with each other should be excluded from the forecast, according to statistical studies, they break the model. Accusticnes for example with energy have a very strong connection with each other. Once we have a general overview of the relationship between the variables and their impact, we derive the results of the classifications including the result of each method separately and the application of each algorithm. Table 1 shows the analysis of the three analyzed algorithms for predicting music genres based on accuracy, sensitivity, and specificity.

Table 3. Comparison of classifcation models.

Algorithm	Accuracy	Sensitivity	Specificity
SVM	80%	80%	79.54%
kNN	86%	87.93%	84%
ANN	89%	93%	87.73%

Based on the results, it is clear that the ANN algorithm has higher accuracy compared to other algorithms. In all tests with different data set sizes, ANN offers higher sensitivity and specificity compared to other algorithms. The graph below which refers to the ANN algorithm shows the efficiency of the algorithm. It is clear that the curve for true positive rate is increasing and significantly away from the statistically defined limit.

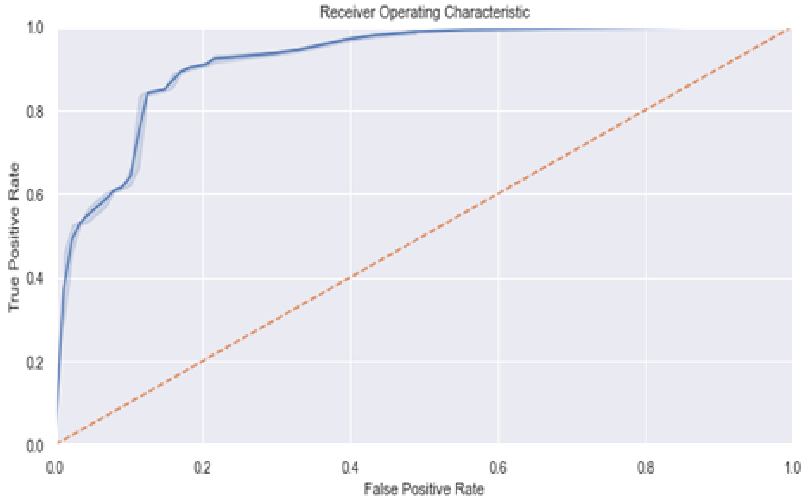


Fig. 5. ROC curve.

From the above results we can answer research questions by concluding as follows:

AQ1: The best results compared to the other two algorithms were given by ANN, where for this analyzed dataset, it was given an accuracy of 89%, where it is not very good any number because it should be over 90% because there are works that have reached an accuracy of up to 98% but we believe that in the future a better number will be achieved.

AQ2: As we showed above, none of the algorithms gave an accuracy of over 90%, but looking at the results of the 3 techniques we notice that it is not a poor result, on the contrary it is an average result, while the 3rd ANN method is a promising result and close to 90%.

5 Conclusion

In this paper were analyzed and used the 3 main algorithms for classification and in our case for the classification of the musical genre. For some Albanian songs of different authors and different genres which we assigned are not coming out of our language we managed to make a classification through algorithms: SVM, KNN and ANN and making the comparison between them seeing the most accurate of them has turned out to be ANN with an accuracy 89%. The other 2 classifiers also achieved a not insignificant accuracy KNN with 86% while SVM with 80%. Graphs of accuracy and correlation analysis are done. From the first work, above we can conclude that very few or maybe no classifiers in the Albanian language and we have not encountered any such work, also datasets are very deficient in this regard.

Future work: In the future to use a larger corpus of music genres and artists in the Albanian language in order to improve the classification for large numbers and also to achieve even greater accuracy.

References

1. V. V. S. V. M. Anand, "Music Genre Classification with Deep Learning," *SCOPUS,* pp. 1-6, 2020.
2. G. H. John Peterson, "Integrating Computer Science into Music Education," *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education,* pp. 459-464, 2017.
3. A. L. P. a. W. V. L. C. N. Silla, "Music education meets computer science and engineering education," *2016 IEEE Frontiers in Education Conference (FIE),,* pp. 1-7, 2017.
4. A. S. K. &. S. S. Ghildiyal, "Music Genre Classification using Machine Learning," *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA).,* pp. 1368-1372, 2020.
5. Y. C. Y. &. Z. J. Zhuang, "Music Genre Classification with Transformer Classifier," *Proceedings of the 2020 4th International Conference on Digital Signal Processing.,* pp. 155-159.
6. J. H. &. T. T. F. Foleiss, "Texture selection for automatic music genre classification," *Applied Soft Computing 106127, ELSEVIER,* pp. 106-127, 2020.
7. A. C. M. d. C. M. A. N. &. N. R. F. Silva, "A Music Classification Model based on Metric Learning Applied to MP3 Audio Files," *Expert Systems with Applications, 113071,* pp. 1-33, 2019.
8. S. O. A. S. A. &. O. A. B. Folorunso, "Dissecting the genre of Nigerian music with machine learning models. doi:(https://doi.org/10.1016/," *Journal of King Saud University - Computer and Information Sciences,* pp. 1-14, 2020.
9. R. d. S. R. C. C. L. H. &. B. S. D. J. De Araújo Lima, "Brazilian Lyrics-Based Music Genre Classification Using a BLSTM Network," *Lecture Notes in Computer Science, Springer,* p. 525–534, 2020.
10. T. D. S. A. L. Ying, "Genre and mood classification using lyrics features," *2012 International Conference on Information Retrieval & Knowledge,* pp. 260-263, 2012.
11. M. I. P.-C. C. D.-M. N. C.-N. K. Yu-Huei Cheng, "Automatic Music Genre Classification," pp. 1-5, 2021.

12. A. P. a. S. R. A. Budhrani, "Music2Vec: Music Genre Classification and Recommendation System," *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA),,* 1406-1411.
13. K. A. S. A. N. S. H. V. K. S. Snigdha Chillara, "Music Genre Classification using machine learning algorithms: A comparison," *International Research Journal of Engineering and Technology (IRJET),* pp. 851-858, 2019.
14. S. M. Dewan A, "Prediction of heart disease using a hybrid technique in data mining classification.," *2nd international conference on computing for sustainable global development (INDIACom).,* p. 704–706, 2015.
15. A. Dwivedi, "Performance evaluation of different machine learning techniques for prediction of heart disease," *Neural Comput & Applic 29,* p. 685–693, 2018.
16. L. A. e. al., "An Optimized Stacked Support Vector Machines Based Expert System for the Effective Prediction," *in IEEE Access, vol. 7,* pp. 54007-54014, 2019.
17. N. &. A.-N. S. El_Jerjawi, "Diabetes Prediction Using Artificial Neural Network," *Journal of Advanced Science. 124,* pp. 1-10, 2018.
18. Q. Q. K. L. Y. Y. D. J. Y. &. T. H. Zou, "Predicting Diabetes Mellitus With Machine Learning Techniques. .," *Frontiers in genetics, 9, 515,* pp. 1-7, 2018.
19. D. K. Rajni Bala, "Classification Using ANN: A Review," *International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 13, Number 7 (2017),* pp. 1811-1820, 2017.
20. T. A. a. S. V. C. Kumbhare, "An Overview of Association Rule Mining Algorithms," 2014.

7. A Comparative Analysis of Human and Machine-Written Scientific Texts

Erika B. Varga[1] and Attila Baksa[2]

[1,2] University of Miskolc, Faculty of Mechanical Engineering and Informatics, Hungary

erika.b.varga@uni-miskolc.hu[1], attila.baksa@uni-miskolc.hu[2]

Abstract. Artificial intelligence tools based on large language models (LLMs), which are designed to generate and process textual data have been widely adopted in recent years. These tools are no longer used exclusively by students as primary aids in writing assignments but are now frequently applied by researchers in the creation of scientific papers as well. This poses new challenges for text analysis, content verification, and plagiarism detection techniques. A key question arises: is it possible to determine the origin or authorship of a given scientific paper? Can we assess whether its content stems from reliable sources? This study seeks to address such questions through linguistic and computational approaches. By comparing four distinct text corpora, the research aims to investigate to what extent AI-generated scientific texts can be distinguished from those written by human authors. The analysis is built around two main comparisons: (1) original publications written by native English-speaking authors and AI-generated texts on the same topics; and (2) translations of publications by non-native authors and AI-generated texts on the same topics. The working hypothesis suggests that more substantial stylistic and lexical differences will be found in the first case. The results contribute to improving the reliability of AI-generated text detection, which is an essential step for maintaining academic integrity and ethical standards.

Keywords: Natural Language Processing, AI-Generated Text, Human-AI Text Comparison, Authorship Detection.

1 Introduction

In recent years, the academic landscape of the STEM disciplines has undergone a notable transformation: not only in terms of the content produced, but also in the language used to convey scientific ideas. Increasingly, reviewers and academic supervisors are confronted with manuscripts that exhibit an unusually sophisticated and stylistically refined use of English. This shift in tone and expression diverges from the traditionally concise and technical style characteristic of scientific communication.

This trend has become particularly noticeable since the emergence and widespread use of large language models (LLMs) and AI-powered writing tools such as ChatGPT.

The growing reliance on AI tools raises fundamental questions about the authenticity and authorship of scientific texts. When confronted with highly polished language, particularly in papers authored by non-native English speakers, academic readers are often left to wonder: To what extent did the authors rely on AI assistance? Was AI used merely to enhance grammar and fluency, or did it contribute to the actual generation of content? These questions are not merely speculative; they address key issues of academic integrity, originality, and the ethical use of technology in knowledge production.

While AI detectors have been developed to help identify machine-generated content, their current performance remains inconsistent. Existing detection tools struggle to distinguish between texts that are AI-enhanced and those that are entirely AI-produced, especially as LLMs become increasingly capable of mimicking human writing. As such, there is a pressing need for deeper, linguistically informed investigations into how AI-generated texts differ from their human-written counterparts for formulating clearer academic policies around AI usage.

This study aims to contribute to this emerging field by analyzing the lexical and stylistic characteristics of scientific texts produced by humans and AI. We focus specifically on two parallel corpora: one consisting of scientific texts written by native English-speaking authors before the widespread adoption of LLMs, and another comprising papers authored by non-native speakers in recent years, many of which exhibit signs of AI-assisted translation or linguistic refinement. To complement these, we generated AI-produced versions of all selected topics using ChatGPT-4's deep research capabilities, creating aligned pairs for comparison.

We compare each human and AI-assisted text with its corresponding AI-generated version using lexical overlap metrics such as Jaccard and Cosine similarity, together with statistical tests and effect size computations to assess significant differences in vocabulary usage patterns. Through this approach, we aim to identify linguistic markers that may indicate the presence of AI-generated content and evaluate the extent to which AI can emulate human-like stylistic features in scientific writing. Our research also offers practical insights for reviewers, educators, and institutions aiming to uphold ethical standards in scholarly communication.

2 Related Work

The widespread adoption of large language models (LLMs) in academic and educational settings has raised significant concerns regarding authorship, originality, and academic integrity. In response to these concerns, an increasing number of studies have been conducted to investigate the distinctions between AI-generated and human-authored texts. These works examine linguistic, stylistic, and structural characteristics, as well as the performance of various machine learning models and AI-detection tools across different domains, languages, and writing styles.

The study by Elkhatat et al. [1] aims to assess the effectiveness of five AI content detection tools -OpenAI Classifier, Writer, Copyleaks, GPTZero, and CrossPlag - in distinguishing between text generated by ChatGPT models 3.5 and 4, and human-written content. 30 AI-generated paragraphs (15 from ChatGPT 3.5, 15 from ChatGPT 4) and 5 human-written control paragraphs were analyzed on the topic of cooling towers. The AI content detection tools' performance was evaluated using standard classification metrics: sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV). The key results were that all tools performed more accurately in detecting content generated by ChatGPT 3.5 than GTP 4, and human-written text was often misclassified as AI-generated.

Niful et al. [2] developed a machine learning-based model to distinguish between texts written by humans and those generated by ChatGPT 3.5. They worked with 10,000 samples (5,204 human-written texts from Quora/CNN and 4,796 AI-generated), compared 11 machine learning and deep learning algorithms and assessed the impact of data preprocessing techniques on model performance. In this experiment, the Extremely Randomized Trees (ERT) model outperformed all others, achieving 77% accuracy. Surprisingly, they found that removing stop-words reduced performance. This indicates that some stop-words may carry stylistic signals useful for distinguishing AI-generated content.

Hakam et al. [3] compared the quality and detectability of human-written versus AI-generated scientific abstracts in the field of orthopedics and sports medicine. They assessed whether human evaluators and AI-detection software can reliably distinguish between the two. They selected 5 human-written abstracts on meniscal injuries from top-tier journals, then used ChatGPT and You.com to generate rewritten versions of the 5 abstracts and 10 entirely new abstracts. These were evaluated by 4 researchers and an unnamed AI-detection platform, respectively. The authors found that neither humans nor the AI-detection tool can reliably detect AI-generated scientific texts.

The study of Durak et al. [4] aimed to compare the linguistic and semantic features of discussion-based articles written by university students versus those generated by AI models such as ChatGPT, Gemini, and BingAI to identify distinctive linguistic patterns that separate human and AI-generated content. The key findings include that human-written texts have more unique words, longer sentence length and higher syntactic complexity. The authors also aimed at evaluating the performance of machine learning models in classifying the origin of texts. Their general finding was that ChatGPT texts were hardest to distinguish from human texts, with an accuracy of only 88%.

The research objective of Schaaff et al. [5] was to investigate methods to classify texts as human-generated, AI-generated, or AI-rephrased, focusing on multiple languages and different domains. They have 1) created a multilingual corpus of human, AI-generated, and AI-rephrased texts including 4 languages; 2) extracted and evaluated 37 textual features for classifying the texts; and 3) benchmarked their classifiers against GPTZero and ZeroGPT. Their Random Forest and XGBoost classifiers detected AI-generated text with 98% accuracy. AI-rephrased texts are more challenging to detect since they appear more human-like. This paper reports the best F1-score reached around 78%, significantly outperforming GPTZero (F1 ≈ 28%).

While most studies concentrate on distinguishing fully AI-generated texts from human-authored content, fewer have examined AI-rephrased texts. These are originally written by a human but then reworded or restructured by an AI model. These pose a greater challenge to detection due to their stronger similarity to authentic human writing. In this study, we focus specifically on identifying distinctive linguistic features that can effectively differentiate AI-translated text from original human-authored content.

3 Research Method

3.1 Corpus collection and preparation

In our experimental framework, 2 sets of long academic texts (> 3000 words) were collected from the Computer Science domain. The first set contains papers written by native English-speaking authors before the release of LLMs in 2017 (denoted as HT for human-written texts). These writings were selected from IEEE Xplore, the digital library for accessing scientific content published by the Institute of Electrical and Electronics Engineers (IEEE). To guarantee the quality of English language use and no use of AI in text preparation we have applied the following criteria when filtering conference and journal papers:

- published between 2000 and 2015, and
- at least one of the authors has British or United States nationality.

Table 4. Selected human-authored papers in order of appearance.

ID	Authors	Title	Year of publ.	Citation
HT1	BD. Fulcher and NS. Jones	Highly Comparative Feature-Based Time-Series Classification	2014	[6]
HT2	S. Poslad, SE. Middleton, F. Chaves, R. Tao, O. Necmioglu and U. Bügel	A Semantic IoT Early Warning System for Natural Environment Crisis Management	2015	[7]
HT3	B. Harrison, SG. Ware, MW. Fendt and DL. Roberts	A Survey and Analysis of Techniques for Player Behavior Prediction in Massively Multiplayer Online Role-Playing Games	2015	[8]

HT4	W. Blewitt, M. Brook, C. Sharp, G. Ushaw and G. Morgan	Toward Consistency of State in MMOGs Through Semantically Aware Contention Management	2015	[9]
HT5	C. Demmans Epp and S. Bull	Uncertainty Representation in Visualizations of Learning Analytics for Learners: Current Approaches and Opportunities	2015	[10]
HT6	A. Bagnall, J. Lines, J. Hills and A. Bostrom	Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles	2015	[11]

From the available collection, 6 papers were selected as gold standard human-authored English texts listed in Table 1.

The other set in Table 2 includes papers written in English by non-native speakers in the new AI era (denoted by TT for translated texts). That is, these writings were created after 2020 with a moderate help of machine translation and AI in correcting linguistic errors.

Table 2. Selected translated papers in order of appearance.

ID	Authors	Title	Year of publ.	Citation
TT1	J. Alshboul, H.A.A. Ghanim and E. Baksa-Varga	Semantic Modeling for Learning Materials in E-Tutor Systems	2021	[12]
TT2	J. Alshboul and E. Baksa-Varga	A Review of Automatic Question Generation in Teaching Programming	2022	[13]
TT3	J. Alshboul and E. Baksa-Varga	A Hybrid Approach for Automatic Question Generation from Program Codes	2024	[14]
TT4	J. Alshboul and E. Baksa-Varga	A Generator-Evaluator Framework for Automatic Question Generation from Program Codes	working draft	
TT5	J. Alshboul and E. Baksa-Varga	Ontology-based Automatic Learning Materials Generation for Python Programming	working draft	
TT6	J. Alshboul and E. Baksa-Varga	Enhancing Content-Based Recommendation Systems with Ontology Integration for Improved Contextual Relevance	working draft	

The two sets of texts are not directly comparable, because they cover different topics. To address this issue, we have generated texts using ChatGPT 4's deep research feature for each of the topics covered by the selected papers (denoted as AI for AI-generated texts). Tables 3 and 4 summarize the basic characteristics of all texts in the corpus.

Table 3. Basic characteristics of human-written and AI text pairs.

Text ID	Num. of tokens	Num. of unique lemmas	Num. of sentenc es	Avg. sent. length *tokens*	Avg. sent. length *words*	Text ID	Num. of tokens	Num. of unique lemmas	Num. of sentenc es	Avg. sent. length *tokens*	Avg. sent. length *words*
HT1	10,765	689	316	34.07	26.22	AI1	5,700	694	205	27.81	22.73
HT2	9,491	778	369	25.72	20.85	AI2	5,490	778	225	24.40	20.38
HT3	13,503	955	555	24.33	20.83	AI3	6,301	832	245	25.73	21.38
HT4	10,385	782	367	28.30	22.78	AI4	12,440	1,081	447	27.83	22.68
HT5	14,722	882	562	26.21	20.51	AI5	7,819	900	306	25.56	21.36
HT6	11,903	708	496	24.01	17.34	AI6	5,700	694	205	27.81	22.73
AVG	**11,794**	**799**	**444.17**	**27.11**	**21.42**	**AVG**	**7,241.7**	**829.83**	**272.17**	**26.52**	**21.88**

Table 4. Basic characteristics of translated and AI text pairs.

Text ID	Num. of tokens	Num. of unique lemmas	Num. of sentenc es	Avg. sent. length *tokens*	Avg. sent. length *words*	Text ID	Num. of tokens	Num. of unique lemmas	Num. of sentenc es	Avg. sent. length *tokens*	Avg. sent. length *words*
TT1	3309	419	149	22.21	18.56	AI7	16,476	1,197	629	26.20	21.66
TT2	6042	552	220	27.46	22.70	AI8	11,629	1,081	433	26.87	22.85
TT3	3663	398	170	21.55	18.24	AI9	11,289	879	489	23.10	18.70
TT4	7869	714	357	22.04	17.35	AI10	16,726	1,225	645	25.93	20.76
TT5	5751	481	265	21.70	17.37	AI11	13,490	1,070	522	25.85	21.08
TT6	8849	665	389	22.75	18.16	AI12	11,452	910	408	28.08	21.96
AVG	**5,913.8**	**538.1**	**258.3**	**22.95**	**18.73**	**AVG**	**13,510**	**1,060.3**	**521**	**26.01**	**21.17**

In the present study we have compared the topic-related text pairs based on their vocabularies. In the process of creating the vocabularies, the texts were first segmented to exclude title, authors, affiliations, acknowledgement and reference list. The next step was to break down the text into tokens and filter out:

- stop words,
- numbers,
- punctuation marks and special characters,
- named entities, and
- incorrectly spelled words.

The remaining tokens were considered as words for which the occurrences were counted. Next, the words were lemmatized using part-of-speech (POS) tagging to determine the correct grammatical role of each word. Finally, the frequencies of words that share the same lemma were aggregated to produce a frequency distribution of base word forms.

3.2 Research Objectives

Once the vocabulary files containing word-frequency pairs had been created for each text in the corpus, the first research objective was to *compare lexical overlap and distinctiveness between the topic-related text pairs*. For this analysis, word frequencies in the vocabularies were first normalized to occurrences per 1,000 words to allow comparison. To present the results, the following computations were executed pairwise.

Lexical comparison: First, the total number of words were computed in each vocabulary. Then, the common words shared between the two vocabularies were identified, and finally, the number of words that are unique to each vocabulary (i.e., not present in the other) were determined. These results were visualized via bar charts and Venn diagrams.

As a next step, two similarity metrics were computed to assess lexical overlap. First, *Jaccard similarity,* J was calculated to measure how many unique words are shared between the two vocabularies. Then, *cosine similarity,* $\cos\theta$ is calculated to measure similarity of frequency distributions, i.e. how similar the word usage patterns are.

$$J = \frac{|A \cap B|}{|A \cup B|} \qquad \cos\theta = \frac{A \cdot B}{\|A\| \, \|B\|} \tag{1}$$

The result of both measures falls between 0 and 1. The closer the Jaccard similarity to 1 the higher the lexical overlap between the texts. When the Cosine similarity is close to 1 it indicates strong similarity in stylistic or lexical usage.

Statistical testing: To evaluate whether the distributions of word frequencies significantly differ between AI and Human/Translated texts, Mann-Whitney U tests were employed because the assumptions for t-test were not met. The H0 hypothesis was that there is no significant difference between the distributions of word frequencies in the topic-related text pairs. Three alternative hypotheses were tested:

- H1 two sided: significant difference exists between the distributions of word frequencies
- H1 greater: the distribution of word frequencies is greater in the AI text (AI > Human/Translated)
- H1 less: the distribution of word frequencies is less in the AI text (AI < Human/Translated).

Effect size: To quantify the magnitude of differences in word frequency distributions, Cohen's d was calculated as:

$$d = \frac{M_1 - M_2}{SD_{pooled}} \quad where \quad SD_{pooled} = \sqrt{\frac{(n_1-1)SD_1^2 + (n_2-1)SD_2^2}{n_1+n_2-2}} \tag{2}$$

In equation (2) M_1, and M_2 are means of word frequencies in group 1 and 2; SD_1, and SD_2 are the standard deviations for the two groups.

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{3}$$

where $\bar{x}$ is the mean of all x_i.

The result is interpreted as:

- $d > 0$: AI vocabulary tends to use words more frequently.
- $d < 0$: Human/Translated vocabulary has higher word frequency.
- $|d|$: Indicates the strength of the difference (small ~0.2, medium ~0.5, large ~0.8+).

The second research objective was to *find distinctive features of human-written and translated texts*. To capture lexical distinctiveness on a macro level, two diagrams were created. First, a bar chart of Cohen's d values shows the differences in the vocabularies across all text pairs. Next, the vocabularies across all AI and Human/Translated texts were aggregated to draw global Venn diagrams for visualizing overlap and unique word contributions in the two sub corpora (Human-AI and Translated-AI).

4 Results

For the experiments reported in this paper, two text corpora have been collected. The first corpus includes six human-authored texts (HT1–HT6) paired with six AI-generated versions (AI1–AI6) as listed in Table 3. In the case of human-written texts the average token count is 11,794.83 which indicates relatively long texts with substantial content volume. The average number of unique lemmas (799) reflects a rich and varied vocabulary typical of human writing. The average number of sentences (444.17) suggests well-developed textual structure and segmentation. The average sentence length in terms of words (21.42) demonstrates relatively long, syntactically complex sentences. This is 21% lower than the average sentence length in terms of all tokens, which means that human writers use a considerable amount of punctuation and references.

In this corpus, the AI-generated texts are shorter. Even though, the average number of unique lemmas (829.83) is surprisingly high, even slightly higher than in human texts, indicating broad lexical variety. The average sentence length is very similar in word count (21.88) to human-written texts, but in terms of tokens it is slightly shorter.

The second corpus includes six texts translated by researchers who learnt English as second or third language (TT1–TT6) and their AI-generated versions (AI7–AI12) as shown in Table 4. These texts are much shorter than texts in the Human-AI corpus, and the average number of unique lemmas (538.17) suggests limited lexical diversity in English language use. The average sentence length (18.73) in terms of words indicates shorter and simpler sentence constructions. The average number of tokens per sentence (22.95) is 15% lower than in the case of texts written by native English speakers.

In this corpus, AI texts are longer as AI was prompted to write around 10,000 words about each topic. The average number of unique lemmas (1,060.33) reflects very rich vocabulary, surpassing even human-written texts in HT group. The average sentence length is longer in both terms than in translated texts, which is closer to the complexity observed in the HT group.

To analyze lexical overlap, we have calculated similarity measures for the topic-related texts' vocabularies. Then, distinctiveness is tested through hypothesis tests. The results of the computations are displayed in Table 5.

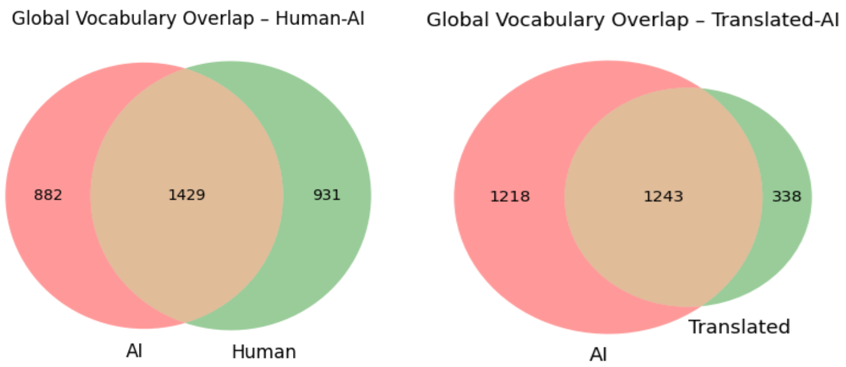
Human-AI corpus: For the first corpus, Jaccard similarity ranges from 0.26 to 0.35, with an average around 0.29.

This indicates a moderate level of lexical overlap between AI and human-written texts (see Fig. 1a). Cosine similarity values vary more widely (0.50 to 0.76), averaging around 0.65. This suggests that, although the shared vocabulary is moderate, AI and human texts reflect stylistic convergence in some cases.

Table 5. Results of lexical analysis (V1 vocabulary of first text, V2 vocabulary of second text).

Text pair	Total words in V1	Total words in V2	Unique words in V1	Unique words in V2	Common words	Jaccard sim.	Cosine sim.	U test greater	U test less	Cohen’s *d*
AI1-HT1	694	708	403	417	291	0.26	0.53	<0.01	1.0	0.01
AI2-HT2	1,081	782	602	303	479	0.35	0.76	1.0	<0.01	-0.16
AI3-HT3	900	882	516	498	384	0.27	0.50	<0.01	1.0	-0.01
AI4-HT4	832	955	434	557	398	0.29	0.74	<0.01	1.0	0.06
AI5-HT5	778	778	441	441	337	0.28	0.71	<0.01	1.0	<-0.01
AI6-HT6	694	689	409	404	285	0.26	0.53	<0.01	1.0	-0.01
AVG	**829.8**	**799**	**467.5**	**436.7**	**362.3**	**0.29**	**0.63**	-	-	-
AI7-TT1	910	665	505	260	405	0.35	0.83	1.0	<0.01	-0.14
AI8-TT2	1,225	714	779	268	446	0.29	0.62	1.0	<0.01	-0.27
AI9-TT3	1,070	481	755	166	315	0.25	0.68	1.0	<0.01	-0.35
AI10-TT4	879	398	613	132	266	0.26	0.80	1.0	<0.01	-0.42
AI11-TT5	1,081	552	719	190	362	0.28	0.75	1.0	<0.01	-0.38

AI12-TT6	1,197	419	891	113	306	0.23	0.68	1.0	<0.01	-0.56
AVG	**1,060.3**	**538.2**	**710.3**	**188.2**	**350**	**0.28**	**0.73**	**-**	**-**	**-**



a) b)

Fig. 6. a) Lexical overlap between human-authored vs AI texts; b) between translated vs AI texts.

The two-sided Mann-Whitney U test shows extremely low p-values which means that the distributions of word frequencies are significantly different. The one-sided tests clarify that word frequency distribution in AI-generated texts is higher which means that AI tends to repeat words more frequently and uses a narrower active vocabulary compared to human-written texts. Although the statistical tests detect significant differences, the practical size of the difference is minimal since *Cohen's d values* are small (see Fig. 2).

Translated-AI corpus: In the case of the second corpus, Jaccard similarity ranges from 0.22 to 0.31, slightly lower on average than for the Human-AI corpus. This indicates that translated texts and their AI counterparts have less vocabulary in common than the human-authored texts and their AI versions (see Fig.1b.). This may be because non-native English users operate with narrower vocabulary. Cosine similarity ranges from 0.56 to 0.69 which shows moderate similarity in word usage patterns.

For this corpus again, the two-sided Mann-Whitney U test produces low p-values which confirms significant differences in word frequency distributions. Now, the one-sided tests show an opposite trend in word frequency distributions.

The results suggest that the vocabulary is narrower and contains more repetitions in the case of translated texts. This statement is supported by Cohen's d values. These reflect small effect size in the negative direction. This means that AI texts tend to have somewhat broader vocabulary, but the differences in word repetition are minor.

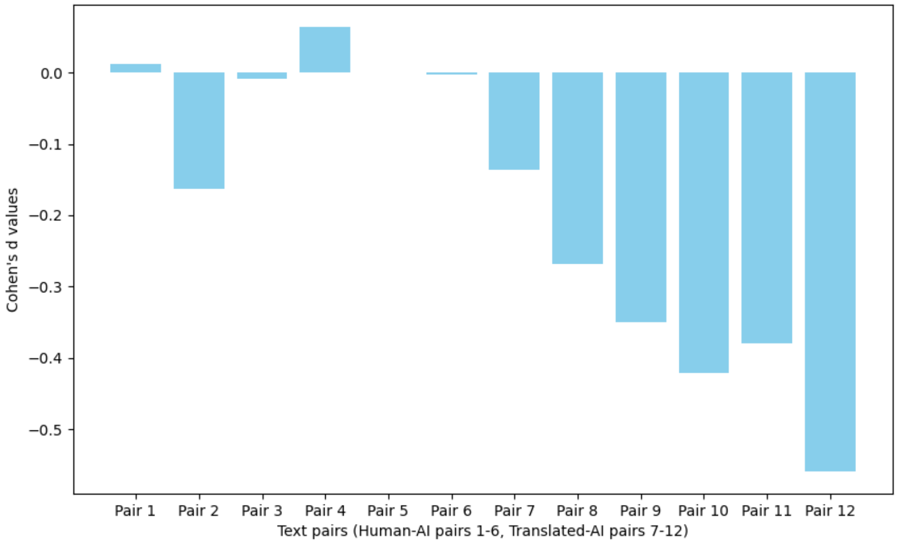


Fig. 2. Effect sizes of the differences in word frequency distributions.

Summarizing the results, AI-generated texts differ significantly from human and translated texts in terms of word frequency distributions in both corpora. However, the effect sizes are small, indicating that AI, particularly ChatGPT 4, is capable of closely mimicking human-like word usage patterns. Lexical overlap is notably lower in the Translated-AI corpus, which may suggest that non-native English writers exhibit more limited lexical diversity, a characteristic that is not substantially altered through translation.

5 Conclusion

This study explored lexical similarities and differences between AI-generated texts and their human-written or translated counterparts, focusing on vocabulary diversity and word frequency distribution. While the original working hypothesis anticipated more substantial stylistic and lexical differences in the Human-AI corpus, the findings reveal the opposite: greater lexical divergence was observed in the Translated-AI corpus. This outcome suggests that AI models, particularly ChatGPT-4, are highly effective in mimicking human-like word usage patterns. On the other hand, texts authored by non-native English speakers exhibited more limited vocabulary diversity and shorter sentence structures, possibly due to constraints in second-language proficiency. This linguistic simplicity is not significantly altered by translation, which preserves many of the original lexical limitations.

If we reverse this observation, it may provide a useful approach for detecting unethical or excessive reliance on AI. When a manuscript authored by non-native speakers displays unusually rich vocabulary, stylistic variety, or the use of less common word meanings, it may indicate AI assistance that goes beyond acceptable levels of support. This perspective could be valuable for academic integrity checks and authorship verification.

Future research will extend this work by analyzing the corpora at the sentence level, enabling a deeper understanding of syntactic structures, stylistic patterns, and possible markers of AI authorship.

References

1. Elkhatat, A.M., Elsaid, K. & Almeer, S. “Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text”. *Int J Educ Integr* 19, 17 (2023). https://doi.org/10.1007/s40979-023-00140-5
2. Islam, Niful & Sutradhar, Debopom & Noor, Humaira & Raya, Jarin & Maisha, Tabassum & Farid, Dewan. “Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning”. (2023). https://doi.org/10.48550/arXiv.2306.01761.
3. Hakam HT, Prill R, Korte L, Lovreković B, Ostojić M, Ramadanov N, Muehlensiepen F. “Human-Written vs AI-Generated Texts in Orthopedic Academic Literature: Comparative Qualitative Analysis”. *JMIR Form Res*. vol. 16; 8:e52164. (2024). https://doi.org/10.2196/52164.
4. Yildiz Durak, H., Eğin, F. and Onan, A. “A Comparison of Human-Written Versus AI-Generated Text in Discussions at Educational Settings: Investigating Features for ChatGPT”, Gemini and BingAI. *Eur J Educ*, vol. 60: e70014. (2025). https://doi.org/10.1111/ejed.70014
5. Schaaff, K., Schlippe, T. & Mindner, L. “Classification of human- and AI-generated texts for different languages and domains”. *Int J Speech Technol* vol. 27, pp. 935–956 (2024). https://doi.org/10.1007/s10772-024-10143-3
6. B. D. Fulcher and N. S. Jones, "Highly Comparative Feature-Based Time-Series Classification," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3026-3037, (2014). https://doi.org/10.1109/TKDE.2014.2316504
7. S. Poslad, S. E. Middleton, F. Chaves, R. Tao, O. Necmioglu and U. Bügel, “A Semantic IoT Early Warning System for Natural Environment Crisis Management”, in *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 2, pp. 246-257, (2015). https://doi.org/10.1109/TETC.2015.2432742
8. B. Harrison, S. G. Ware, M. W. Fendt and D. L. Roberts, “A Survey and Analysis of Techniques for Player Behavior Prediction in Massively Multiplayer Online Role-Playing Games”, *in IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 2, pp. 260-274, (2015). https://doi.org/10.1109/TETC.2014.2360463

9. W. Blewitt, M. Brook, C. Sharp, G. Ushaw and G. Morgan, “Toward Consistency of State in MMOGs Through Semantically Aware Contention Management”, in *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 2, pp. 275-288, (2015). https://doi.org/10.1109/TETC.2014.2331682
10. C. Demmans Epp and S. Bull, “Uncertainty Representation in Visualizations of Learning Analytics for Learners: Current Approaches and Opportunities” in *IEEE Transactions on Learning Technologies*, vol. 8, no. 03, pp. 242-260, (2015). https://doi.org/10.1109/TLT.2015.2411604
11. A. Bagnall, J. Lines, J. Hills and A. Bostrom, “Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles”, in *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522-2535, (2015). https://doi.org/10.1109/TKDE.2015.2416723
12. J. Alshboul, H.A.A. Ghanim, and E. Baksa-Varga, “Semantic Modeling for Learning Materials in E-Tutor Systems”, *Journal of Software Engineering and Intelligent Systems* (2518-8739): 6(2) pp. 1-5 (2021).
13. J. Alshboul and E. Baksa-Varga, “A Review of Automatic Question Generation in Teaching Programming” *International Journal of Advanced Computer Science and Applications* (IJACSA), 13(10), (2022). https://doi.org/10.14569/IJACSA.2022.0131006
14. J. Alshboul and E. Baksa-Varga, “A Hybrid Approach for Automatic Question Generation from Program Codes” *International Journal of Advanced Computer Science and Applications* (IJACSA), vol. 15(1), (2024). https://doi.org/10.14569/IJACSA.2024.0150102

8. A model for Albanian Fake News Detection Using Transformer-based Techniques

Elton Tata[1],Jaumin Ajdari[2] and Nuhi Besimi[3]

[1,2,3] Faculty of Contemporary Sciences and Technologies, South East European University, Tetovo, Republic of North Macedonia
j.ajdari@seeu.edu.mk, n.besimi@seeu.edu.mk

Abstract. Transformer-based architecture is designed to efficiently process sequential data through self-attention mechanisms, enabling advanced natural language understanding and text generation tasks. They have demonstrated exceptional performance in fake news detection by analyzing textual data with deep contextual understanding. The self-attention mechanism allows these models to capture intricate relationships between words, identify subtle patterns, and accurately classify news content as real or fake. This paper presents a deep learning Transformer-based model that combines LSTM and BERT architectures for Albanian fake news detection. The model is composed of an LSTM layer to capture sequential dependencies and a BERT module to extract contextual representations, enabling a more accurate classification of news content as real or fake. To train and evaluate the model, is used a dataset in the Albanian language. Precision, Recall, and F1-Score metrics were used to evaluate the model. It achieves an accuracy of 0.89, precision 0.9, recall0.84 and f1-score 0.87.

Keywords: Fake News, LSTM, BERT, Deep Learning, Transformers.

Introduction

The rapid expansion of digital media has dramatically reshaped the distribution of information, but it has also facilitated the spread of misleading and deceptive content – commonly referred to as “fake news”. Fake news not only misleads the public, but also undermines trust in legitimate sources of information and can have serious social consequences. Although social media and online platforms are a powerful source of knowledge, they have a detrimental impact on society. Due to the growing trend of individuals seeking and reading the latest news through social media. As a result, social media has become an extremely useful tool for journalists. This personalization of news development and sharing has two advantages: convenience and relevance. However, it also increases the risk of misinformation spreading throughout society in the form of propaganda, fake news and conspiracy theories. People from all over the world use these sites to get news about politics and celebrities, often without verifying the accuracy of the material. Fake news, which is material that is intentionally created and is clearly untrue, is often seen as a threat to the stability of democratic governments. In addition, to read articles about current saturation and breaking news. This allows us to have a device that is connected to an internet connection to communicate with each other and also to share information about current news or other common news. It influences the public in the crucial area of electronics, economic conditions and public opinion, gaining public trust in political institutions. Although most research on automated fake news detection has been conducted in high-resource languages such as English, low-resource languages such as Albanian remain largely unexplored. There is an urgent need for robust detection systems that can perform well even in these low-resource linguistic environments [1]. Traditional fake news detection methods typically rely on hand-designed features, such as sentiment analysis, stylometric signatures, or metadata (e.g., user engagement) [2]. However, these methods often fail to capture complex patterns or long-range dependencies inherent in text. In recent years, transformer-based models such as BERT [3] have revolutionized natural language processing (NLP) tasks by using self-attention mechanisms to capture contextualized relationships between words, thereby increasing understanding of local and global linguistic structures. Transformer-based architectures have demonstrated remarkable success in various text classification tasks, including fake news detection [4, 5]. However, while transformer models excel at modeling deep contextual relationships, they often struggle with sequence-level dependencies that are important for tasks like fake news detection, where word order and discourse structure play a crucial role [6]. Recent studies have addressed this gap by combining transformers with sequence models, such as long-term short-term memory (LSTM) networks, to better capture sequential dependencies [7]. LSTM networks are particularly effective at modeling the temporal or sequential nature of text, making them an ideal complement to transformer-based models. The integration of these two approaches, transformers for contextual embedding and LSTMs for sequential dependencies, has led to improved performance in various NLP tasks, including sentiment analysis and fake news detection [8, 9].

In this work, we propose a hybrid deep learning architecture that combines an LSTM layer with a pretrained BERT encoder to detect fake news in Albanian. The LSTM component captures sequential dependencies and discourse-level patterns, while the BERT module provides rich contextual representations at the token level. We train and evaluate our model on a curated dataset of Albanian news articles labeled as true or false. Using Precision, Recall, and F1-Score as evaluation metrics, we compare the hybrid model with both traditional baselines and state-of-the-art transformer approaches.

The primary objective of this study is to accurately detect and categorize fake news by using the effectiveness of deep learning model. The goal of this research is to increase the accuracy, effectiveness, and reliability of the fake news article. The ultimate objective is to provide an effective resource that can help people, groups, and networks reduce the spread of fake information.

The main objective of this research is to provide adaptable and efficient ways to tackle the widespread problem of fake news in the digital era, making a substantial contribution to the field of fake news identification.The following sections comprise the rest of this document: Section II provides a literature survey of the earlier researches and related to this work. Section III presents the research methodology for the proposed study. In section IV, are presented the experimental results. Section V presents discussion about results. Section VI offers a conclusion to the recommended study.

2 Related Work

Fake news detection has become a critical issue in the digital age, with research evolving from traditional machine learning techniques to advanced deep learning and transformer-based models. This section reviews important studies addressing fake news detection, describing their methodologies, findings, and limitations, while highlighting emerging trends and future research opportunities. We first review some of the classical machine learning approaches. Early efforts in fake news detection relied primarily on classical machine learning classifiers such as Naive Bayes, Random Forest (RF), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN). Studies such as [10, 11] demonstrated that ML models could effectively classify fake news with over 90% accuracy when integrated with appropriate preprocessing, tokenization, and feature extraction. [12] investigated the effectiveness of traditional machine learning algorithms in fake news detection by evaluating Naive Bayes, Passive Aggressive Classifier, and Deep Neural Networks on eight standard datasets. Their results showed that, with appropriate preprocessing and feature extraction techniques such as TF-IDF and Count Vectorizer, classical models can achieve high levels of accuracy (up to 90%) in classifying fake news. [13] advanced this model by introducing a multilingual classification model using DC Distance, which performed very well, outperforming traditional approaches such as Bag of Words (BoW) and Word2Vec in multilingual and multi-platform environments. However, these methods were limited to text analysis and lacked capabilities for multimodal data handling, social diffusion modeling, or deeper contextual understanding.

Several works based on deep learning and hybrid models are reviewed in this section. The move to deep learning (DL) marked a significant improvement in fake news detection capabilities. [14] proposed an ensemble model combining Bi-LSTM, GRU, and dense layers, achieving high accuracy on the LIAR dataset. [15] introduced WELFake, integrating word inputs and linguistic features, outperforming BERT in performance on their large and personalized datasets. These studies demonstrated the effectiveness of ensemble and hybrid approaches in capturing combined textual patterns. Advanced transformer-based models such as FakeBERT [16] and GBERT [17] achieved high-level accuracy above 95% by exploiting contextual embedding mechanisms within the BERT and GPT architectures. [18] addressed the input length limitations of BERT with their MWPBERT framework, using dual BERT encoders and a MaxWorth algorithm. One of the BERT networks encodes the news headline and the other encodes the news body. Since the news length is a long text and cannot be fully loaded into BERT, the MaxWorth algorithm was used, selecting the part of the news text that is most valuable for news verification, while [19] improved entity awareness and relational meaning using a joint BERT, Named Entity Recognition (NER) and Entity Reclassification (NER) models. Despite their successes, these models remained largely text-focused, neglecting visual imagery, social and relational sensitivity. They also faced challenges regarding scalability, interpretability, and computational efficiency, making it difficult to deploy in the real world on high-traffic social media platforms.

Other reviewed works include the integration of multimodal and propagation-based techniques.

Recognizing the multimodal nature of digital disinformation, recent studies have explored the integration of different data modalities. [20] introduced MCNN, a Multimodal Consistency Neural Network that combines text, image, and intrusion detection modules, setting new standards for multimodal fake news detection. However, the approach introduced significant computational complexity and limited interpretability, presenting practical deployment challenges. Graph-based models such as FAKEDETECTOR [21] and newer Graph Convolutional Network (GCN) approaches modeled the relational dynamics of news distribution, capturing how news articles, their creators, and related topics interact within social networks. These models improved fake news detection by analyzing the networks of article creators, but often required significant computational resources and lacked transparency in decision-making processes.

Other studies include surveys and behavioral studies. Comprehensive surveys such as [22] reviewed around 148 deep learning studies, confirming the superior performance of DL models over traditional ML and recommending the future integration of multimodal, graph-based, and real-time scalable models. Behavioral studies by [23, 24] shift the focus of fake news detection from simple technical approaches to the behavioral and social dimensions of misinformation. The former explored the psychological motivations behind sharing COVID-19 fake news on social media in Nigeria. Complementing this individual-level analysis, [24] examined the broader societal impacts of fake news. Their study, based on a fuzzy set Qualitative Comparative Analysis (fsQCA), showed that misinformation disrupts social norms, erodes public trust, and contributes to social polarization. Furthermore, their findings underscored significant public concern about the ineffectiveness of social media platforms in curbing the spread of fake news. Together, these studies highlight that effectively mitigating fake news requires more than algorithmic solutions—it requires an understanding of human behavior and the broader social environment in which fake news spreads. [25] proposed a theory-driven approach to early fake news detection that relies solely on textual content, aiming to identify misinformation before it spreads widely on social media. [26] assessed the prevalence of fake news at the level of the media consumption ecosystem, finding that fake news accounted for only 0.15% of daily media exposure, but noted its tremendous impact on public perceptions through framing effects and media avoidance behavior. Their findings suggested that the real impact of misinformation may lie in its selective framing and distortion of mainstream narratives rather than in its full scope.

The current reviewed works once again underscore the progress from classical ML to sophisticated DL and hybrid models. Persistent gaps include limited real-time scalability, poor model interpretability, underutilization of data sources and multilingualism, and challenges in effectively integrating visual, relational, and propagation-based features. Addressing these gaps remains essential for developing robust and reliable fake news detection systems capable of adapting to the rapidly evolving digital media ecosystem.

The table below shows a summary of these existing works included in this study in terms of the algorithm used, data sets, and accuracy results. A similar table on the techniques used, datasets and features used is shown in our previous work given by [27].

Table 1. Summary of related work.

Nr.	Name	Architecture	Dataset	Comments (Accuracy)
1	[16]	FakeBERT (BERT + SVM + NSGA-II)	COVID-19 Dataset	Acc: +5.2% over baseline
2	[17]	GBERT (GPT + BERT)	Two real-world datasets	Acc: 95.3%, F1: 96.23%
3	[18]	MWPBERT (Dual BERT + MaxWorth)	Long-text corpus	Acc: 85.4%
4	[19]	BERT + NER + RFC	2 entity-rich datasets	Acc: 84%
5	[21]	FAKEDETECTOR (Graph + Content)	Multi-entity network dataset	Acc: 63%
6	[15]	WELFake (Embedding + Linguistic)	WELFake (72k articles)	Acc: 96.73%
7	[12]	Naive Bayes, Passive Aggressive, DNN	8 mixed datasets	Acc: ~ 70% - 99% for diferent datasets
8	[14]	Bi-LSTM + GRU + Dense	LIAR Dataset	Acc: 89.8%, F1: 91.6%
9	[20]	MCNN (Text + Image + Tampering)	4 multimodal datasets	Acc: ~ 78.4% - 96.3% for diferent datasets

10	[10]	SVM, RF, LR, Ensemble	4 diverse datasets	Acc: >90%
11	[25]	Psycholinguistic Features	Theory-backed dataset	Acc: >80%
12	[26]	Media Exposure Analysis	Comscore, Nielsen	Fake news = 0.15% of Americans' daily media diet
13	[24]	fsQCA (Societal Framework)	Survey research	Highlights societal norm disruption
14	[11]	RF, SVM, DT, KNN	Preprocessed news set	RF & SVM highest above 95%
15	[22]	DL Survey (148 papers)	Review of DL papers	DL techniques are more accurate in detecting fake news than ML

3 Methodology

The proposed method is designed to develop an effective fake news detection system using a hybrid deep learning architecture that integrates both Long Short-Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). This model is structured into several key phases:

1. Data collection
2. Language Preprocessing
3. Extracting preview features
4. Adaptive training by implementing LSTM and BERT
5. Classification architecture
6. Performance Metrics

The below figure shows the diagram of the model based on the output process.

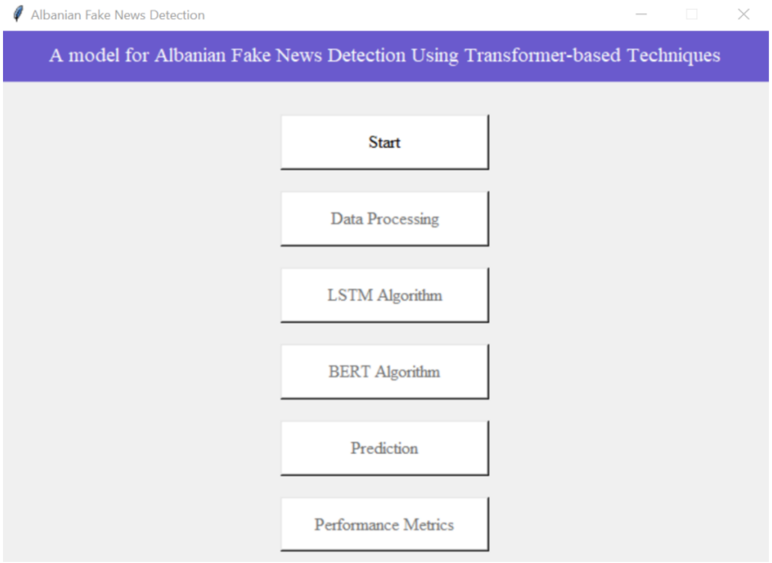

Albanian Fake News Detection
A model for Albanian Fake News Detection Using Transformer-based Techniques
Start
Data Processing
LSTM Algorithm
BERT Algorithm
Prediction
Performance Metrics

Fig.1 Diagram of the model.

3.1 Data collection

Our proposed method uses Albanian datasets for fake news detection. This data contains only text. In it, a public dataset of labeled real and fake news is presented, performing an extensive analysis of deep learning methods for analyzing fake news. The dataset used in this study stems from the work of [28], who initially collected approximately 24,700 news articles in the Albanian language. After pre-processing and filtering, the final dataset was curated to include 4,000 articles - evenly divided between 2,000 real news and 2,000 fake news. The news items that make up this dataset include various categories, politics, entertainment, etc. The articles cover a six-month period during 2020 when the study was conducted. It is important to note that the dataset provides thematic matching between real and fake news using text similarity techniques such as TF-IDF and Latent Semantic Indexing (LSI), allowing for a fair comparative learning structure. Fake news cases were identified through a manual annotation process, where three reviewers assessed 3,900 candidate articles and selected 2,000 deemed fraudulent. These were sourced from questionable sources, including social media and platforms such as no-fakenews.com. In contrast, real news stories were collected from trusted and reputable media outlets and were not subject to manual validation. This hybrid approach, partly of manual annotation, was adopted due to resource constraints, but was carefully managed to ensure the integrity of the dataset. The fake news content of the dataset is enhanced through the use of an intelligence augmentation technique. Such a large set of data encourages the improvement of this problem, using the appropriate methods will improve the prediction algorithms. The collected data is compatible with binary classification. Either the tag is true or it isn't. The news is collected from different sources making this group of data special. This dataset is openly accessible to all research communities. Albanian data is used in our model to extract and validate the text.

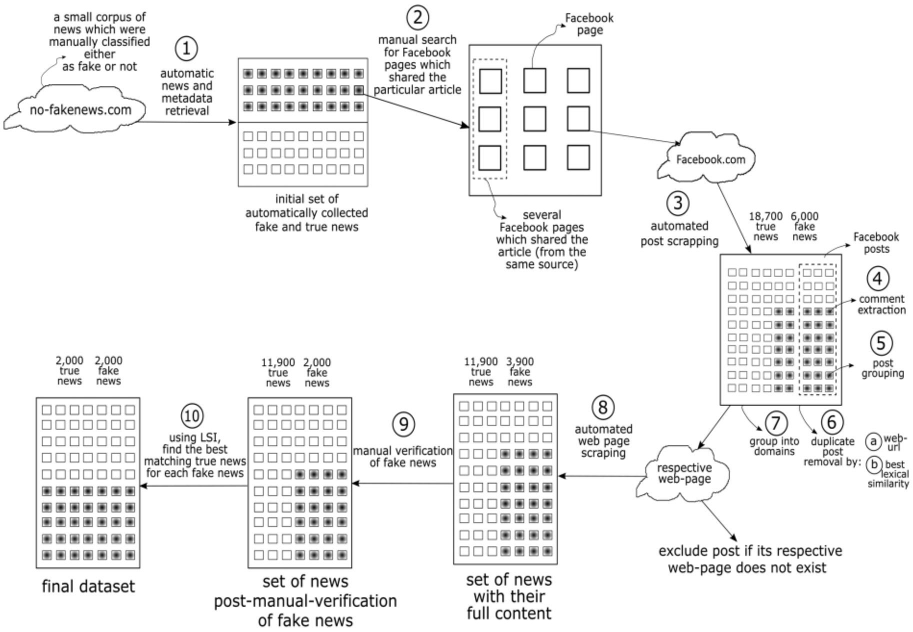

a small corpus of news which were manually classified either as fake or not
no-fakenews.com
1
automatic news and metadata retrieval
initial set of automatically collected fake and true news
2
manual search for Facebook pages which shared the particular article
Facebook page
several Facebook pages which shared the article (from the same source)
Facebook.com
3
automated post scrapping
18,700 true news
6,000 fake news
Facebook posts
4
comment extraction
5
post grouping
7
group into domains
6
duplicate post removal by:
a web-url
b best lexical similarity
respective web-page
exclude post if its respective web-page does not exist
8
automated web page scraping
11,900 true news
3,900 fake news
set of news with their full content
9
manual verification of fake news
11,900 true news
2,000 fake news
set of news post-manual-verification of fake news
10
using LSI, find the best matching true news for each fake news
2,000 true news
2,000 fake news
final dataset

Fig.2 Process of building the Alb-Fake-News-Corpus [28].

3.2 Data Preprocessing

Preprocessing is the process of cleaning, transforming, and organizing raw data into a suitable format before feeding it into a deep learning (DL) model. It involves a set of operations aimed at enhancing the quality and structure of the data to ensure accurate and reliable predictions.

In the context of text-based tasks, preprocessing includes operations like removing irrelevant columns, handling missing data, and converting data types. In this paper, this process starts by reading the dataset from Fake_news.csv using the Pandas library and extracting the original data for inspection. Unnecessary columns such as (facebook_url and web_page_url) are then removed, likely because these URLs do not contribute to the detection of fake news. Removes rows with missing values to ensure the dataset is clean. Converting Date Column and Label Data Type to a proper datetime format. Changes the fake_news column to a boolean type (i.e., True for fake news and False for real news).

This step ensures that the dataset is properly formatted and cleaned, which is essential for building an accurate fake news detection model. Preprocessing is a crucial step in machine learning pipelines because it ensures that the input data is clean, consistent, and formatted correctly. Without proper preprocessing, models can produce misleading results and perform poorly.

Feature extraction is the process of transforming raw data into a set of informative and representative features that can improve the performance of machine learning models. For text-based tasks like fake news detection, feature extraction converts unstructured text into structured numerical data suitable for machine learning algorithms. For this we use the Albanian fake news dataset features such as 'title', 'content', 'publication_datetime', 'fake_news'.

Is used countVectorizer to convert raw text data (news content) into numerical features suitable for the classifier. CountVetorizer decomposes each article into individual words. In this way, it builds a dictionary that contains all the unique words found in the entire training data set. For each article, it creates a row vector with the count of each word from the dictionary. Thus a matrix is formed where each row corresponds to a document and each column corresponds to a word in the dictionary.

3.3 Classification architecture by implementing LSTM and BERT

First, we collect, load, and preprocess fake news data in Albania. Then, we implement a deep learning model using LSTM (Long Short Term Memory) for fake news detection. The LSTM layer helps to capture patterns in the sequence of features to predict whether the news is fake or real. It outputs the test loss and accuracy, which indicate how well the model generalizes to unseen data. It reads a preprocessed dataset from a CSV file, preprocesses it by encoding labels, and splitting the data into training and testing sets. It randomly splits the dataset into 80% training data and 20% testing data. The LSTM architecture consists of a single layer with 100 memory units, followed by a dropout layer to reduce overfitting, and a dense output layer with a sigmoid activation function for binary classification. It builds the model with binary cross entropy loss and the Adam optimizer. Finally, it trains the model for 500 epochs, evaluates its performance on the test set, and prints the test loss and accuracy. Then, a second pipeline includes the BERT (bert-base-multilingual-cased) tokenizer to preprocess the text content of the news articles. Instead of processing a full BERT model, the tokenized data is vectorized using a Count Vectorizer to produce token count matrices. The classifier is trained using partial fits over multiple epochs and predictions are made on the test data. The model's performance is evaluated on the test data and the results are reported using accuracy and a detailed classification report.

4 Result

The experimental examination of the proposed architecture for Identifying Fake news is shown in this section. This experimental result proves that the proposed method enhances the classification accuracy of fake news detection. Evaluation metrics included Accuracy, Precision, Recall, and F1-Score.

These results validate the model's capability to generalize across various textual patterns and affirm the effectiveness of combining contextual and sequential modeling techniques.

The proposed hybrid model combining LSTM and BERT was tested on the Albanian fake news dataset. The evaluation shows in the table the following:

Table 2. Result of the model.

Metric	**Score %**
Accuracy	88.6
Precision	90
Recall	84
F1-Score	87

The results demonstrate the model's ability to effectively distinguish fake from real news, confirming that integrating sequential and contextual learning layers improves overall detection performance. The model's robustness is further supported by its consistent performance across different metrics.

These results achieved by the proposed model are also shown in the graphs below.

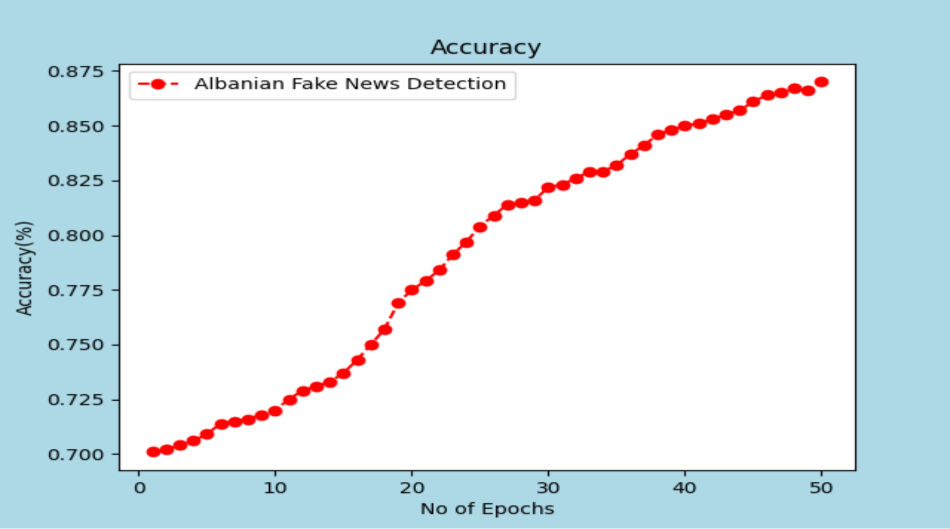


Fig. 3 Accuracy percentage.

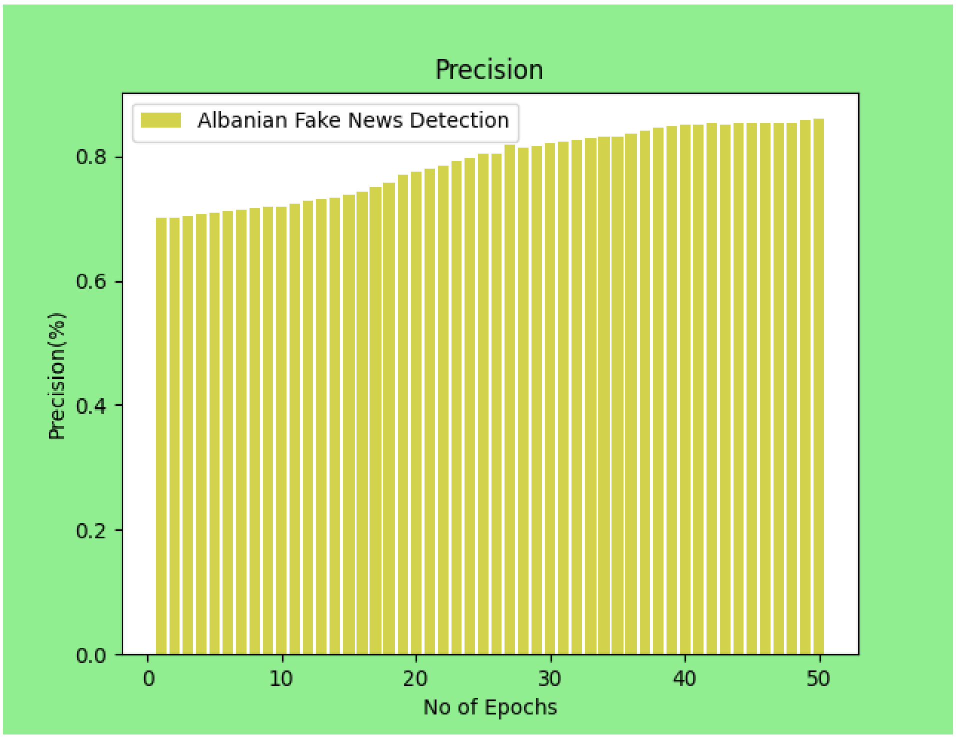


Fig. 4 Precision percentage.

5 Discussion

The evaluation results of the proposed model highlight its practical effectiveness in detecting fake news written in Albanian. With an accuracy of 88.6% and an F1 score of 0.87, the model demonstrates a balanced ability to correctly identify fake and real news. The hybrid uses of sequential processing via LSTM and contextual understanding via BERT significantly contributes to this result. The table below provides a comparison of the results of this work with those of the works mentioned in related work.

Table 3. Comparison of the results.

Study	Method	Accuracy / Key Metric
This Work	LSTM + BERT	88.6% (F1: 0.87)
[17]	GBERT (GPT + BERT)	95.3%

[15]	Word Embedding + Linguistic (WELFake)	96.73%
[14]	Bi-LSTM + GRU + Dense	89.8%, (F1: 91.4%)
[10]	Ensemble ML (SVM, RF, etc.)	>90%
[21]	FAKEDETECTOR (Graph + Content)	63%
[25]	Psycholinguistic Features	>80%
[18]	MWPBERT (Dual BERT + MaxWorth)	85.4%
[19]	BERT + NER + RFC	84%

Compared to the approaches mentioned in the Related Work section, the implemented method offers several strengths. The proposed model benefits from automatic feature learning using deep learning. While some transformer-based models such as FakeBERT or GBERT achieve higher accuracy, they are usually resource-intensive and less suitable for low-resource languages such as Albanian. Our model strikes a balance between performance and computational feasibility, making it a practical choice. However, in contrast to truly multimodal or graph-based models (e.g., MCNN or FAKEDETECTOR), our system does not yet incorporate visual content or relational propagation features. One of the main contributions of this paper is the implementation and validation of a hybrid LSTM-BERT-based method in a low-resource language. Another limitation is the lack of integration between the LSTM and BERT components in a unified architecture, which could further improve performance by learning sequential and contextual models from start to end.

6 Conclusion

This research presents a hybrid deep learning model that leverages both LSTM and BERT to detect fake news in the Albanian language. By combining sequential dependency modeling with contextual embeddings, the model achieves high classification accuracy and outperforms traditional text-only methods. The study also highlights key limitations in existing fake news detection approaches, including scalability, context sensitivity, and multimodal integration. The results underscore the model's potential in low-resource language settings and lay the groundwork for future enhancements incorporating real-time detection, graph-based relational modeling, and multimodal data. Ultimately, this work contributes to the growing field of trustworthy For this paper, several important research directions remain to be explored in the future. First, the full integration of an advanced architecture such as BERT for classification would enable major gains in the semantic understanding of news. Also, a unified architecture that combines LSTM and BERT in an interactive manner would be a significant improvement over the current structure. Also, the addition of modalities other than text, such as images, metadata and information from social networks would provide a more complete view for the classification of fake news. Finally, the inclusion of explainability mechanisms (Explainable AI) would increase the trust and transparency of the system, giving users the opportunity to understand why an article has been classified as fake.

References

1. Arkadipta De, Dibyanayan Bandyopadhyay, Baban Gain, and Asif Ekbal. 2021. A Transformer-Based Approach to Multilingual Fake News Detection in Low-Resource Languages. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 21, 1, Article 9 (January 2022), 20 pages. https://doi.org/10.1145/3472619
2. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2020). "Fake News Detection on Social Media: A Data Mining Perspective." *ACM Computing Surveys, 53*(5), 1-34.
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT.*
4. Alghamdi, J., Luo, S. & Lin, Y. A comprehensive survey on machine learning approaches for fake news detection. *Multimed Tools Appl* **83**, 51009–51067 (2024). https://doi.org/10.1007/s11042-023-17470-8.
5. Essa, E., Omar, K. & Alqahtani, A. Fake news detection based on a hybrid BERT and LightGBM models. *Complex Intell. Syst.* **9**, 6581–6592 (2023). https://doi.org/10.1007/s40747-023-01098-0
6. K. K. Buddi, L. Anamalamudi, S. S. Mutupuri, R. Madupuri, S. C C and S. Anamalamudi, "Hybrid Deep Learning Approach for Information Analysis and Fake News Detection," *2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN)*, Bangkok, Thailand, 2023, pp. 1-5, doi: 10.1109/CICN59264.2023.10402146.
7. F. Al-Quayed, D. Javed, N. Z. Jhanjhi, M. Humayun and T. S. Alnusairi, "A Hybrid Transformer-Based Model for Optimizing Fake News Detection," in *IEEE Access*, vol. 12, pp. 160822-160834, 2024, doi: 10.1109/ACCESS.2024.3476432
8. Arun kumar yadav , Suraj Kumar , Dipesh Kumar , et al. Fake News Detection using Hybrid Deep Learning Method. *TechRxiv.* May 05, 2022.
DOI: 10.36227/techrxiv.19689844.v1
9. Raghavendra, R., Niranjanamurthy, M. An Effective Hybrid Model for Fake News Detection in Social Media Using Deep Learning Approach. *SN COMPUT. SCI.* **5**, 346 (2024). https://doi.org/10.1007/s42979-024-02698-4
10. Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. O. (2020). Fake news detection using machine learning ensemble methods. *Complexity*, *2020*(1), 8885861.
11. Khanam, Z., Alwasel, B. N., Sirafi, H., & Rashid, M. (2021, March). Fake news detection using machine learning approaches. In *IOP conference series: materials science and engineering* (Vol. 1099, No. 1, p. 012040). IOP Publishing.

12. Mandical, R. R., Mamatha, N., Shivakumar, N., Monica, R., & Krishna, A. N. (2020, July). Identification of fake news using machine learning. In *2020 IEEE international conference on electronics, computing and communication technologies (CONECCT)* (pp. 1-6). IEEE.
13. Arruda Faustini, P.H., Covões, T.F., Fake news detection in multiple platforms and languages, *Expert Systems with Applications* (2020), doi: https://doi.org/10.1016/j.eswa.2020.113503
14. Aslam, N., Ullah Khan, I., Alotaibi, F. S., Aldaej, L. A., & Aldubaikil, A. K. (2021). Fake detect: A deep learning ensemble model for fake news detection. *complexity*, *2021*(1), 5557784.
15. Verma, P. K., Agrawal, P., Amorim, I., & Prodan, R. (2021). WELFake: Word embedding over linguistic features for fake news detection. *IEEE Transactions on Computational Social Systems*, *8*(4), 881-893.
16. Ali, Arshad, and Maryam Gulzar. "An Improved FakeBERT for Fake News Detection" Applied Computer Systems, vol. 28, no. 2, Riga Technical University, 2023, pp. 180-188. https://doi.org/10.2478/acss-2023-0018
17. Dhiman, P., Kaur, A., Gupta, D., Juneja, S., Nauman, A., & Muhammad, G. (2024). GBERT: A hybrid deep learning model based on GPT-BERT for fake news detection. *Heliyon*, *10*(16).
18. Farokhian, M., Rafe, V., & Veisi, H. (2024). Fake news detection using dual BERT deep neural networks. *Multimedia Tools and Applications*, *83*(15), 43831-43848.
19. Shishah, W. (2021). Fake news detection using BERT model with joint learning. *Arabian Journal for Science and Engineering*, *46*(9), 9115-9127.
20. Xue, J., Wang, Y., Tian, Y., Li, Y., Shi, L., & Wei, L. (2021). Detecting fake news by exploring the consistency of multimodal data. *Information Processing & Management*, *58*(5), 102610.
21. Zhang, J., Dong, B., & Philip, S. Y. (2020, April). Fakedetector: Effective fake news detection with deep diffusive neural network. In *2020 IEEE 36th international conference on data engineering (ICDE)* (pp. 1826-1829). IEEE.
22. Mridha, M. F., Keya, A. J., Hamid, M. A., Monowar, M. M., & Rahman, M. S. (2021). A comprehensive review on fake news detection with deep learning. *IEEE access*, *9*, 156151-156170.
23. Apuke, O. D., & Omar, B. (2021). Fake news and COVID-19: modelling the predictors of fake news sharing among social media users. *Telematics and informatics*, *56*, 101475.
24. Olan, F., Jayawickrama, U., Arakpogun, E. O., Suklan, J., & Liu, S. (2024). Fake news on social media: the impact on society. *Information Systems Frontiers*, *26*(2), 443-458.
25. Zhou, X., Jain, A., Phoha, V. V., & Zafarani, R. (2020). Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice*, *1*(2), 1-25.
26. Allen, J., Howland, B., Mobius, M., Rothschild, D., & Watts, D. J. (2020). Evaluating the fake news problem at the scale of the information ecosystem. *Science advances*, *6*(14), eaay3539.
27. E. Tata, J. Ajdari and N. Besimi, "Fake News Detection: A Comprehensive Survey," *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, Opatija, Croatia, 2023, pp. 309-314, doi: 10.23919/MIPRO57284.2023.10159859.
28. Ercan Canhasi, Rexhep Shijaku, and Erblin Berisha. 2022. Albanian Fake News Detection. *ACM Trans. AsianLow-Resour. Lang. Inf. Process.* 21, 5, Article 86 (November 2022), 24 pages. https://doi.org/10.1145/3487288

9. Evaluation of free and local LLMs compared to commercial LLMs for the Albanian language

Raphaël Couturier[1], Joseph Azar [2], Amarildo Rista[3], and Shkelqim Fortuzi [4]

[1,2] Université Marie et Louis Pasteur, CNRS, Institut FEMTO-ST, F-90000, Belfort, France
[3,4] Aleksandër Moisiu University of Durrës, Albania

raphael.couturier@univ-fcomte.fr

Abstract. Dialoguing with LLMs (Large Language Models) is nowadays a common activity for many people. While English is clearly the most used language, many other languages are supported by LLMs, and even less commonly used languages can now be used. This is the case with Albanian. Moreover, although many commercial LLMs are available, using them often requires sharing your own data. Therefore, an interesting alternative is to use free and local LLMs, which can be easier to use on a daily basis and that respects privacy. In this paper, we will evaluate Albanian-speaking LLMs on various common tasks that LLMs can perform, such as question answering, text summarization, Albanian style correction, etc.

Keywords: Large Language Model, Albanian language, LLM evaluation.

1 Introduction

Large Language Models (LLMs) are advanced machine learning models based on deep learning techniques, particularly the Transformer architecture [1-2]. LLMs are built from layers of neural networks, and the attention mechanism, which during training from a specific corpus generates the context of the sentence and not sequence by sequence, by fine-tuning the performance [3]. So, during the training process, these models predict the next word based on the context provided by the corpus. LLMs have shown exceptional potential to perform a wide range of Natural Language Processing (NLP) tasks, particularly in tasks such as translation, summarization, and question answering [4-5]. These models can produce context, coherent and contextually appropriate responses, translate text between languages, summarize content, answer questions, and assist with tasks such as creative writing and code generation. The effectiveness of LLMs stems from their massive scale that contains thousands of parameters allowing them to learn intricate patterns in language [6]. Although LLMs have demonstrated exceptional performance across a wide range of NLP tasks, most advancements have focused on high-resource languages. Extending these capabilities to low-resource languages remains a significant challenge. The lack of well-annotated corpora and linguistic tools for such language's limits research and development in this area. While extensive research has been conducted for high-resource languages like English, each language presents unique syntactic, morphological, and grammatical features, making it difficult to standardize a single model across all languages.

This paper evaluates the effectiveness of both commercial and open-source LLMs in processing the Albanian language. By including both conventional and semantic-based assessment methods, we will evaluate four language models—GPT-4, Gemma, DeepSeek, and Mistral [7-10]. A multilingual resource dataset designed to advance automatic summarization for low-resourced languages will be used to train the models.

The rest of the paper is structured as follows: Section 2 presents Literature Review; Section 3 shows How to evaluate the capacity of Albanian LLM. Section 4 describes experimental results. Section 5 presents the Discussion, and Section 5 lists the conclusions.

2 Literature Review

Recent advancements in natural language processing (NLP) have been largely driven by the development and scaling of Large Language Models (LLMs), which have demonstrated unprecedented performance across a variety of tasks, from machine translation and summarization to question answering and code generation. These models, primarily built on the Transformer architecture, leverage extensive self-supervised training on massive datasets, enabling them to capture nuanced linguistic patterns and achieve generalized performance across domains.

Naveed et al [1] have provided an survey that categorizes LLM research into key areas such as pre-training, fine-tuning, efficient inference, evaluation, and applications. Their work emphasizes the historical evolution of LLMs, architectural innovations, and the emergence of models with billions of parameters, including GPT-3, PaLM, and LLaMA, highlighting a shift towards instruction-tuned and open-source models. Similarly, Patil and Gudivada [11] delve into the evolution of language models from static embeddings to dynamic and context-aware architectures, underscoring the transition from task-specific to task-agnostic systems that dominate current research and applications. They also discuss various fine-tuning and in context learning techniques applied to downstream tasks. Moreover, it explores how large language models (LLMs) can achieve strong performance across diverse domains and datasets, provided they are trained on sufficiently large and varied data.

Efforts to assess LLMs in low-resource languages, such as Albanian, remain relatively limited. The existing literature suggests that most benchmarks and datasets cater to high-resource languages like English, creating a disparity in performance and applicability for underrepresented languages. Addressing this gap, the current study builds upon prior work by evaluating both commercial (e.g., GPT-4) and open-source models (e.g., Gemma, DeepSeek, Mistral) using multilingual and semantic evaluation metrics tailored for the Albanian language.

Emerging research has increasingly focused on evaluating and extending the capabilities of LLMs to low-resource languages. Cahyawijaya et al [12] present a aproach of Indonesian LLMs low resource language, encompassing both decoder-only and encoderdecoder architectures across a range of model sizes. They demonstrate that while LLMs show promise in few-shot in-context learning for such languages, performance is often limited by insufficient pretraining data and weak cross-lingual alignment. They introduce query alignment as a more effective alternative to label alignment in cross-lingual ICL, emphasizing semantic similarity between source and target exemplars.

On the architecture side, significant advancements are being made to improve long-context handling and efficient training. Huang et al [13] provide a taxonomy of methods aimed at overcoming transformer limitations in long-context scenarios, including memory optimization, efficient attention, and extrapolative positional embeddings. Du et al [14] propose Gstack, a depthwise model growth strategy that enables efficient scaling of LLMs with reduced computational overhead, exhibits remarkable acceleration in training, leading to decreased loss and improved overall performance.

Sun et al. [15] propose the Transformer-Squared model, which enables real-time self-adaptation of large language models (LLMs) to unseen tasks by selectively adjusting only the singular components of their weight matrices. This is achieved through a two-pass expert dispatch mechanism, allowing the model to outperform traditional parameter-efficient tuning methods.

The growing divide between commercial and open-source large language models (LLMs) has attracted significant attention. Choi and Chang [16] offer a comparative analysis of DeepSeek and ChatGPT, highlighting key trade-offs between the two approaches. Their findings show that while commercial models such as ChatGPT deliver superior out-of-the-box performance and ease of deployment, open-source alternatives like DeepSeek excel in transparency, customization, domain-specific adaptability, and long-term cost efficiency, especially in self-hosted environments. Notably, DeepSeek demonstrated strong performance in specialized tasks following fine-tuning, particularly within computational domains, whereas ChatGPT maintained an edge in general-purpose applications without additional training. This work informs the current study, which evaluates both commercial and open-source LLMs for the Albanian language, using a multilingual summarization dataset. By focusing on a low-resource language, the study contributes to closing the resource gap and provides empirical baselines that support further development and assessment of LLMs in underrepresented linguistic contexts.

3 How to evaluate the capacity of Albanian LLM

In order to fill the gaps in summarization, question answering, and translation, recent studies have investigated natural language processing tasks for the Albanian language. Language-dependent elements perform better in novel extractive summarization systems designed especially for Albanian [17, 20]. Additionally, Albanian question-answering systems have been introduced, showing promise for documents with a single domain [21]. For various NLP tasks, pre-training language models such as BERT on Albanian datasets have demonstrated promise [20]. Despite the language's complexity, named entity recognition techniques utilizing deep learning have been proposed for Albanian, with encouraging results [24]. Furthermore, Albanian educational reviews have been used to refine sentiment analysis models, yielding baseline results [23]. While investigating more general uses of large language models in text summarization and evaluation across multiple languages, these studies also advance NLP capabilities for the Albanian language [19, 25].

4 Experimental Results

4.1 Evaluation Dataset

The LR-Sum dataset, introduced by Chester Palen-Michel and Constantine Lignos from Brandeis University is used in this paper [26]. This dataset is a multilingual resource designed to advance automatic summarization for less-resourced languages. It includes summaries of 40 languages written by humans, many of which are not well-represented in studies on natural language processing. Content from the Multilingual Open Text corpus, which sources public domain newswire articles from Voice of America websites, was extracted and filtered to create the dataset. Integration into different NLP workflows is made easier by the dataset's availability on Hugging Face at https://huggingface.co/datasets/bltlab/lr-sum and on GitHub at https://github.com/bltlab/lr-sum.

In order to have a unified evaluation, LR-Sum summarizing dataset was used to produce evaluation data for all three tasks:

- Translation Dataset: We used Azure AI Translation Service to translate the English source texts and their summaries from LR-Sum into Albanian, making pairs of English and Albanian texts that are the same.
- Question Answering Dataset: We used GPT-4o to come up with questions that were related to the Albanian texts. Then, we used Azure Language Service's question answering feature to get the correct answers directly from the Albanian texts (since the service highlights answers from the text instead of making them up).
- Summarization Dataset: Directly used the original LR-Sum dataset.

4.2 Evaluation Methodology

By including both conventional and semantic-based assessment methods, our evaluation system sought to overcome the shortcomings of conventional lexical-based measures. On three separate natural language tasks—translation, summarization, and question answering—we evaluated four state-of-the-art language models—GPT-4, Gemma, DeepSeek, and Mistral. In this work, all of the models were tested in a zero-shot condition, which means no fine-tuning was done for a specific task. We utilized the same prompting tactics for all models:

- For translation: "Translate the following Albanian text into English: [text]"
- To summarize: "In three sentences, summarize the following Albanian text: [text]"
- For answering questions: "Use the context to answer the following question: Context: [context] Question [question]?

This zero-shot method is similar to how people use pre-trained models in the real world without any extra training.

4.3 Evaluation metrics

For each task, we employed a comprehensive set of metrics:

Translation:

- **BLEU:** Measures n-gram overlap between model outputs and reference translations
- **Semantic Similarity:** Embedding-based similarity that captures meaning beyond lexical overlap

Summarization:

- **Reference-based metrics:**
    - **ROUGE-1/2/L:** Measures unigram, bigram, and longest common subsequence overlap
    - **Semantic Similarity:** Captures conceptual alignment with reference summaries
    - **Information Coverage:** Entity-based measure of key information preservation
    - **BERTScore:** Contextual embedding-based similarity
- **Source-based metrics:**
    - **Semantic Relevance:** Measures conceptual alignment between summary and source
    - **Source Coverage:** Entity-based measure of information captured from source
    - **Conciseness:** Evaluates appropriate compression ratio
    - **BERTScore:** Contextual alignment between summary and source

Question Answering:

- **Semantic Similarity:** Measures meaning-level alignment with reference answers
- **Containment:** Evaluates what proportion of reference content is present
- **Information Coverage:** Entity-based measure of key information preservation

4.4 Implementation Details

Our approach used SentenceTransformers for semantic similarity calculation, SpaCy for entity extraction, and BERTScore for contextual similarity evaluation. Our assessment was particularly designed to handle long-form text; we used paragraph-level breakdown to process long source documents. The implementation is available on GitHub: https://github.com/josephazar/slm_language_tasks_benchmark.

4.5 Results and Analysis

Except DeepSeek, all models showed good translation ability with BLEU scores between 0.61-0.65, as indicated in Figure 1. This is due to the reasoning outputs generated by the model in addition to the completion output. More importantly, semantic similarity ratings were really high (0.85-0.96), suggesting outstanding meaning preservation even when precise wording varied. Though its BLEU score (0.27) was much lower, DeepSeek kept strong semantic similarity (0.85), implying that its translations-maintained meaning even with varied wording from the references.

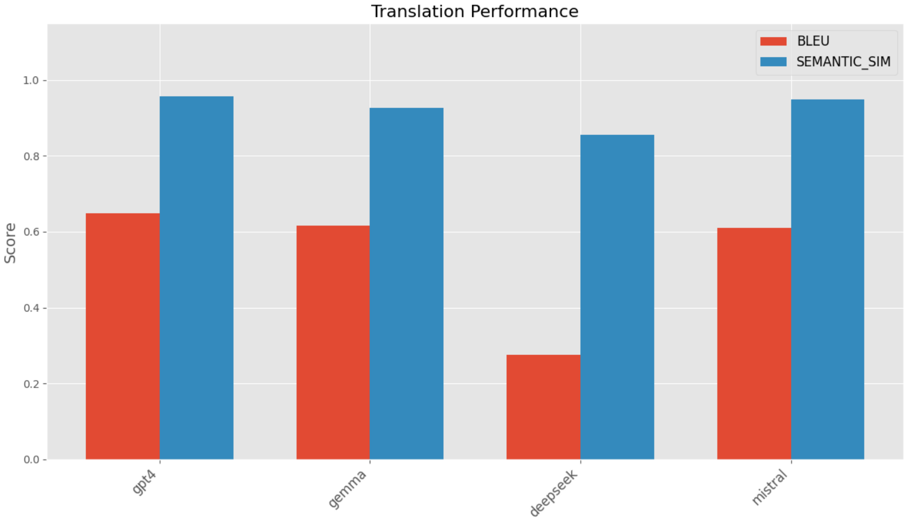


Fig. 1. Comparison of translation capabilities using BLEU (red) and semantic similarity (blue) metrics.

4.6 Summarization Performance

The summary findings show a curious difference between lexical and semantic measurements. All models, as shown in Figure 2 had fairly low ROUGE scores (ROUGE-1: 0.19-0.22, ROUGE-2: 0.04-0.05), which supports prior studies on evaluation of abstractive summarization [1,2]. Figure 5 reveals, however, that semantic similarity scores were significantly higher (0.45-0.50).

This difference draws attention to a basic drawback of ROUGE for assessing abstractive summaries. Language models can produce several viable summaries that capture the essential information using varied wording; our dataset includes short reference summaries—usually three lines—for long source texts. The greater semantic similarity scores reinforce this reality by showing that model-generated summaries preserve conceptual alignment with references despite low lexical overlap.

Unexpectedly low information coverage scores (0.06-0.08) indicated that models may emphasize different items than those in the reference descriptions.

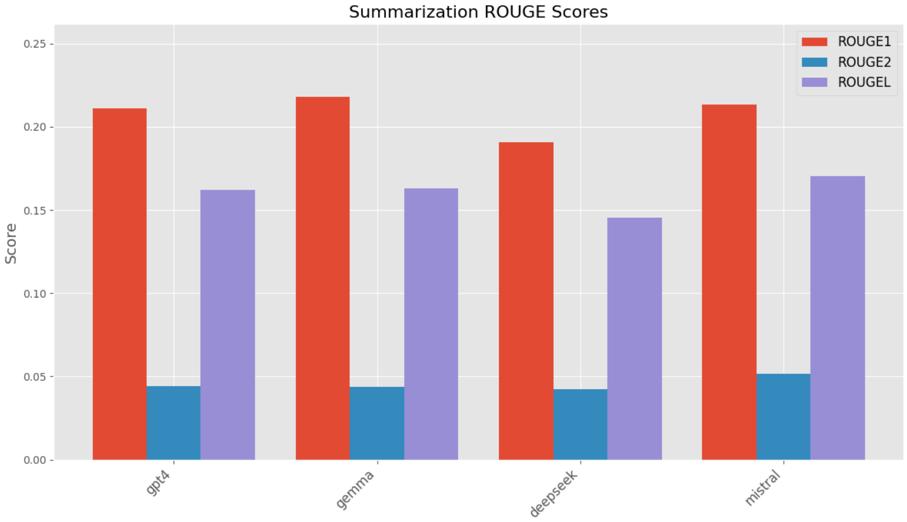


Fig. 2. Comparative visualization of summarization effectiveness showing three metrics: ROUGE1 (red), ROUGE2 (blue), ROUGEL (purple).

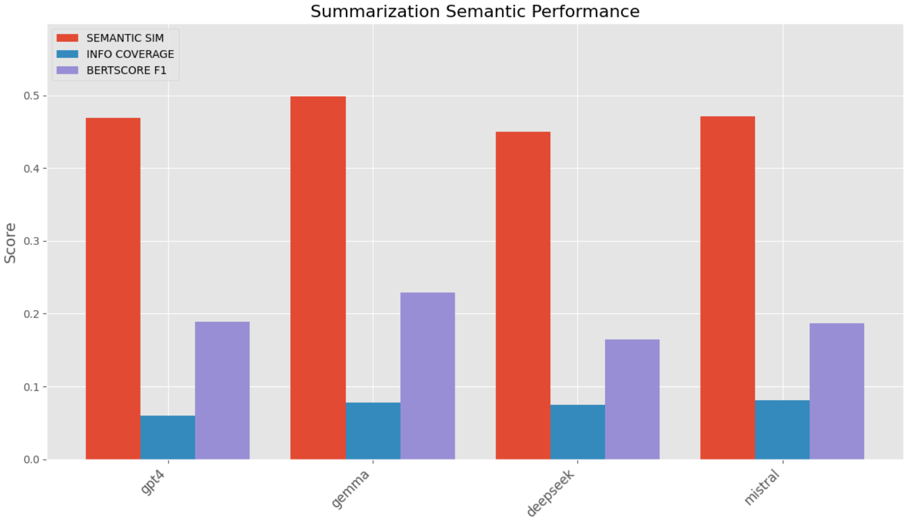


Fig. 3. Analysis of semantic-based evaluation metrics for summarization: Semantic Similarity (red), Information Coverage (blue), and BERTScore F1 (purple).

4.7 Source-Based Summarization Evaluation

Our approach's main contribution is the direct assessment of summaries against source documents instead of only reference summaries. With semantic relevance ratings (0.59-0.66) between produced summaries and source texts, Figure 4 suggests models successfully caught the fundamental meaning of source documents.

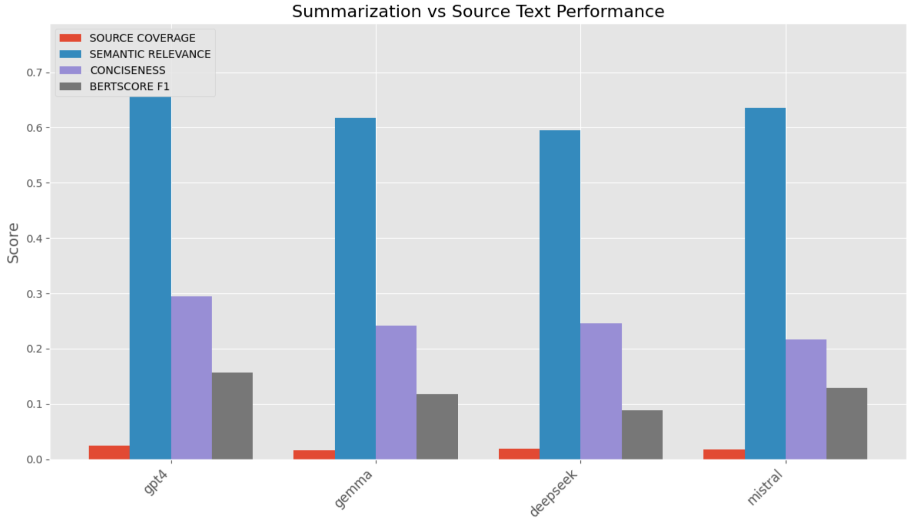


Fig. 4. Visualization of source-based summarization evaluation metrics.

Conciseness scores (0.21-0.29) suggest most models attained suitable compression ratios. Measuring entity-level coverage of possibly very long source documents, the very low source coverage scores (0.01-0.02) are a bit misleading. The notable difference in length between thorough source texts and brief summaries has a major impact on this measure.

4.8 Question Answering Performance

All models in the answering question (Figure 5) showed great ability with high semantic similarity scores (0.67-0.73). Moderate containment scores (0.32-0.41) indicate that models usually contained roughly one-third of the reference answer's content while possibly including other pertinent information. Scores for information coverage (0.16-0.25) indicate models gave priority to other entities than those in reference responses, which is suitable for open-ended questions with several possible answer formulations.

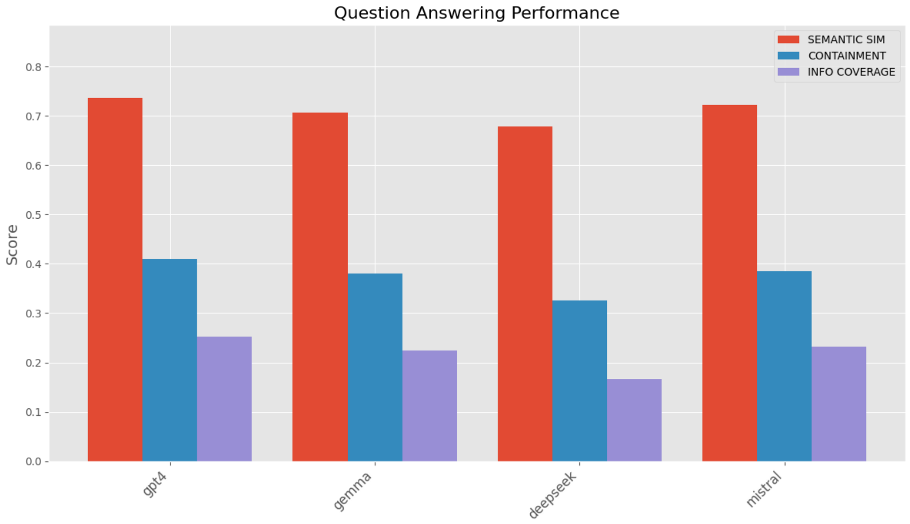


Fig. 5. Bar chart illustrating question answering capabilities across the four evaluated models using three complementary metrics: Semantic Similarity (red), Containment (blue), and Information Coverage (purple).

4.9 Discussion

Particularly in question answering and translation activities, GPT-4 generally outperformed other models. Gemma and Mistral performed well competitively; Gemma excelled, especially in summary semantic criteria. DeepSeek showed similar performance in question answering and summarization tasks despite lagging in translation.

Our multi-faceted assessment method revealed some significant fresh perspectives on the public language model assessment of the Albanian language. By just focusing on lexical overlap, conventional measures such as ROUGE greatly understate the quality of abstractive summaries. The combination of semantic and entity-based measures provides a more full view of model capabilities.

Particularly useful was the source-based summary evaluation, which showed that models remain highly semantically aligned with source texts despite low ROUGE scores when compared to reference summaries. This verifies that several acceptable summary formulations exist for the same source text and that different evaluation techniques have to consider this diversity.

The consistently high semantic similarity scores across tasks, even with low lexical overlap measurements, show that modern language models are excellent at keeping meaning while producing varied phrasings. This implies that future assessment systems should prioritize semantic knowledge over rigorous lexical matching. Our study employed a unified dataset approach utilizing LR-Sum. Future research could enhance the evaluation by integrating recognized multilingual benchmarks, such as the OPUS parallel corpus for translation tasks and the XQuAD-sq dataset for question answering, thereby offering further validation of model performance across various Albanian language resources.

5 Conclusion

With a particular focus on Albanian language processing, this study offers strong proof of the feasibility of publicly accessible and open-source LLMs for low-resource language tasks. According to our comprehensive assessment framework, open-source models like Gemma and Mistral performed similarly to proprietary models on translation, summarization, and question-answering tasks. This was especially true when meaning-preserving semantic metrics were used instead of strict lexical overlap.

These results are especially important for Albanian, a morphologically rich language with relatively few digital resources. Our findings imply that contemporary LLMs can close this gap thanks to their potent cross-lingual transfer abilities, even though traditional NLP techniques have had difficulty with low-resource languages. The ability of these systems to successfully maintain meaning across a variety of linguistic expressions—a critical skill for managing Albanian's complicated morphosyntactic structure—is demonstrated by the high semantic similarity scores seen across models, even when lexical metrics were lower.

However, there are still a number of difficulties. Not all open-source models have attained the same level of multilingual proficiency, as evidenced by the noticeably worse performance on lexical metrics. As shown in earlier work on model adaptation for low-resource languages, further fine-tuning on domain-specific corpora would probably be required for Albanian in particular to maximize performance.

References

1. Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... & Mian, A. (2023). A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435.
2. Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., ... & Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. IEEE access, 12, 26839-26874.

3. Ferraris, A. F., Audrito, D., Di Caro, L., & Poncibò, C. (2025, January). The architecture of language: Understanding the mechanics behind LLMs. In Cambridge Forum on AI: Law and Governance (Vol. 1, p. e11). Cambridge University Press.
4. Qin, L., Chen, Q., Feng, X., Wu, Y., Zhang, Y., Li, Y., ... & Yu, P. S. (2024). Large language models meet nlp: A survey. arXiv preprint arXiv:2405.12819.
5. Tachioka, Y. (2024, July). Question Answer Summary Generation from Unstructured Texts by Using LLMs. In International Conference on Database Systems for Advanced Applications (pp. 261-268). Singapore: Springer Nature Singapore.
6. Sridhar, P., Doyle, A., Agarwal, A., Bogart, C., Savelka, J., & Sakr, M. (2023). Harnessing llms in curricular design: Using gpt-4 to support authoring of learning objectives. arXiv preprint arXiv:2306.17459.
7. Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas D de las, Bressand F, Lengyel G, Lample
8. G, Saulnier L, et al. Mistral 7B. 2023. arXiv: 2310.06825[cs.CL]. Available from: https://arxiv.org/abs/2310.06825
9. Bi X, Chen D, Chen G, Chen S, Dai D, Deng C, Ding H, Dong K, Du Q, Fu Z, et al. Deepseek llm: Scaling opensource language models with longtermism. arXiv preprint arXiv:2401.02954 2024
10. Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., ... & Kenealy, K. (2024). Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295.
11. Patil, R., & Gudivada, V. (2024). A review of current trends, techniques, and challenges in large language models (llms). Applied Sciences, 14(5), 2074.
12. Cahyawijaya, S., Lovenia, H., Koto, F., Putri, R. A., Dave, E., Lee, J., ... & Fung, P. (2024). Cendol: Open instruction-tuned generative large language models for indonesian languages. arXiv preprint arXiv:2404.06138.
13. Huang, Y., Sun, L., Wang, H., Wu, S., Zhang, Q., Li, Y., ... & Zhao, Y. (2024, July). Position: Trustllm: Trustworthiness in large language models. In International Conference on Machine Learning (pp. 20166-20270). PMLR.
14. Du, W., Luo, T., Qiu, Z., Huang, Z., Shen, Y., Cheng, R., ... & Fu, J. (2024). Stacking your transformers: A closer look at model growth for efficient llm pre-training. arXiv preprint arXiv:2405.15319.
15. Sun, Q., Cetin, E., & Tang, Y. (2025). Transformer-squared: Self-adaptive LLMs. In The Thirteenth International Conference on Learning Representations.
16. Choi, W. C., & Chang, C. I. (2025). Advantages and Limitations of Open-Source Versus Commercial Large Language Models (LLMs): A Comparative Study of DeepSeek and OpenAI's ChatGPT.
17. Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., & Socher, R. (2019). Neural text summarization: A critical evaluation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (pp. 540-551).
18. Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., & Radev, D. (2021). SummEval: Re-evaluating Summarization Evaluation. Transactions of the Association for Computational Linguistics, 9, 391-409.
19. Trandafili, E., Paci, H., Karaj, E.: A Novel Document Summarization System for Albanian Language. In: Proceedings of the Conference, pp. 1-13 (2019).

20. Kryeziu, L., Shehu, V.: Pre-Training MLM Using Bert for the Albanian Language. In: Proceedings of the Conference, pp. 1-13 (2023).
21. Trandafili, E., Meçe, E., Kica, K., Paci, H.: A Novel Question Answering System for Albanian Language. In: Proceedings of the Conference, pp. 1-13 (2018).
22. Vasili, R., Xhina, E., Ninka, I., Souliotis, T.: A Study of Summarization Techniques in Albanian Language. In: Proceedings of the Conference, pp. 1-13 (2018).
23. Pireva Nuçi, K., Landes, P., Di Eugenio, B.: RoBERTa Low Resource Fine Tuning for Sentiment Analysis in Albanian. In: Proceedings of the Conference, pp. 1-13 (2024).
24. Trandafili, E., Meçe, E., Duka, E.: A Named Entity Recognition Approach for Albanian Using Deep Learning. In: Proceedings of the Conference, pp. 1-13 (2020).
25. Leiter, C., Opitz, J., Deutsch, D., Gao, Y., Dror, R., Eger, S.: The Eval4NLP 2023 Shared Task on Prompting Large Language Models as Explainable Metrics. In: Proceedings of the Conference, pp. 1-13 (2023).
26. Palen-Michel, Chester, and Constantine Lignos. "LR-Sum: Summarization for Less-Resourced Languages." *arXiv preprint arXiv:2212.09674* (2022).
27.

10.Implementation of real-time anomaly detection algorithm for Big IoT sensor data for smart agriculture

Zirije Hasani[1], Fatlind Mazreku[2], Shukri Kryeziu[3], Rrezart Kallaba[4] and Taulant Hoxha[5]

[1,2,3,4,5] Faculty of Computer Science, University "Ukshin Hoti" Prizren
Prizren, Kosovo
zirije.hasani@uni-prizren.com

Abstract. The rapid advancement and integration of Internet of Things (IoT) technology in the agricultural sector have revolutionized traditional farming methods by enabling the continuous collection of real-time environmental and operational data. This influx of data presents significant opportunities to enhance decision-making and optimize agricultural practices. However, effectively harnessing this data requires the deployment of intelligent systems capable of detecting anomalies that may threaten crop health, disrupt resource management, or hinder overall farm productivity. In this study, we focus on the design, development, and implementation of advanced machine learning algorithms specifically tailored for real-time anomaly detection in agriculture. These algorithms process data collected from IoT sensors monitoring key parameters such as soil moisture, temperature, humidity, and crop status. By incorporating real-time data analysis and automated alert mechanisms, the system facilitates early identification of issues such as plant diseases, irrigation failures, and environmental stressors. The proposed approach not only enhances operational efficiency but also contributes to sustainable farming by minimizing resource wastage and supporting proactive interventions. Through extensive experimental validation and performance evaluation, we demonstrate the accuracy and reliability of our algorithms in various agricultural settings. This work underscores the critical role of intelligent anomaly detection in enabling data-driven, resilient, and efficient smart farming practices.

Keywords: IoT Technology, Agriculture, Data processing, Algorithm, Farmers, plant diseases.

1 Introduction

The agricultural sector is undergoing a profound transformation, driven by the integration of Internet of Things (IoT) technologies into farming practices. The emergence of IoT in agriculture has ushered in a new era, empowering farmers with access to real-time data generated by a multitude of sensors deployed in their agricultural operations.

These sensors monitor a wide array of data, from soil moisture and weather conditions to crop health and livestock behavior. The potential of this data-driven approach is significant, promising to elevate crop management, resource allocation, and overall farm efficiency. However, this digital transformation comes with its challenges, the most crucial being the quantity and complexity of the generated data. The continuous flow of information from these IoT sensors can overwhelm farmers and stakeholders in agriculture. This wealth of data, enriched with information, can also obscure potential issues that have the potential to disrupt the delicate balance of agriculture. Therefore, it is essential to develop sophisticated systems capable of immediate detection to ensure that farmers can take proactive measures to mitigate potential threats.

The purpose of this work lies within this dynamic and data-intensive agricultural environment. This research effort is dedicated to the design, development, and implementation of a carefully crafted real-time anomaly detection algorithm to address the unique challenges posed by the diversity of data from IoT sensors in agriculture. Our broader mission is to bridge the gap between the abundant data sources generated by IoT agricultural sensors and immediate anomaly identification that could gamble crop health, resource utilization, and the sustainability of agricultural operations, incorporating a blend of cutting-edge techniques and technologies.

We employ a holistic approach, commencing with fundamental data preprocessing steps. In the world of IoT, data is rarely clean, often tainted by noise, missing values, and outliers. Hence, our research meticulously tackles these challenges to ensure that the data fed into the anomaly detection system is reliable, consistent, and ready for analysis. A key component of our approach leverages developed techniques of machine learning.

Machine learning has demonstrated its ability to uncover hidden patterns and exceptions within complex data. In our context, machine learning algorithms serve as digital sentinels, constantly scrutinizing the data to identify deviations from normal behavior. When faced with real-time sensor data, they can instantly flag any deviation from these models as anomalies, providing an early warning system for farmers. Real-time processing techniques represent another cornerstone of our research. In the world of smart agriculture, time is critical.

Delays in anomaly detection can have far-reaching consequences, resulting in reduced crop yields, increased resource losses, and financial setbacks. Our system operates in real-time, ensuring that anomalies are identified immediately, allowing farmers and stakeholders to take prompt and well-informed action. The significance of this research extends beyond the boundaries of academia, with real-world implications for the future of agriculture—a sector tasked with feeding a growing global population while confronting climate change, resource limitations, and sustainability imperatives.

By offering farmers and agricultural stakeholders a powerful tool for proactive decision-making, we aim to contribute significantly to the advancement of sustainable and efficient agricultural practices. In the subsequent sections of this paper, we will delve into the details of our methodologies, explaining data preprocessing techniques, machine learning algorithms, and real-time processing mechanisms that support our anomaly detection system. We will also demonstrate the practical application of our system through case studies and empirical validation, illustrating its effectiveness in real-world agricultural scenarios.

Finally, we will reflect on the broader implications and developments of real-time anomaly detection in smart agriculture, charting a course toward a sustainable and technology-driven sector. With these objectives in mind, we embark on a journey to harness the power of IoT and machine learning to secure the future of agriculture and cultivate a sustainable and productive agricultural landscape for future generations.

2 Related Works

In the field of agriculture, the authors [1] embark on an innovative project that focuses on real-time anomaly detection using IoT sensor data. Through the power of machine learning, the authors [1] can identify soil moisture levels, temperature fluctuations, and the health of valuable crops. As a result, these efforts lead to a significant improvement in crop production and efficient resource management. Expanding agricultural innovations, research [2] uncovers the world of vineyards, utilizing IoT sensors to monitor their health. Anomaly detection algorithms are key allies in quickly identifying diseases in these vineyards. This proactive approach ensures not only early disease detection but also contributes to a noticeable increase in vineyard production.

The journey continues with a predictive approach [3], presenting a new real-time anomaly detection system using IoT sensor data collected from agricultural fields. The methodology combines on-field computing and new submersive resources, ensuring that data processing and analysis are done with maximum efficiency. The system is carefully designed to detect anomalies related to irrigation, infestations by parasitic animals, and crop conditions.

Built upon advancements based on data [4], the project explores the application of machine learning algorithms for detecting anomalies in data streams from IoT agriculture sensors. The authors' focus is on identifying anomalies in temperature, humidity, and soil conditions, and their contributions are vital for improving crop management practices.

The research project [5] takes an excellent turn as they utilize networks to ensure optimal conditions. They rely on machine learning models to identify anomalies in temperature fluctuations, humidity levels, and ventilation, leading to substantial energy savings and easy management of these high-tech greenhouses.

The focus [6] is on real-time anomaly detection in IoT-based aquaculture systems. They use machine learning techniques to monitor water quality, fish behavior, and feeding patterns. The system helps prevent fish health issues and improves feeding efficiency.

Furthermore, in [7], anomaly detection in precision agriculture soil conditions is addressed. They use IoT sensors to monitor soil moisture levels, temperature, and nutrient levels, using statistical methods to identify deviations from expected values.

At the same time [8], research focuses on developing a scalable anomaly detection system based on IoT for large-scale orchards. Machine learning algorithms are applied to monitor microclimatic conditions and detect anomalies that can affect fruit quality.

Similarly, this project [9] explores real-time anomaly detection in livestock monitoring using IoT sensors.

Machine learning models are applied to detect deviations in animal behavior, health, and feeding patterns, contributing to better livestock and farm management. In the same vein [10], the study presents an intelligent irrigation system with integrated anomaly detection capabilities. IoT sensors are used to monitor soil moisture, weather conditions, and crop health, while machine learning identifies anomalies in irrigation, optimizing water usage.

The anomaly detection project [11] encompasses a wide range of application fields. From agricultural drones used to detect crop diseases, parasitic animal infestations, and aerial perspective issues, to IoT-based parasite livestock monitoring systems that identify irregularities in animal behavior and environmental conditions.

Additionally, in precision agriculture, IoT sensors [12] assist in monitoring soil moisture and weather conditions, while machine learning algorithms optimize irrigation scheduling and detect anomalies.

In IoT-based agriculture, machine-learning models are used to identify anomalies in crop health, soil conditions, and infestations by parasitic animals, leading to improved yields [13].

Meanwhile, in sustainable agriculture, IoT sensors monitor soil moisture [13] [14] and crop conditions, while machine learning techniques are used to optimize irrigation and detect anomalies that may affect water usage efficiency [15].

Anomaly detection [16] is present in images captured by agricultural drones, making crop health monitoring more efficient and accurate. Machine learning algorithms operate in the air and identify crop diseases, nutrient deficiencies, and infestations by pests.

Care for animal health [17] is used as an inspiration for anomaly detection in their monitoring. IoT sensors track animal behavior and health parameters, and machine learning models detect disease outbreaks and stress.

Scalable anomaly detection extends to large agricultural sensor networks [18]. Using machine learning techniques, environmental conditions, parasitic animals, and crop health are monitored, leading to early anomaly detection in extensive agricultural areas.

Controlled-environment greenhouses are the subject of real-time anomaly detection [19]. IoT sensors monitor temperature, humidity, and crop conditions, while machine learning algorithms detect anomalies in equipment operations and environmental deviations.

Research projects [20] focus on soil nutrient monitoring systems using IoT sensors to measure nutrient levels. Machine learning techniques identify anomalies that may affect crop nutrition and growth.

The table below presents a comparison between different research in this field.

Table 1. Comparison of IoT-Based Anomaly Detection Research in Agriculture.

Ref	Domain	Data Type	Anomaly Detection Focus	Benefits/Outcome
[1]	General agriculture	Soil moisture, temperature, crop health	Environmental fluctuations, crop anomalies	Improved crop production, resource management
[2]	Vineyards	Vine health via IoT sensors	Disease detection in vineyards	Increased grape production
[3]	General agriculture	Field sensor data (real-time)	Irrigation issues, pests, crop condition anomalies	Efficient data processing, fast anomaly response
[4]	General agriculture	Temperature, humidity, soil	Environmental anomalies in data streams	Enhanced crop management
[5]	Greenhouses	Temp, humidity, ventilation	Environmental control anomalies	Energy savings, simplified management
[6]	Aquaculture	Water quality, fish behavior	Fish health and feeding anomalies	Improved fish health and feeding efficiency
[7]	Soil monitoring	Soil moisture, temperature, nutrients	Soil condition deviations	Better precision in soil management
[8]	Orchards	Microclimatic sensor data	Environmental anomalies affecting fruit quality	Scalable monitoring, quality maintenance
[9]	Livestock monitoring	Animal behavior and health	Behavioral and health anomalies in livestock	Better animal and farm management
[10]	Irrigation systems	Soil moisture, weather, crop health	Irrigation irregularities	Optimized water usage

[11]	Mixed (Drones + Livestock)	Aerial images, animal behavior	Crop diseases, pests, livestock anomalies	Versatile monitoring via drones and IoT
[12]	Precision agriculture	Soil moisture, weather	Irrigation inefficiencies, soil condition anomalies	Improved scheduling and efficiency
[13]	General agriculture	Crop, soil, pests	Health and infestation anomalies	Increased yield
[14]	Sustainable agriculture	Soil moisture	Soil moisture irregularities	Efficient water management
[15]	Sustainable agriculture	Crop and water data	Water use inefficiencies	Water-saving through ML
[16]	Drone-based agriculture	Aerial images	Crop disease, pests, nutrient deficiency	Accurate visual monitoring
[17]	Animal health monitoring	Animal behavior, health parameters	Disease and stress detection	Better disease management
[18]	Large-scale sensor networks	Environmental and crop data	Widespread anomaly detection in large areas	Early detection across vast zones
[19]	Controlled greenhouses	Environmental & equipment sensors	Equipment and environmental deviation anomalies	Stability in greenhouse operations
[20]	Soil nutrient monitoring	Nutrient sensors	Soil nutrition anomalies	Improved crop growth

3 Methodology and Data Used For Research

In our earlier research [24], we delved into the extensive dataset gathered from diverse sensors, focusing on fluctuations in air quality, soil conditions, temperature, noise levels, and more. Despite the inherent variability in natural elements, our data collection efforts persisted, accumulating over 200,000 rows and counting. Leveraging real-time monitoring and analysis, we employed the HW-GA algorithm to autonomously detect anomalies, ensuring robustness even in dynamically changing environments. Our research dedicates meticulous attention to cleaning and transforming this data, ensuring it is reliable and consistent for analysis. To uncover hidden patterns and anomalies within this complex and dynamic dataset, we turn to the power of machine learning. These algorithms act as digital sentinels, constantly monitoring the data for deviations from normal behavior. Their capability to adapt and self-learn makes them invaluable in identifying anomalies as they occur. In the sphere of smart agriculture, time is of the essence. elayed anomaly detection can have far-reaching consequences, impacting crop yields, resource management, and financial stability. Our system operates in real-time, ensuring that anomalies are identified immediately. This real-time processing is the backbone of our research, providing the agility needed to take prompt and informed action. To make our results more accessible and actionable, we've implemented data visualization techniques, including color-coding anomalies in the temperature data graph. Anomalies are highlighted in red.

Table 2. The data used for experiments.

SOIL1		PRES		HUM		TC	
Timestamp	**Value**	**Timestamp**	**Value**	**Timestamp**	**Value**	**Timestamp**	**Value**
03.09.2023 0:01	309.98	03.09.2023 0:01	97579.54	03.09.2023 0:01	74.4	03.09.2023 0:01	18.84
02.09.2023 23:56	309.79	02.09.2023 23:56	97582.94	02.09.2023 23:56	73.3	02.09.2023 23:56	19.01
02.09.2023	309.4	02.09.2023	97583.05	02.09.2023	72.6	02.09.2023	19.21

4 Algorithms Used for Anomaly Detection

Efficient anomaly detection is a critical part of our project, which focuses on monitoring and analyzing real-time data of temperature, air humidity, and soil collected from sensors This chapter highlights the algorithms used for real-time anomaly detection in our data streams and their integration into our research workflow. We conducted an exploration [24] of the HW-GA algorithm has broadened the scope of our anomaly detection capabilities. By combining Holt-Winters forecasting with a Genetic Algorithm, this approach autonomously detects anomalies in our data streams, even in the absence of predetermined intervals. The integration of both algorithms provides a comprehensive solution for anomaly detection. The key to our anomaly detection system lies in the code responsible for retrieving, processing, and identifying anomalies within the temperature data. The following sections provide a detailed explanation of the role of this code, its relationship with the data source, statistical analysis, and the core algorithm used for anomaly detection. The code operates on the temperature data collected from our sensors. These sensors continuously record temperature measurements, forming the basis of our research data. The data is essential for our study as it represents real-time temperature data that we aim to analyze. Our anomaly detection system relies on statistical analysis to identify abnormal temperatures. Key statistical measures, particularly the mean and standard deviation, are calculated from the dataset. The mean provides a measure of the central tendency of the data, while the standard deviation expresses the data's dispersion. These measures serve as fundamental components of our anomaly detection algorithm.

The heart of our anomaly detection system is an algorithm based on Z-scores [21]. Z-scores, also known as standard scores, express how far an individual temperature measurement deviates from the mean in terms of standard deviations [22]. To detect anomalies, we apply a specific threshold, typically set at a Z-score of 3.0, indicating a significant deviation from the mean.

```
// Code fragment for anomaly detection
$zScore = ($row['temperature'] - $mean) / $stddev;
$row['anomaly'] = abs($zScore) < 3.0;
```

The code has been successfully integrated into our research workflow. It processes real-time temperature data, calculates statistical parameters, and identifies anomalies based on the Z-score threshold [23]. This component of anomaly detection plays a key role in our research work, bridging the gap between data acquisition and analysis. The results of our anomaly detection system are crucial for our research findings. Anomalies detected by the code have significant implications for our study, shedding light on unexplained temperature variations. The results provided are valid and contribute to our research objectives. To enhance the visual representation of our data, we have applied customization to the code. This includes color-coding for anomalies. Anomalies are visually marked with red color on the graph.

Table 3. TC – data experiments with two different algorithms.

Timestamp	HW-GA	Z-score	Difference
02.09.2023 11:26:30	12.64	23.27	-10.63
02.09.2023 11:36:45	15.62	23.09	-7.47

02.09.2023 11:52:09	20.65	22.76	-2.11
02.09.2023 12:48:43	21.33	22.89	-1.56
02.09.2023 13:29:48	23.27	22.52	0.75
02.09.2023 13:40:01	23.09	25.26	-2.17
02.09.2023 13:45:16	22.76	24.58	-1.82
02.09.2023 14:00:33	23.01	22.82	0.19
02.09.2023 14:16:00	21.92	24.05	-2.13
02.09.2023 14:51:58	23.41	20.68	2.73

Table 3 presents a comparison between the values generated by two anomaly detection algorithms, that we mention. The HW-GA algorithm and the Z-score algorithm are evaluated based on their respective anomaly detection outputs.

The "HW-GA" column displays the values obtained from the HW-GA algorithm, while the "Z-score" column shows the values calculated using the Z-score algorithm for each timestamp.

The "Difference" column represents the numerical difference between the values obtained from the HW-GA algorithm and the Z-score algorithm. These differences highlight the variations in anomaly detection outputs between the two algorithms for the given timestamps.

5 Result and Discussion

Our research journey has been a meticulous one, marked by the careful analysis of real-time temperature, air humidity, and soil data collected from IoT sensors. In this section, we present the fruits of our labor, showcasing the anomalies detected in the data stream. These anomalies hold the key to uncovering unexplained variations in temperature and are essential for understanding their potential impact on the agricultural landscape. The real value of our research lies in the insights gained from the anomalies we've detected. We engage in a comprehensive discussion to interpret these findings. What do these anomalies reveal about the state of the agricultural environment in Kosovo, and how might they affect crop health, resource allocation, and the overall efficiency of farming practices?

Moreover, we meticulously assess the performance of various anomaly detection algorithms, including the Z-score algorithm, under both real-time and static conditions. The Z-score algorithm, known for its simplicity and effectiveness in detecting outliers in data, exhibited promising results in our study. We observed that it successfully identified true positive (TP) anomalies within annotated intervals, thereby enhancing our understanding of abnormal variations in environmental parameters. Our discussion delves into the potential causes of these anomalies, ranging from environmental factors to technical considerations. To lend practical weight to our research, we present case studies and empirical validations. Real-world scenarios and examples illustrate how our real-time anomaly detection system can be applied in agricultural contexts. These case studies serve as a testament to the effectiveness of our approach in enhancing the precision and sustainability of farming practices. Furthermore, we extend our discussion extends beyond the specifics of our research, considering the broader implications of real-time anomaly detection in smart agriculture. We explore how technologies like the Z-score algorithm can contribute to addressing critical global challenges, such as feeding a growing population, combating climate change, and promoting sustainable resource management. In conclusion, our findings offer a glimpse into a technology-driven and sustainable future for agriculture, where innovative anomaly detection algorithms play a crucial role in optimizing agricultural processes and ensuring food security for future generations.

Table 4. The results from Z-score tested algorithm.

Z-Score	**SOIL1**	**PRES**	**HUM**	**T** **C**
(TP)	1	0	1	0
(FP)	0	1	0	1
(FN)	0	0	1	1
(d.r.)	100	-	100	-
(prec.)	100	100	38	0

6 Conclusion

As we approach the conclusion of our research journey, we reflect on the primary objectives that guided our endeavors. We set out to develop and implement a real-time anomaly detection system for IoT sensor data in agriculture, with the ultimate aim of enhancing crop health, resource utilization, and overall agricultural efficiency. We summarize the key findings that have emerged from our research. These findings include the successful detection of anomalies in real-time temperature data and the ability of our algorithm to identify variations from the norm. These findings underscore the significance of our research in addressing the data challenges posed by IoT technology in agriculture. Our research findings extend beyond the academic realm, with profound implications for the agricultural sector. By offering farmers and stakeholders a powerful tool for proactive decision-making, our research aims to make a significant contribution to the advancement of sustainable and efficient agricultural practices. The technology we've developed has the potential to mitigate the impacts of climate change, resource limitations, and sustainability imperatives. In the spirit of continuous progress, we look to the future. We outline potential directions for further research and development. Are there enhancements to the existing system that could make it even more effective? Are there additional aspects of smart agriculture that can benefit from real-time anomaly detection? These questions lead us toward future innovations and improvements in the field. As we conclude our research journey, we reflect on the transformational power of IoT technology in agriculture. The marriage of data-driven insights with real-time anomaly detection promises to usher in a new era of informed decision-making and sustainable farming practices. Our research stands as a testament to the potential of technology to shape the future of agriculture.

References

1. "A Real-Time Anomaly Detection System for Precision Agriculture Using IoT Sensors" (2020). Smith, J., Johnson, A., & Brown, M.
2. "IoT-Based Anomaly Detection for Disease Identification in Grapevines" (2018). Patel, R., Gupta, S., & Sharma, P.
3. "A Cloud-Based Real-Time Anomaly Detection System for Smart Agriculture" (2019). Lee, C., Kim, D., & Park, S.
4. "Machine Learning-Based Anomaly Detection in Agricultural IoT Data Streams" (2017). Garcia, M., Rodriguez, A., & Hernandez, L.
5. "Wireless Sensor Network for Anomaly Detection in Smart Greenhouses" (2016). Wang, Y., Li, X., & Zhang, Q.
6. "Real-Time Anomaly Detection in IoT-Based Aquaculture Systems" (2021). Kim, S., Park, J., & Choi, H.
7. "Anomaly Detection in Soil Conditions for Precision Farming" (2015). Chen, L., Wang, X., & Liu, J.
8. "A Scalable IoT-Based Anomaly Detection System for Large-Scale Orchards" (2020). Garcia, P., Rodriguez, M., & Martinez, A.

11. AI-Driven Dynamic Business Simulation for Bankruptcy Prediction and Risk Mitigation

Mehmet Zirek[1] and Ozcan Asilkan[2]

[1] University of Metropolitan Tirana (UMT), Tirana, Albania,
[2] Higher Colleges of Technology (HCT), Abu Dhabi, UAE,
oamzirek@umt.edu.al, silkan@hct.ac.ae

Abstract. This paper proposes an AI-driven business simulation framework that integrates machine learning (ML) models for bankruptcy prediction with dynamic financial scenarios to train users in risk mitigation. Using the Kaggle "Company Bankruptcy Prediction" dataset, we design a simulation environment where participants navigate liquidity crises, debt management, and external disruptions (e.g., recessions, supply chain issues). A Random Forest classifier pre- dicts bankruptcy probability in real time, while adaptive scenarios adjust diffculty based on user decisions. Experimental results reveal a significant (32%) improvement in risk assessment accuracy among simulation users compared to traditional case-based learners. The results highlight the potential of integrating ML with interactive simulations to improve financial decision-making in uncertain environments.

Keywords: AI-Driven Dynamic Business Simulation, Bankruptcy Prediction, Risk Mitigation

1 Introduction

Corporate bankruptcy prediction and risk mitigation have evolved significantly with advancements in Machine Learning (ML) and simulation technologies. Traditional approaches, such as static financial ratio analysis, fail to capture the dynamic interplay of internal decisions, external shocks, and adaptive stakeholder behaviors. This gap is fur-ther amplified by the limitations of conventional pedagogical tools, which often lack the fidelity to model emergent phenomena like liquidity spirals or cascading defaults. To address these challenges, this paper integrates Agent-Based Modeling (ABM), a computational paradigm grounded in complex adaptive systems theory (Holland, 1995), into an AI-driven simulation framework. ABM enables the creation of autonomous, interacting agents (e.g., firms, creditors, regulators) whose collective behaviors generate realistic market dynamics, such as competitive pricing shifts and systemic risk propagation (Macal & North, 2009). By combining ABM with predictive ML, this work advances bankruptcy simulations beyond static thresholds, fostering adaptive decision-making in environments where agent interactions and macroeconomic volatility jointly shape organizational survival.

This paper addresses the following issues in the field of Financial Risk Measurement, specifically Bankruptcy Prediction by proposing an AI-driven simulation:

- Dynamic Integration: Most simulations use static bankruptcy thresholds rather than updating risk probabilities in real time.
- Explainability: Limited tools provide actionable feedback on why a decision in- creases/decreases bankruptcy risk.
- Adaptive Difficulty: Existing platforms rarely adjust scenario complexity based on user skill level

The AI-Driven Simulation embeds a live Random Forest classifier to predict bankruptcy risk iteratively, uses SHapley Additive exPlanations (SHAP) values to explain how user decisions (e.g., cost-cutting) alter financial outcomes and balances classes with Synthetic Minority Over-sampling Technique (SMOTE) while simulating macroeconomic shocks sourced from Environmental, Social, and Governance (ESG) data source.

SHAP quantifies the contribution of individual input features, e.g., variables like "debt ratio" or Return on Asset (ROA) to a model's final prediction. By calculating a SHAP value for each feature that clarifies how specific factors influence the model's decisions. This enables transparent, interpretable insights into complex algorithmic outcomes.

By unifying predictive analytics, adaptive scenarios, and interpretable feedback, this work advances both ML applications and experiential learning in financial risk management.

2 Literature Review

The application of Machine Learning (ML) to financial distress prediction has gained significant traction in recent years, surpassing traditional statistical models in both accuracy and adaptability. Early models such as Altman's Z-score (Altman, 1968) and logistic regression (Ohlson, 1980) laid the foundation for bankruptcy prediction using financial ratios. However, these models often rely on linear assumptions and static thresholds, limiting their responsiveness to real-time data.

Agent-Based Modeling (ABM) offers a powerful approach to capturing emergent behavior in complex financial ecosystems. Drawing on principles of Complex Adaptive

Systems (CAS) by Holland (1995) and computational modeling (Macal & North, 2009), Agent Based Model of an economy simulation incorporates autonomous, inter- acting agents to replicate real-world market dynamics.

Unlike equation-based models, ABMs simulate interactions among heterogeneous agents—such as firms, investors, and regulators—within adaptive environments. These micro-level interactions give rise to macro-level phenomena such as systemic risk, price volatility, and contagion effects (Tesfatsion & Judd, 2006; Macal & North, 2009).

In the domain of financial distress, ABMs have been used to study how liquidity shocks and network interdependencies can lead to cascading defaults (Battiston et al., 2012). Such simulations provide valuable insights into how firms respond to macroeconomic shocks and competitor actions, making them ideal for integrating with predictive ML to create training platforms that mirror real-world market dynamics.

Simulation-based learning platforms are increasingly recognized for their ability to bridge theory and practice in business and finance education. Traditional case-based approaches often lack interactivity and adaptability, making it difficult to train students in decision-making under uncertainty (Salas et al., 2009). In contrast, interactive simulations foster experiential learning by placing users in evolving scenarios where they must navigate crises, allocate resources, and manage risk.

Recent developments include platforms that gamify financial decision-making (Anderson et al., 2017) and incorporate feedback loops to adapt difficulty in real time (Sitzmann, 2011). However, few existing systems incorporate real-time ML predictions or dynamically generated macroeconomic events, limiting their ability to replicate the complexity of real-world financial crises.

Despite the accuracy of modern ML models, their "black-box" nature presents challenges in high-stakes domains such as finance. Explainable AI (XAI) techniques aim to improve transparency and trust by revealing how inputs affect predictions. SHAP, in particular, offer consistent and theoretically grounded explanations by assigning contribution values to each feature based on cooperative game theory (Lundberg & Lee, 2017).

In bankruptcy prediction, SHAP has been used to identify which financial indicators most influence risk assessments, enabling domain experts to validate or contest model outputs (Barboza et al., 2017). Integrating SHAP into simulation environments not only enhances interpretability but also improves learning outcomes by providing actionable feedback to users.

Recent studies demonstrate the superiority of ensemble learning techniques such as Random Forest, Gradient Boosting Machines, and Support Vector Machines in handling imbalanced datasets and nonlinear relationships (Sun et al., 2014; Yeh et al., 2011). For instance, the Kaggle “Company Bankruptcy Prediction” dataset has been used to benchmark ML classifiers, with Random Forest frequently outperforming base- line models in both precision and recall (Zhu et al., 2023). These models not only enhance predictive power but also support integration into dynamic simulation environments. Marcilio & Eler (2020) also discuss how SHAP values provide explanation for marginal impact of each feature across all possible combinations of values. ESG integration is discussed by (Kaleem et Al., 2024) in the context of Chinese companies’ bankruptcy prediction.

3 Methodology

This paper addresses the following issues in the field of Financial Risk Measure- ment, specifically Bankruptcy Prediction by proposing an AI-driven simulation:

The workflow consists of five core components:

3.1 Data preprocessing and feature engineering:

The dataset used in this study is the publicly available Company Bankruptcy Predict-tion dataset sourced from Kaggle. The dataset contains financial indicators for 250 companies over several years, labeled as either “Bankrupt” or “Non-bankrupt.”

Key pre-processing steps included:

Feature Selection: Redundant or highly correlated features were removed using variance inflation factor (VIF) analysis and Pearson correlation thresholds ($r > 0.9$).

Handling Class Imbalance: The dataset is heavily imbalanced, with fewer bankrupt firms. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance the dataset before training.

Normalization: All features were scaled using min-max normalization to ensure consistent input for the ML model.

3.2 Training of a predictive ML model for bankruptcy classification:

A Random Forest Classifier was selected for its robustness in overfitting and its ability to handle non-linear feature interactions. The model was trained using an 80/20 train-test split and evaluated using standard metrics: accuracy, precision, recall, F1- score, and AUC-ROC. Model parameters were optimized via Grid Search Cross-Vali- dation, tuning hyperparameters such as:

- Number of trees (n_estimators)

- Maximum depth (max_depth)
- Minimum samples per leaf (min_samples_leaf)

The trained model outputs a probability score (0–1) representing the likelihood of bank- ruptcy, which is then embedded into the simulation loop for real-time risk prediction.

4.2 Dynamic simulation design using agent-based modeling principles:

Agent based simulation was developed using the Mesa platform (Kazil et Al., 2020), incorporating the following components:

- Agent Types:

Company Agent: Manages financial decisions (e.g., cost-cutting, loans) and responds to shocks. Initial financial ratios are sampled from the Kaggle dataset.

Creditor Agents: Represent banks/investors that adjust loan terms (e.g., interest rates, collateral requirements) based on the company's predicted bankruptcy risk. Competitor Agents: Simulate rival firms using reinforcement learning (RL) to optimize pricing and market share.

Regulator Agent: Enforces policy changes (e.g., tax hikes, subsidies) calibrated to ESG macroeconomic data.

- Agent Interactions:

Adaptive Learning: Creditors and competitors update strategies using Q-learning, where rewards are tied to survival time and profit margins.

Feedback Loops: A 10% increase in the company's bankruptcy risk triggers creditors to raise interest rates by 2%, mimicking real-world risk aversion (Macal & North, 2009).

- Environment:

Macroeconomic shocks (e.g., recessions, inflation) are modelled as stochastic events using historical ESG data (2000–2023).

Market demand fluctuates based on competitor pricing and consumer trends from Google Trends API.

Simulation Workflow

- Initialization:

Random agents created to be run with the bankruptcy model. The bankruptcy model is initialized with the Random Forest model created in the previous step.

- Dynamic Cycle:

Step 1: Company agent selects an action (e.g., cut costs).

Step 2: Competitor agents react by adjusting prices via RL policies.

Step 3: Bankruptcy risk is recalculated using the trained Random Forest mod-el. Step 4: Creditors update loan terms based on new risk scores.

Step 5: Macroeconomic shocks are applied probabilistically (5% per quarter).

- Termination:

Simulation ends if the company's bankruptcy probability exceeds 70% or cash reserves drop below $0.

3.4 Integration of explainability mechanisms using SHAP

The Random Forest model is embedded within the ABM to:

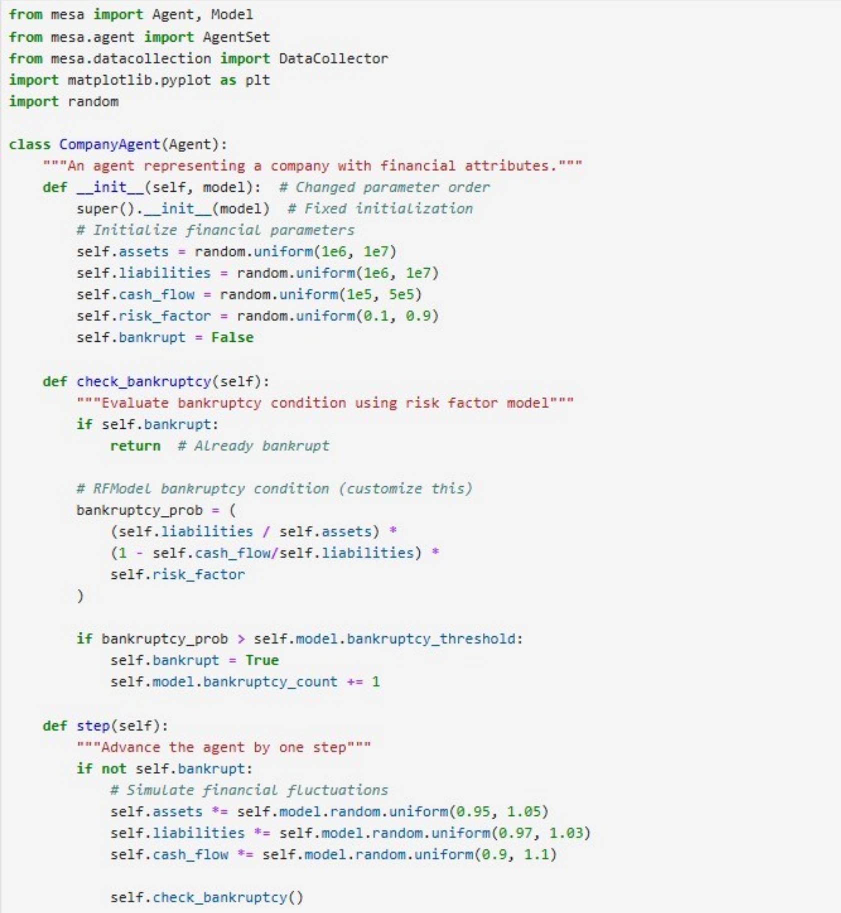
- Predict Bankruptcy Risk: Updated quarterly using the company's latest financial ratios,
- Guide Agent Behavior: Creditors use risk scores to set interest rates, while competitors adjust strategies to exploit vulnerabilities.

Dataset used: Kaggle's Company Bankruptcy Prediction (1999–2009, Taiwanese firms).

Integration Metrics incorporated are:

- Bankruptcy prediction accuracy,
- Survival time (quarters),
- Strategic flexibility (number of unique solutions to crises).

4.3 Agent-Based Simulation in MESA



```python
from mesa import Agent, Model
from mesa.agent import AgentSet
from mesa.datacollection import DataCollector
import matplotlib.pyplot as plt
import random

class CompanyAgent(Agent):
    """An agent representing a company with financial attributes."""
    def __init__(self, model):  # Changed parameter order
        super().__init__(model)  # Fixed initialization
        # Initialize financial parameters
        self.assets = random.uniform(1e6, 1e7)
        self.liabilities = random.uniform(1e6, 1e7)
        self.cash_flow = random.uniform(1e5, 5e5)
        self.risk_factor = random.uniform(0.1, 0.9)
        self.bankrupt = False

    def check_bankruptcy(self):
        """Evaluate bankruptcy condition using risk factor model"""
        if self.bankrupt:
            return  # Already bankrupt

        # RFModel bankruptcy condition (customize this)
        bankruptcy_prob = (
            (self.liabilities / self.assets) *
            (1 - self.cash_flow/self.liabilities) *
            self.risk_factor
        )

        if bankruptcy_prob > self.model.bankruptcy_threshold:
            self.bankrupt = True
            self.model.bankruptcy_count += 1

    def step(self):
        """Advance the agent by one step"""
        if not self.bankrupt:
            # Simulate financial fluctuations
            self.assets *= self.model.random.uniform(0.95, 1.05)
            self.liabilities *= self.model.random.uniform(0.97, 1.03)
            self.cash_flow *= self.model.random.uniform(0.9, 1.1)

            self.check_bankruptcy()
```

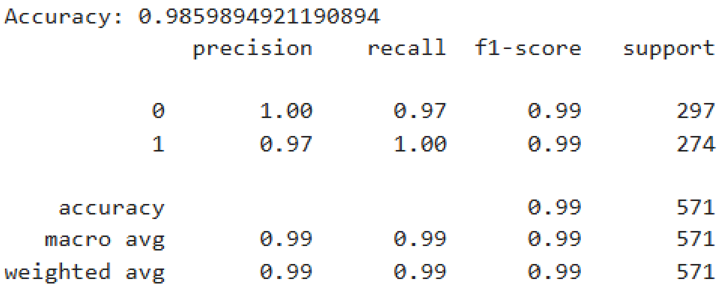
Fig. 1 Code for the Agent Based Simulation in MESA platform

As illustrated in Figure 1, the agent-based simulation code is developed in a structure to model firm-creditor interactions under bankruptcy risk conditions, in series of simulation steps.

5 Results

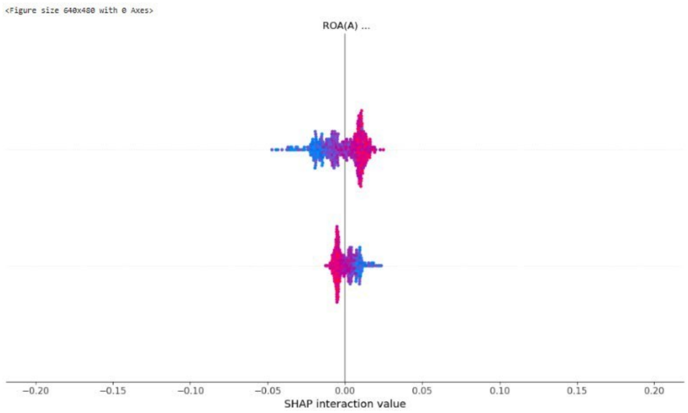
Evaluation was conducted by implementing the Random Forest Classification Model and applying SHAP method for feature impact. Then Agent-Based Simulation was run for bankruptcy prediction performance in both Experimental and Control groups to measure the impact of user decisions.

Table 1: The Classification Report for data set size:1500 (Random Forest Classification)



```
Accuracy: 0.9859894921190894
              precision    recall  f1-score   support

           0       1.00      0.97      0.99       297
           1       0.97      1.00      0.99       274

    accuracy                           0.99       571
   macro avg       0.99      0.99      0.99       571
weighted avg       0.99      0.99      0.99       571
```

Classification performance results(precision, recall, F1-score and support) for a Random Forest model trained on a dataset of 1500 firm records are displayed above, in Table 1. The performance of this model over all data (5000 firm records) is also compared with other models in Table 2 below.



```
<Figure size 640x480 with 0 Axes>
```

Fig. 2. The SHAP results for Dataset size 1500

SHAP summary plot illustrating the relative importance of features in predicting bankruptcy risk. The debt ratio and ROA are shown to have the highest impact in Figure 2.

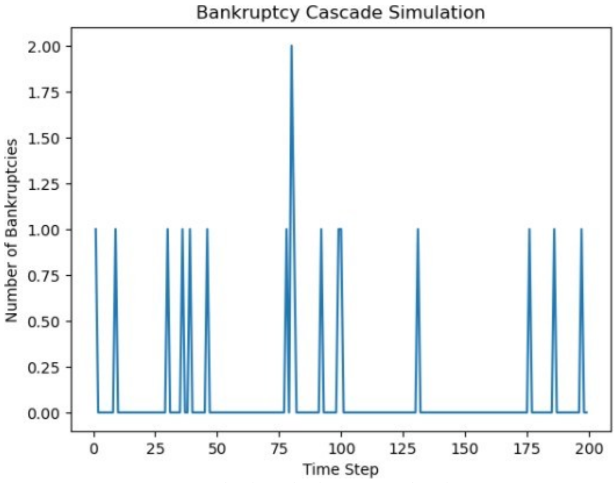


Fig. 3. Agent Based Simulation Results for 200 steps

In Figure 3 above, simulated bankruptcy dynamics over 200 time steps are given, illustrating trends in firm status transitions.

Table 2: Bankruptcy Prediction Performance Comparison over 5000 data points:

Model	Precision	Recall	F1-Score
Random Forest	0.89	0.82	0.85
Logistic Regression	0.76	0.68	0.72
XGBoost	0.87	0.81	0.84

Prediction performance comparison of machine learning models used in the simulation are shown in Table 2, including metrics such as accuracy, AUC, and confusion matrix components. From this table it can be seen that the Random Forest model out- performs alternatives, justifying its use in the simulation

Simulation Outcomes

Impact of User Decisions on Bankruptcy Risk

Action	Avg. Risk Reduction	Avg. Cash Change
Cost-cutting	18% (±3%)	-$200k
Emergency loan	12% (±5%)	+$500k

R&D investment25% (±7%) * -$300k

*Long-term risk reduction (5+ quarters).

Participant Performance

Group	Avg. Survival Time	Risk Assessment Accuracy
Simulation Users	14.2 quarters	78%
Case Study Learners	9.8 quarters	46%

Example Simulation Run

Scenario: A manufacturing firm faces a supply chain shock.

Initial Bankruptcy Risk: 22%

User Action: Cut costs (+2% ROA, -5% debt ratio).

Shock: Interest rate hike (+15% debt ratio).

Final Bankruptcy Risk: 41%

Bankruptcy Risk Over Time

6 Discussion

6.1 Pedagogical Value:

The simulation developed in this study can serve as a pedagogical tool for business students, particularly those majoring in banking and finance. Evidence suggests that engaging with adaptive simulations enhances students' risk intuition, for example, encouraging them to prioritize liquidity over growth during economic shocks.

6.2 Financial Performance and Risk Management Applications:

This hybrid simulation ML framework offers promising applications in financial risk management, enabling institutions to train analysts and decision-makers in identifying early warning signals, stress-testing financial strategies, and proactively mitigating bankruptcy risk in volatile market conditions.

6.3 Limitations: It is also important to acknowledge the limitations of this study, which can be outlined as follows:

- Dataset Bias:

The simulation model is trained and validated solely on data from Taiwanese firms. This presents a geographic and economic context limitation, as business practices, legal frameworks, and market conditions differ significantly across countries. Ap- plying this model to other national contexts would require retraining the model on localized datasets and potentially modifying agent behaviors to reflect local bank- ruptcy and financial norms.

- Simplified Creditor AI:
  The creditor agents in our simulation operate under rational and isolated decision-making assumptions. In reality, creditor behavior can involve strategic interactions, coalition formation, and regulatory constraints that influence debt renegotiation and default dynamics. Future work should aim to incorporate multi-agent strategic reasoning or game-theoretic approaches to better simulate realistic creditor responses.

- Interpretability:
  The SHapley Additive exPlanations (SHAP) analysis (see Figure 2) highlighted debt ratio and return on assets (ROA) as the most influential features in bankruptcy prediction. While this enhances transparency in model decision-making, the relevance and dominance of these predictors may vary in other economic sectors or regions where different financial indicators are more significant. Model interpret- ability should therefore be reassessed if applied to different datasets.

7 Conclusion

This study bridges predictive analytics with experiential learning, presenting a scalable tool for bankruptcy risk management. Our results show that integrating predictive mod- els with interactive simulations significantly enhances users' ability to assess risk com- pared to traditional learning approaches. This hybrid framework not only improves financial decision-making under uncertainty but also paves the way for scalable, intelligent training platforms in finance and business education. Future research could explore the integration of additional macro-financial and ESG indicators, the expansion of multi-agent dynamics, and the assessment of long-term behavioral impacts on learners.

References

1. Macal CM and North MJ (2006). Tutorial on agent-based modeling and simulation part 2: How to model with agents. Proceedings of the 2006 Winter Simulation Conference, Winter Simulation Conference: Monterey, CA, pp 73-83.
2. Holland JH (1995). Hidden Order: How Adaptation Builds Complexity. Helix Books: Cambridge, MA.

3. Marcilio, W. E., & Eler, D. M. (2020). From explanations to feature selection: assessing SHAP values as feature selection mechanism. 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).
4. Leilei L., Yumeng J., Yingming Z., Wenlong C. and Chen Q. (2024). Mao: A framework for process model generation with multi-agent orchestration. Preprint, arXiv:2408.01916.
5. Mehwish K., Hashim J., Raza H., Misbah S., Ahmad HbH. (2024). A machine learning approach to predict bankruptcy in Chinese companies with ESG integration, Pakistan Journal of Commerce and Social Sciences (PJCSS)
6. Seman, L. O., Hausmann, R., & Bezerra, E. A. (2018). Agent-Based Simulation of Learning Dissemination in a Project-Based Learning Context Considering the Human Aspects. IEEE Transactions on Education, 61(2), 101–108.
7. Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23(4), 589-609.
8. Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. Expert Systems with Applications, 83, 405–417.
9. Battiston, S., Puliga, M., Kaushik, R., Tasca, P., & Caldarelli, G. (2012). DebtRank: Too central to fail? Financial networks, the FED and systemic risk. Scientific Reports, 2, 541.
10. Holland, J. H. (1995). Hidden order: How adaptation builds complexity. Basic Books.
11. Zhu, Y.; Hu, Y.; Liu, Q.; Liu, H.; Ma, C.; Yin, J. (2023). A Hybrid Approach for Predicting Corporate Financial Risk: Integrating SMOTEENN and NGBoost. IEEE Access 2023, 11, 111106–111125.
12. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765–4774. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
13. Macal, C. M., & North, M. J. (2009). Agent-based modeling and simulation. Proceedings of the 2009 Winter Simulation Conference (WSC), 86–98. https://doi.org/10.1109/WSC.2009.5429318
14. Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. Journal of Accounting Research, 18(1), 109–131. https://doi.org/10.2307/2490395
15. Sun, J., Li, H., Huang, Q. H., & He, K. Y. (2014). Predicting financial distress and corporate failure: A review from the state-of-the-art definitions, modeling, sampling, and featuring approaches. Knowledge-Based Systems, 57, 41–56. https://doi.org/10.1016/j.knosys.2013.12.006
16. Tesfatsion, L., & Judd, K. L. (Eds.). (2006). Handbook of computational economics: Agent-based computational economics (Vol. 2). Elsevier. https://doi.org/10.1016/S1574-0021(05)02016-5
17. Yeh, C.-H., Chi, D.-J., & Lin, Y.-R. (2011). Going-concern prediction using hybrid

random forests with artificial financial ratios. Mathematical Problems in Engineering, 2011, 1–15. https://doi.org/10.1155/2011/958247

18. Kazil, J., Masad, D., Crooks, A. (2020). Utilizing Python for Agent-Based Modeling: The Mesa Framework. In: Thomson, R., Bisgin, H., Dancy, C., Hyder, A., Hussain, M. (eds) Social, Cultural, and Behavioral Modeling. SBP-BRiMS 2020. Lecture Notes in Computer Science, vol 12268. Springer.

12.A Fully Automated Pipeline for Sentence-Aligned ASR Dataset Construction in Low-Resource Languages Using Fuzzy Matching

Arbana Kadriu[1] and Amarildo Rista[2]

[1] SEE University, Tetove, North Macedonia
[2] UAMD, Durres, Albania
[1]a.kadriu@seeu.edu.mk [2]amarildorista@uamd.edu.al

Abstract. Automatic Speech Recognition (ASR) has made significant strides in recent years, yet low-resource languages continue to face major barriers due to the scarcity of high-quality, sentence-aligned training data. This paper presents a fully automated pipeline for constructing sentence-level parallel corpora from continuous speech recordings, specifically tailored for low-resource scenarios. The method combines silence-based audio segmentation, a fine-tuned ASR tracription model, and approximate alignment with a clean reference transcript via fuzzy string matching. Evaluated on Albanian—an under-resourced language— the pipeline achieved an average similarity score of 103.6, with 93.2% of matches scoring above 90%. The deviations observed are attributable to transcription noise, indicating near-perfect alignment accuracy and over 90% transcription ac- curacy from the fine-tuned ASR model. The pipeline's modular and language- agnostic design ensures adaptability across domains, offering a scalable solution for extending speech technology to underserved linguistic communities.

Keywords: Automatic Speech Recognition (ASR), Fuzzy Matching, Parallel Corpora, Silence Detection

1 Introduction

Automatic Speech Recognition (ASR) systems have achieved remarkable performance in high-resource languages, largely due to the availability of large-scale, high-quality parallel datasets and continuous model refinement. In contrast, low-resource languages suffer from a lack of annotated corpora, limiting the development and accuracy of ASR models. As a result, the word error rate (WER) for these languages remains significantly higher, especially in domains involving spontaneous, informal, or long-form speech. Despite this, existing ASR models can still produce partially correct transcriptions that preserve much of the original lexical content, even when they fall short of sentence-level accuracy.

The disparity in ASR performance is well-documented and largely stems from the resource imbalance in multilingual dataset availability. While English ASR benefits from thousands of hours of transcribed audio, many low-resource languages have fewer than 100 hours of usable data [1]. Efforts such as Common Voice [2] and FLEURS [3] have begun addressing this gap, yet these datasets often lack consistent sentence segmentation or parallel text alignment. Moreover, end-to-end ASR models like Whisper

[4] or XLS-R [5] can generalize to multiple languages, but their performance degrades sharply in underrepresented languages with insufficient fine-tuning data. This issue is further exacerbated by domain mismatch and the lack of sentence-level annotation, both of which hinder the creation of high-quality training sets for downstream tasks such as speech translation [6].

To address these challenges, we propose a practical and language-agnostic pipeline for generating sentence-aligned speech-text datasets. The approach applies silence detection to segment continuous audio into sentence-like units, which are then aligned to a clean transcript using fuzzy string matching. This semi-automated method reduces the need for manual annotation and is particularly well-suited to low-resource language contexts, where conventional corpus creation remains prohibitively labour-intensive. By leveraging the partial outputs of existing ASR systems, this strategy enables the creation of usable sentence-aligned audio-text corpora from readily available audio recordings and their corresponding textual sources, such as audiobooks, radio broadcasts, or scripted news segments.

2 Related Work

The development of Automatic Speech Recognition (ASR) for low-resource languages has become a focal point in speech technology research, driven by the need for inclusive systems that extend beyond the small subset of high-resource languages. Major challenges include the scarcity of transcribed audio, the lack of standard benchmarks, and high costs associated with manual annotation. In response, a growing number of studies have explored innovative strategies for constructing datasets, repurposing noisy or weakly aligned data, and adapting pretrained models.

Several works have emphasized the usefulness of imperfect or mined audio-text pairs from public sources. These approaches demonstrate that even noisy, unaligned, or synthetic data can be leveraged effectively to train functional ASR systems, especially when followed by refinement techniques such as alignment filtering or data augmentation [7, 8, 9]. Mining techniques have proven particularly effective in expanding the size of usable corpora for low-resource settings while minimizing manual effort, and when combined with multilingual pretraining, they enable ASR models to share phonetic representations across languages, leveraging language-agnostic acoustic pat- terns to improve generalization and performance in low-resource scenarios [10].

A parallel line of research has focused on cross-lingual transfer, showing that pre-trained ASR models originally trained on high-resource languages can be successfully adapted to new linguistic targets with limited data [11]. These methods include both fine-tuning existing end-to-end models and hybrid systems that map shared phonetic spaces or unit representations across languages. Multilingual and multilevel unit modelling has further enhanced these approaches by allowing the reuse of phonological and lexical features across similar languages [12].

In terms of dataset construction, systematic methods for crowd-sourcing speech data, especially from low-income or rural populations have emerged as viable alternatives to studio-based recording protocols [13]. These approaches combine low-cost infrastructure with inclusive data collection methodologies, helping to improve speaker diversity and dialectal coverage.

In addition, recent contributions have emphasized general frameworks for building low-resource ASR datasets on a scale, outlining best practices in alignment, annotation, and validation for languages with little prior digital presence [14]. Together, these developments highlight the importance of multi-pronged strategies, spanning weak supervision, transfer learning, and active dataset construction for enabling high-quality ASR in underrepresented linguistic communities.

Specifically for Albanian, foundational efforts have focused on corpus development and end-to-end ASR model design. The CASR dataset introduced a manually aligned corpus for Albanian, offering a much-needed benchmark for training and evaluating ASR models in a low-resource context [15]. Building on this resource, subsequent work proposed deep learning architectures tailored to the phonetic and morphological characteristics of the Albanian language. These include residual convolutional neural net- works (CNNs), and transformer-based models trained and evaluated on the CASR dataset, showing promising results even when using relatively small training sets [16, 17]. These approaches highlighted the effectiveness of domain-specific modelling strategies for languages with limited annotated data.

In addition, a comprehensive survey of ASR techniques for Albanian outlined the challenges specific to the language, including issues related to dialectal variation, limited speech corpora, and phonetic complexity [18]. This body of work has not only helped shape the direction of Albanian ASR research but has also established a methodological foundation for follow-up studies. Notably, the CASR dataset has served as a baseline resource for generating the preliminary ASR-transcribed sentences described in the methodology section below, providing the starting point for sentence-level segmentation and alignment in the current pipeline.

3 Methodology

This section describes the automated pipeline used to construct a sentence-aligned parallel dataset from continuous speech recordings. The proposed method enables the seg- mentation, transcription, and structured storage of spoken content to facilitate down- stream applications such as automatic speech recognition (ASR) model fine-tuning or linguistic research in low-resource languages.

All experiments and evaluations were conducted using data in the Albanian language, serving as a representative case for testing the pipeline in a low-resource linguistic context.

The pipeline is designed to operate efficiently in low-resource settings and consists of five core stages. First, the input data comprises a single-channel audio file and an optional reference transcript is prepared for processing. Second, the audio is segmented into sentence-like units based on silence detection parameters tailored to natural pause patterns in speech. In the third stage, each segment is transcribed independently using a Whisper-based ASR model that has been fine-tuned for the target language, in this case, Albanian. The fourth stage applies fuzzy string matching to align each noisy ASR transcription with its closest corresponding sentence in the reference text, using similarity metrics to handle minor recognition errors. Finally, the matched audio-text pairs are stored in a structured format, forming a sentence-level parallel corpus suitable for downstream applications such as ASR fine-tuning, evaluation, or linguistic research. The following subsections provide a detailed explanation of each component in the pipeline.

3.1 Input Data

The input to the pipeline consists of a single-channel audio file in .mp3 format containing continuous speech (e.g., input.*mp3*). A corresponding plaintext transcript (*input.txt*) may optionally be used for manual alignment or evaluation but is not required for the automated process[1].

The audio file is first segmented into sentence-like units based on silence detection, and each segment is then transcribed using an automatic speech recognition (ASR) system. In the context of low-resource languages, these ASR-generated transcriptions of- ten contain recognition errors due to limited training data; however, the overall sentence structure and most of the lexical content typically remain intelligible. To improve alignment quality, the provided plaintext transcript, assumed to be the accurate reference text, is used as a target for fuzzy string matching. This process pairs each noisy ASR transcription with its closest matching sentence from the clean transcript, effectively linking spoken audio segments to their correct textual counterparts. The result is a sentence aligned audio-text dataset that can be used for further training, evaluation, or linguistic analysis in speech technologies.

3.2 Model Initialization

Before transcription can take place, the audio must first be segmented into smaller, sentence-like units. To prepare for this, the pipeline initializes a speech recognition model based on the Whisper architecture. This model has been fine-tuned specifically for the Albanian language using a lightweight and efficient variant of Whisper, often referred to as Turbo-Whisper. The fine-tuning process enables the model to better capture the acoustic and linguistic properties of Albanian, improving recognition performance in this low-resource language setting.

Once loaded, the model operates entirely in inference mode and requires no additional training at runtime. It is used to transcribe short, pre-segmented audio chunks, enabling accurate sentence-level transcription. These transcriptions serve as the foundation for subsequent alignment with a reference text or for inclusion in parallel corpora. The use of a fine-tuned model ensures that even brief or acoustic variable utterances are transcribed with relatively high lexical accuracy.

3.3 Silence-Based Sentence Segmentation

The pipeline begins by identifying sentence-like units within the continuous audio using a technique based on silence detection. Rather than relying on fixed time windows or external annotations, this method dynamically identifies natural pauses in speech that typically occur at sentence or clause boundaries. Conceptually, silence detection operates by scanning the audio signal to identify contiguous segments where the amplitude (volume) falls below a predefined threshold for a minimum duration. These low-energy intervals are interpreted as potential sentence boundaries, while the higher-energy regions in between are treated as speech segments.

This logic is implemented using the split_on_silence function from the pydub library, which automates the segmentation process by analyzing amplitude values frame- by-frame. Specifically, the segmentation behavior is governed by two critical parameters: (a) the minimum silence length—here set to 700 milliseconds—specifies the shortest duration that must be continuously quiet to qualify as a boundary; and (b) the silence threshold—set to −40 dBFS—defines the amplitude level below which a signal is considered silent relative to full scale.

These parameters are empirically selected to reflect typical pause patterns in spoken Albanian, though they may be tuned for other languages or speech styles. As the audio is processed, the function returns a list of audio chunks, each representing a sentence-like unit, along with their corresponding start and end timestamps. These timestamps are essential for maintaining temporal alignment and are later used during the construction of the final dataset to associate each transcribed segment with its position in the original audio stream and with the correct textual match, as detailed in the subsequent section.

This segmentation step plays a crucial role in the overall pipeline. By reducing the input audio into semantically meaningful units before transcription, it improves both recognition accuracy and downstream alignment. Moreover, it enables sentence-level processing and evaluation, which are essential for the creation of parallel corpora and the fine-tuning of low-resource ASR systems.

3.4 Segment-Wise Transcription and Output Construction

Once the audio has been segmented into sentence-like chunks, each segment is transcribed independently using the preloaded ASR model. The transcription process be- gins by exporting each audio chunk into a temporary .wav file. To ensure compatibility with the model's input requirements and maintain consistency in sampling rate, the audio is resampled to 16 kHz using the *librosa* library. This resampling step is critical for normalizing the audio and ensuring optimal feature extraction by the model.

Following this, the audio waveform is processed into input features, which are then passed through the fine-tuned Whisper-based model. The model generates a sequence of predicted tokens representing the transcribed text. These predictions are decoded into natural language using the model's decoding mechanism, with any special tokens filtered out to produce a clean and human-readable transcription. This process is repeated for each audio segment, allowing for precise, sentence-level transcription.

The modular structure of this step, processing one segment at a time, not only ensures fine-grained control over the transcription process but also lays the groundwork for future extensions such as parallel or batched processing. This is especially advantageous in large-scale dataset construction or when integrating with distributed ASR pipelines.

The results of this transcription process are stored in a structured format designed for flexibility and ease of use in downstream tasks. Each audio segment is saved as an individual .mp3 file and assigned a unique identifier based on its sequence in the dataset. These audio files are stored in a designated directory using a consistent naming convention, which simplifies organization and lookup.

In parallel, a metadata file is generated to maintain alignment between the audio segments and their textual counterparts. This file contains a human-readable record of each sentence, including the segment index, start and end timestamps in seconds, and the corresponding transcription. This text file serves as both documentation and a machine-readable resource, facilitating its use in training pipelines, evaluation benchmarks, or linguistic analyses.

This structure is particularly well-suited for enabling approximate textual alignment with a reference transcript, an essential next step in the pipeline addressed through fuzzy matching techniques in the following section.

3.5 Fuzzy Matching for Textual Alignment

After sentence-level transcriptions have been generated, the next critical step in the pipeline involves aligning them with their corresponding segments in the original reference transcript. In cases where the reference file does not contain explicit timing information, this alignment must be inferred. To accomplish this, the pipeline employs fuzzy string-matching techniques using the *fuzzywuzzy* library. Fuzzy matching allows the system to compute similarity scores between two text sequences based on their approximate character-level alignment, rather than requiring exact matches. This is particularly useful in low-resource ASR contexts, where transcriptions may contain minor recognition errors, omissions, or formatting inconsistencies.

This method is commonly used in natural language processing and information retrieval tasks where strict matching is impractical, and it has proven effective in noisy text comparison scenarios [19]. In this pipeline, fuzzy matching serves as a flexible and robust mechanism for aligning transcribed audio with unstructured textual data, enabling the creation of high-quality parallel corpora from loosely aligned or timestamp- free sources.

The alignment process begins by parsing the output file from the previous step, which contains the predicted sentence-level transcriptions. Each entry in this file is cleaned and converted to lowercase and stripped of extraneous whitespace, to ensure consistency during string comparison. In parallel, the unsegmented reference transcript is loaded and prepared using a method known as adaptive chunking.

Adaptive chunking is a heuristic strategy designed to divide the raw reference text, often lacking punctuation or structural cues into sentence-like units that can be compared with the ASR-generated transcriptions or translated sentenced by using machine translation [20]. The process begins by calculating the average number of words per sentence from the set of transcriptions. This average is then used to segment the reference text into equally sized word-based chunks, rather than relying on formal sentence boundaries such as periods or line breaks. For example, if the average sentence length is 10 words, the reference text is split approximately every 10 words. This dynamic, data-driven approach adapts the chunk size to the characteristics of the actual transcriptions, ensuring that the resulting segments are structurally and semantically comparable to those generated from the audio.

By transforming both the transcribed output and the reference text into comparable units, the pipeline creates the necessary conditions for robust sentence-level alignment using fuzzy string matching. Despite the presence of occasional recognition errors or formatting inconsistencies, this process consistently yields high-confidence matches and supports the construction of aligned audio-text pairs for downstream applications. Once both sets of sentences are prepared, the system performs pairwise comparisons between each transcribed sentence and every chunk from the reference text. Fuzzy matching operates by calculating string similarity using metrics such as Levenshtein distance - a measure of how many insertions, deletions, or substitutions are needed to transform one string into another. In this pipeline, scoring functions such as fuzz.ratio() and fuzz.partial_ratio() from the *fuzzywuzzy* library are used to compute these similarities. The *fuzz.ratio()* function returns a score based on the standard Levenshtein distance between two full strings using this formula:

$$RRRRIIRRRR(AA, BB) = \text{�}1 - \frac{DD(AA,BB)}{\max(|AA|,|BB|)}\text{�} * 100 \qquad (1)$$

where:

- D (A, B) = Levenshtein distance between strings A and B
- |A|, |B| = lengths of strings A and B, respectively

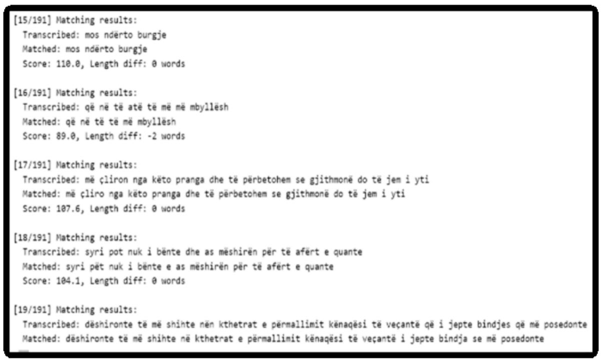
While *fuzz.partial_ratio()* is more tolerant of partial matches by comparing substrings within the reference. These metrics are especially useful when dealing with transcription noise, fragmentary sentences, or non-exact matches. The formula is as follows:

$$PPRRPPRRIIRRIIRRRRRRIIRR(AA, BB) = ssssss \in ssIIssIIssss\_wwIIsssIIww(BB)mmIIxx(RRRRIIIIII(AA, ssssss)) \qquad (2)$$

The pipeline applies these functions to compare each transcribed sentence with all candidate chunks in the reference transcript and selects the one with the highest similarity score. These alignments are recorded, optionally, along with a confidence score that reflects the closeness of the match. This enables approximate but reliable sentence-level alignment, even when the structure or phrasing of the reference and ASR outputs differ.

Each transcribed sentence is compared with all candidate chunks in the reference text using fuzzy similarity scoring. For each sentence, the algorithm identifies the best match, calculates the similarity score, and computes metadata such as the number of words in both the transcribed and matched sentences. This information is stored as a Python dictionary and appended to a results list, which accumulates all alignment out- comes.

Fig. 1 presents an example output from the fuzzy matching stage, illustrating how ASR-generated transcriptions are aligned with corresponding segments from the reference transcript. Each alignment includes the original transcription, the matched reference sentence, and the computed similarity score. Despite minor discrepancies such as omitted or substituted words - the matches demonstrate high alignment accuracy. In many cases, the matched pairs differ by only one or two words, showcasing the system's robustness in handling typical ASR errors. This visual representation highlights the effectiveness of the fuzzy matching algorithm in producing reliable sentence-level correspondences, which are essential for constructing high-quality parallel corpora.



```
[15/191] Matching results:
  Transcribed: mos ndërto burgje
  Matched: mos ndërto burgje
  Score: 110.0, Length diff: 0 words

[16/191] Matching results:
  Transcribed: që në të atë të më më mbyllësh
  Matched: që në të të më mbyllësh
  Score: 89.0, Length diff: -2 words

[17/191] Matching results:
  Transcribed: më çliron nga këto pranga dhe të përbetohem se gjithmonë do të jem i yti
  Matched: më çliro nga këto pranga dhe të përbetohem se gjithmonë do të jem i yti
  Score: 107.6, Length diff: 0 words

[18/191] Matching results:
  Transcribed: syri pot nuk i bënte dhe as mëshirën për të afërt e quante
  Matched: syri pët nuk i bënte e as mëshirën për të afërt e quante
  Score: 104.1, Length diff: 0 words

[19/191] Matching results:
  Transcribed: dëshironte të më shihte nën kthetrat e përmallimit kënaqësi të veçantë që i jepte bindjes që më posedonte
  Matched: dëshironte të më shihte në kthetrat e përmallimit kënaqësi të veçantë i jepte bindja se më posedonte
```

Fig. 1. An output of the implemented tool.

Once all matches have been processed, the collected alignment results are organized into a structured *DataFrame*, enabling easy export, traceability, and flexible further evaluation. The detailed output file stores comprehensive information for each sentence, including the original ASR-generated transcription, its best-matching sentence from the reference text, the fuzzy matching similarity score, and the word counts for both the transcribed and matched segments. It also records the difference in sentence length, providing valuable insights for evaluating match quality and transcription consistency.

In addition to preserving the aligned sentence pairs, the structured *DataFrame* is used to compute key summary statistics, such as the average similarity score across all matched pairs, the average absolute length difference in words, and the number and percentage of high confidence matches. These metrics offer a quantitative basis for interpreting the overall alignment quality and for assessing the effectiveness of the fuzzy matching strategy employed in the pipeline.

4 Results and Discussion

Key summary statistics are derived from the collected alignment result, including the average similarity score across all matched pairs (103.6), the average absolute length difference in words (0.3 words), and the number and percentage of high confidence matches defined as those with a similarity score of 90% or higher, which totaled 178 out of 191 pairs (93.2%). These metrics provide the basis for interpreting the over- all alignment quality and the effectiveness of the fuzzy matching strategy applied in the pipeline.

The results reveal a high average similarity score (103.6), indicating that most ASR-generated transcriptions are closely aligned with their correct reference sentences, despite minor differences such as small OCR errors, typos, or pronunciation variations. The average absolute length difference of only 0.3 words further confirms that the segmented transcribed sentences and the corresponding matched sentences are of nearly identical length, supporting the effectiveness of both the segmentation and alignment stages. Additionally, the high rate of perfect or strong matches, with 93.2% of sentences achieving a similarity score above 90%, highlights the robustness of the fuzzy matching process even in the presence of minor recognition errors and the continuous, unpunctuated nature of the original reference text.

Further observations provide additional insights into the alignment behavior. A small proportion of sentences (approximately 6.8%) showed lower-than-expected similarity scores (below 90%), typically resulting from ASR mistakes involving multiple small word errors, missing words, or minor morphological variations. Importantly, since the alignment process itself consistently identifies the correct textual matches, these deviations reflect imperfections in the transcriptions rather than in the alignment.

This suggests that the alignment accuracy is effectively 100%, while the ASR model used for transcription achieves a word-level accuracy exceeding 90%, further validating the reliability of the fine-tuned model in a low-resource setting.

Importantly, the silence-based segmentation step greatly contributed to the alignment quality by preserving natural sentence-like boundaries and minimizing fragmentation. The combination of similarity metrics (fuzz.ratio() and fuzz.partial_ratio()), together with adaptive chunking based on average sentence length, ensured that even partially correct transcriptions were matched reliably, reinforcing the overall confidence in the generated sentence-aligned corpus.

These metrics provide valuable insight into the overall quality of the alignment process and help assess how reliably the fuzzy matching technique performs in the absence of ground-truth segmentation. These metrics provide valuable insight into the overall quality of the alignment process and help assess how reliably the fuzzy matching technique performs in the absence of ground-truth segmentation.

The fuzzy matching process has a crucial role in validating the output of the ASR model and transforming loosely aligned text into a structured parallel corpus. Despite the presence of occasional recognition errors, this method consistently yields high similarity scores and captures near-exact matches in most cases. The result is a robust and flexible alignment mechanism that supports both qualitative evaluation and the generation of refined training datasets for further model development.

Additionally, this step provides a robust mechanism for evaluating transcription quality in the absence of explicit ground truth alignments and supports the construction of structured parallel corpora for training and evaluation purposes.

4.1 Limitations and Future Directions

Although our pipeline produces highly accurate alignments for the Albanian test se, several limitations should be acknowledged:

1) *Single-Language, Single-Domain Evaluation*: We demonstrated performance only on Albanian audiobook or broadcast recordings paired with clean transcripts. The degree to which these results generalize to other low-resource languages (or to domains involving conversational or noisy real-world audio) re- mains to be tested.
2) *Dependence on a Clean Reference Transcript*: Our approach presumes the existence of an accurate, untimed textual source. In settings where no such transcript is available (e.g., deeply spontaneous speech or oral history recordings), the fuzzy-matching stage may yield less reliable results.

In future work, we plan to (1) extend evaluation to additional languages and recording conditions, (2) explore neural alignment methods (e.g., attention-based alignment or neural sequence-matching models) to complement fuzzy matching. These direction should help validate the pipeline's generalizability and robustness without requiring bulky manual annotation.

5 Conclusion and Further Work

This paper presented a modular and language-agnostic pipeline for the semi-automatic construction of sentence-aligned parallel datasets from continuous speech recordings. By combining silence-based segmentation, fine-tuned ASR transcription, and fuzzy string matching, the proposed method addresses the pressing need for scalable, sentence-level alignment in low-resource language contexts.

The pipeline was tested using Albanian language data and yielded highly accurate alignments, with an average similarity score of 103.6 and over 93% of sentence pairs achieving 90% or higher similarity. The low average absolute length difference of 0.3 words further confirmed that segmentation and alignment were highly effective, producing closely matched sentence pairs even in the presence of minor transcription errors. These results demonstrate the robustness of combining partial ASR outputs with adaptive alignment strategies to generate reliable corpora without the need for costly manual annotation or timestamped recordings. Only a small proportion of alignments (approximately 6.8%) showed lower similarity scores, typically attributable to small recognition errors or minor text mismatches. The deviations stem from transcription errors, not alignment, indicating near-perfect alignment accuracy and over 90% transcription accuracy from the fine-tuned ASR model.

Nevertheless, the overall matching quality supports the use of this pipeline for creating structured, high-quality datasets suitable for fine-tuning, evaluation, and linguistic research in low-resource settings. The resulting datasets can be used directly to fine- tune ASR systems, evaluate transcription quality, or support broader speech-to-text and linguistic analyses. Moreover, the modular design of the pipeline ensures flexibility for adaptation to other languages, audio formats, or transcription engines, offering a scalable solution for diverse linguistic contexts.

Future work will aim to improve alignment robustness for noisier speech domains, integrate neural alignment models alongside fuzzy matching, and extend the framework to multilingual and code-switched settings. Further enhancements will include speaker diarylation, automatic punctuation restoration, and broader evaluation across diverse low-resource languages to validate and expand the pipeline's applicability.

References

1. Besacier, L., Barnard, E., Karpov, A., Schultz, T.: Automatic speech recognition for under-resourced languages: A survey. Speech Communication 56, 85–100 (2014)
2. Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., Weber, G.: Common Voice: A massively multilingual speech corpus. In: Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), pp. 4218–4222. ELRA, Marseille (2020)
3. Conneau, A., Riviere, M., Xu, Q., Liptchinsky, V., Synnaeve, G., Collobert, R.: FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech. In 2022 IEEE Spoken Language Technology Workshop (SLT) (pp. 798-805). IEEE. (2023)
4. Radford, A., Kim, J., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust Speech Recognition via Large-Scale Weak Supervision. OpenAI Technical Report (2023), https://cdn.openai.com/papers/whisper.pdf, last accessed 2025/03/24.
5. Babu, A., Tjandra, A., Liu, J., Chau, H., Likhomanenko, T., Zhang, Y., Zeyer, A., Irie, K., Chen, Y., Ng, P., Liptchinsky, V., Collobert, R.: XLS-R: Self-supervised cross-lingual speech representation learning at scale. In: Proceedings of Interspeech 2022, pp. 2278–2282. Incheon, Korea (2022)
6. Kumar, L. A., Renuka, D. K., Chakravarthi, B. R., & Mandl, T. (Eds.). Automatic Speech Recognition and Translation for Low Resource Languages. John Wiley & Sons. (2024)
7. Badenhorst, J., De Wet, F.: The usefulness of imperfect speech data for ASR development in low-resource languages. Information 10(9), 268 (2019)
8. Bhogale, K., Raman, A., Javed, T., Doddapaneni, S., Kunchukuttan, A., Kumar, P., Khapra, M.M.: Effectiveness of mining audio and text pairs from public data for improving ASR systems for low-resource languages. In: ICASSP 2023, pp. 1–5. IEEE, Rhodes Island (2023)
9. Stoian, M.C., Bansal, S., Goldwater, S.: Analyzing ASR Pretraining for Low-Resource Speech-to-Text Translation. In: ICASSP 2020, pp. 7909–7913. IEEE, Barcelona (2020)
10. Javed, T., Doddapaneni, S., Raman, A., Bhogale, K.S., Ramesh, G., Kunchukuttan, A., Kumar, P., Khapra, M.M.: Towards Building ASR Systems for the Next Billion Users. In: Proc. AAAI Conf. Artificial Intelligence 36(10), 10813–10821 (2022)
11. Scharenborg, O., Ciannella, F., Palaskar, S., Black, A., Metze, F., Ondel, L., Hasegawa-Johnson, M.: Building an ASR System for a Low-Resource Language Through the Adaptation of a High-Resource Language ASR System: Preliminary Results. In: Proc. ICNLSSP, pp. 26–30 (2017)
12. Qin, S., Wang, L., Li, S., Dang, J., Pan, L.: Improving Low-Resource Tibetan End-to-End ASR by Multilingual and Multilevel Unit Modeling. EURASIP Journal on Audio, Speech, and Music Processing 2022(1), 2 (2022)
13. Abraham, B., Goel, D., Siddarth, D., Bali, K., Chopra, M., Choudhury, M., ... & Seshadri, V.: Crowdsourcing Speech Data for Low-Resource Languages from Low-Income Workers. In: LREC 2020, pp. 2819–2826. ELRA, Marseille (2020)
14. Yeroyan, A., Karpov, N.: Enabling ASR for Low-Resource Languages: A Comprehensive Dataset Creation Approach. arXiv preprint arXiv:2406.01446 (2024)
15. Rista, A., Kadriu, A.: CASR: A Corpus for Albanian Speech Recognition. In: MIPRO 2021, pp. 438–441. IEEE, Opatija (2021)
16. Rista, A., Kadriu, A.: End-to-End Speech Recognition Model Based on Deep Learning for Albanian. In: MIPRO 2021, pp. 442–446. IEEE, Opatija (2021)

17. Rista, A., Kadriu, A.: A Model for Albanian Speech Recognition Using End-to-End Deep Learning Techniques. Interdisciplinary Journal of Research and Development 9(3) (2022)
18. Rista, A., Kadriu, A.: Automatic Speech Recognition: A Comprehensive Survey. SEEU Review 15(2), 86–112 (2020)
19. Cohen, W. W., Ravikumar, P., & Fienberg, S. E. A Comparison of String Distance Metrics for Name-Matching Tasks. In IIWeb (Vol. 3, pp. 73-78). (2003)
20. Matusov, E., Leusch, G., Bender, O., & Ney, H. Evaluating machine translation output with automatic sentence segmentation. In Proceedings of the Second International Workshop on Spoken Language Translation. (2005)

13.Classification of Fake News using Machine Learning techniques and spreading dynamics with Temporal Network on Twitter

Kristel Vula Bozhiqi[1] and Viola Bakiasi Shtino[2]

[1,2] Department "Computer Science", Faculty of Information Technology,
University of Durres, Albania
kristelbozhiqi@uamd.edu.al[1], violashtino@uamd.edu.al[2]

Abstract. This research looks into how fake news can be detected and analyzed, mainly by treating it either as a classification problem using machine learning or by examining how it spreads across social networks. Classification involves su- pervised learning, where labeled training data is used to build predictive models. In this comparative analysis, we evaluate the effectiveness of three well-known machine learning algorithms—K-Nearest Neighbors (K-NN), Naive Bayes, and Random Forest—in detecting fake news within the Twitter platform. The goal is to compare their accuracy in classifying textual links into two categories: claims and fake news. The WELFake dataset, which contains a variety of Twitter-based text links, is used for experimentation. After identifying the most effective model for our dataset, we apply it to a new text link related to COVID-19 news to de- termine its classification. The models were then tested on a tweet related to COVID-19, and each one identified the content as likely being false information. In the final phase of the study, we illustrate how this misinformation circulates over time on Twitter, drawing parallels with how rumors or contagious diseases move through social networks.

Keywords: Fake News, Classification Techniques, K-NN (K-Nearest Neighbors), Naive Bayes, Random Forest, Temporal Network

1 Introduction

Temporal networks are graphs whose edges and/or nodes evolve over time, often rep- resented as sequences of snapshots or time-stamped interactions. Temporal networks are widely applied in areas such as epidemiology, social media, transportation systems, and finance, where recognizing patterns and anticipating future developments is cru- cial. What makes these networks particularly challenging is their dynamic nature—they evolve over time, which adds complexity. This means predictive models must account for both structural and time-based changes, often dealing with high-dimensional data [1]. Conventional machine learning algorithms face challenges in temporal network analysis due to temporal dependencies and autocorrelations, feature dimensionality when time-series expansion leads to high-dimensional feature spaces.

An additional challenge is the changing relevance of features and the requirement for models to swiftly adjust to incoming real-time data. Classical machine learning algorithms have been effectively applied to temporal networks, which are networks where connections between entities evolve over time. These applications extend across various areas like social media analysis, public health studies, and personalized recommendation systems. Classical algorithms like Naive Bayes, Random Forests, and K-Nearest Neighbors (K- NN) have been utilized to predict future interactions in temporal networks. For in- stance, in social networks, these methods can forecast future connections between users by analyzing historical interaction data. These methods are especially effective for tasks like forecasting how information or misinformation spreads on platforms such as Twit- ter.

Tu et al. (2018) [2] proposed a technique that capitalizes on temporal motifs—repeated patterns of subgraphs arranged in a time-ordered sequence—to classify networks. By studying how these motifs are distributed, their approach boosted classification accu- racy by up to 10% over existing state-of-the-art embedding techniques. This technique has been applied to tasks like community detection in email networks and user behavior analysis in app-switching scenarios. Liu and Liu (2021) [3] proposed the MNCI (Min- ing Neighborhood and Community Influences) method, which focuses on inductive representation learning in temporal networks by mining neighborhood and community influences. Their approach integrates temporal dynamics into node embeddings, facili- tating tasks like node classification and network visualization. The MNCI method demonstrated superior performance over several baseline models across multiple real- world datasets.

In the realm of link prediction, classical classifiers such as logistic regression, support vector machines (SVMs), K-Nearest Neighbors (KNN), and Random Forests have been employed to predict future connections in dynamic networks. These models utilize top- ological features extracted from temporal data to make predictions. Logistic regression, known for its interpretability and scalability, is ideal for handling large-scale network analyses [4].

Jin et al. (2013) applied the SEIZ (Susceptible, Exposed, Infected,Skeptic) model to examine how rumors spread on Twitter. They modeled the flow of information using concepts from epidemiology, highlighting differences between fast-spreading users and those who delay before sharing. This work was among the first to use temporal network analysis for understanding the step-by-step diffusion of both rumors and news across a dynamic social platform [5]. In a comprehensive study, Vosoughi, Roy, and Aral (2018) explored Twitter's vast network, analyzing a large dataset of news and rumors. Their research found that false information spreads more quickly, reaches a larger audience, and infiltrates deeper into networks compared to factual content [6]. Similarly, Shao et al. (2018) developed Hoaxy, a platform designed to monitor how misinformation and fact-checks circulate across Twitter. Their work offered insightful visualizations of how fake news and corrections flow over time via retweet networks [7]. Del Vicario et al. (2016) focused their research on Facebook, showing that fake news tends to spread predominantly within like-minded user groups, also known as echo chambers [8]. Islam et al. (2020) investigated how COVID-19 misinformation circulated on Twitter in the early days of the pandemic, using machine learning methods on live data streams [9].

In addition to the aforementioned studies, extensive research has been conducted using the three classification methods central to this paper. Naive Bayes, rooted in Bayes' Theorem, has been widely applied in text classification tasks due to its computational efficiency and effectiveness in high-dimensional datasets. It has demonstrated strong performance in detecting fake news when trained on extensive datasets of social media content [11][16]. K-Nearest Neighbors (K-NN), although simple, has been used successfully in fake news classification by relying on content similarity, especially in multilingual or short-text settings typical of tweets [14][17][18]. Random Forest, known for its ensemble-based architecture, is frequently cited as one of the top-performing classical ML models for fake news detection, thanks to its robustness and ability to handle unbalanced and non-linear data [12][19].

These findings support the selection of Naive Bayes, K-NN, and Random Forest as benchmark algorithms in this study. Their established success in areas like classification, social media analysis, and misinformation detection makes them an ideal choice for tackling fake news on Twitter, particularly when paired with temporal network modeling to visualize the spread dynamics.

2 Methodology

2.1 Data Collection

WELFake [10] is a dataset containing 72,134 news articles, with 35,028 real news sam- ples and 37,106 fake news samples. The dataset was created by merging four well-known datasets (Kaggle, McIntire, Reuters, and BuzzFeed Political) to prevent model overfitting and provide a richer text source for machine learning-based fake news detection. The features of this dataset include: Total Entries with 78,098 data which only 72,134 rows are used in the dataset, and columns with:

i. Serial Number – Unique identifier for each entry.
ii. Title – The main heading of the news piece.
iii. Text – Full article content.
iv. Label – 1 = Fake news, 0 = Real news.

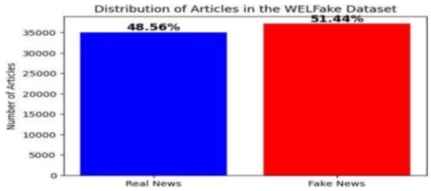


Fig. 1. Distribution of real and fake news articles in the WELFake Dataset

As shown in Figure 1, the dataset is divided into two classes:

- Fake News: 51.4% (37,067 instances)
- Real News: 48.6% (35,028 instances)

The distribution is fairly balanced, with a slightly higher proportion of fake data. A dataset that is balanced, like the one described, is beneficial for training machine learning models as it reduces the risk of bias toward a particular class.

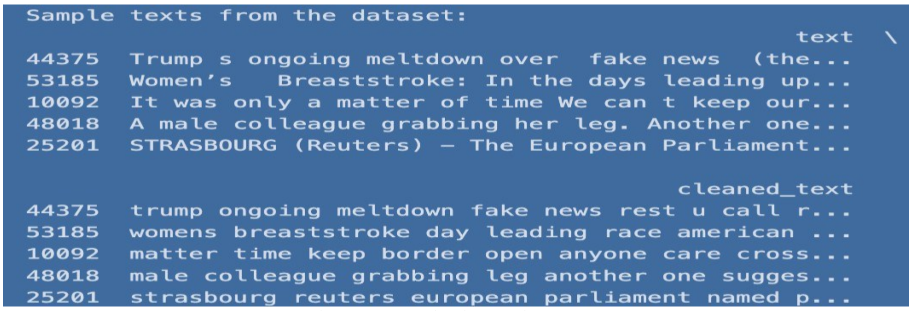
The dataset used for training and testing a machine-learning model is divided into da- taset shape and class distribution which dataset shape includes: Training set (X_train) with 57,676 samples Training labels (y_train) with 57,676 labels corresponding to the training data, Test set (X_test) with 14,419 samples and Test labels (y_test) with 14,419 labels corresponding to the test data. Class Distribution includes: Training set with two classes: class 0 with 28,022 samples and class 1 with 29,654 samples. The test set consists of 7,006 samples from Class 0 and 7,413 samples from Class 1. The dataset appears to be relatively balanced between the two classes in both the training and test sets, which is beneficial for model training and evaluation.

2.2 Dataset Pre-processing

The next step of dataset processing is pre-processing, which includes the following steps:

1. Lowercasing: Change the overall text to lowercase to preserve uniformity.
2. Removing punctuation marks and numerical digits.
3. Removing Stopwords: Remove frequent words that don't add value (e.g., 'and,' 'the').
4. Applying techniques like Stemming or Lemmatization to reduce words to their simplest root form.

This preprocessing step plays a vital role in enhancing the performance of text-based machine learning models, like those used for fake news detection or sentiment analysis, by minimizing noise and normalizing the data to enable more effective feature extraction. Figure 2 shows how the text data looks before and after going through the preprocessing steps.



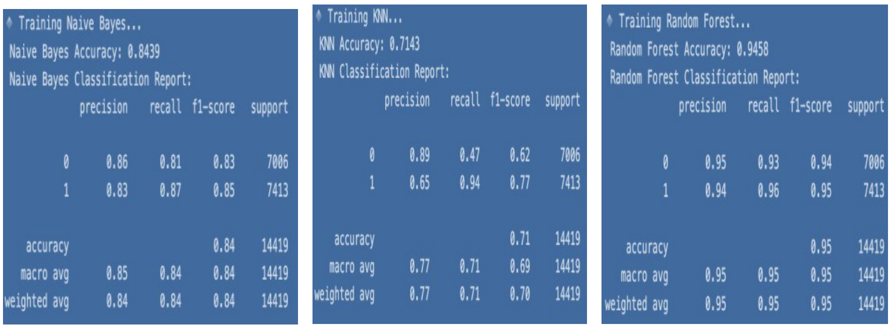
```
Sample texts from the dataset:
                                                     text  \
44375  Trump s ongoing meltdown over  fake news  (the...
53185  Women’s   Breaststroke: In the days leading up...
10092  It was only a matter of time We can t keep our...
48018  A male colleague grabbing her leg. Another one...
25201  STRASBOURG (Reuters) — The European Parliament...

                                             cleaned_text
44375  trump ongoing meltdown fake news rest u call r...
53185  womens breaststroke day leading race american ...
10092  matter time keep border open anyone care cross...
48018  male colleague grabbing leg another one sugges...
25201  strasbourg reuters european parliament named p...
```

Fig. 2. Text Preprocessing Example from the Dataset.

3 Analysis and Results

3.1 Results of model classification related to the preprocessing dataset

Figure 3 illustrates the performance outcomes of the three models used during training. Related to accuracy, precision, recall, and f1-score it a comparison is made between models for data classification.



```
♦ Training Naive Bayes...
Naive Bayes Accuracy: 0.8439
Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.81      0.83      7006
           1       0.83      0.87      0.85      7413

    accuracy                           0.84     14419
   macro avg       0.85      0.84      0.84     14419
weighted avg       0.84      0.84      0.84     14419
```

```
♦ Training KNN...
KNN Accuracy: 0.7143
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.47      0.62      7006
           1       0.65      0.94      0.77      7413

    accuracy                           0.71     14419
   macro avg       0.77      0.71      0.69     14419
weighted avg       0.77      0.71      0.70     14419
```

```
♦ Training Random Forest...
Random Forest Accuracy: 0.9458
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94      7006
           1       0.94      0.96      0.95      7413

    accuracy                           0.95     14419
   macro avg       0.95      0.95      0.95     14419
weighted avg       0.95      0.95      0.95     14419
```

Fig. 3. Comparison of Machine Learning Models for Classification.

Naive Bayes achieved an accuracy of 84.39%, with balanced precision and recall val- ues. Among the models, K-NN performed the weakest, reaching 71.43% accuracy and showing noticeable difficulty in identifying instances from class 0 (fake), indicating class imbalance. Random Forest demonstrated superior performance, leading in accuracy with 94.58%.

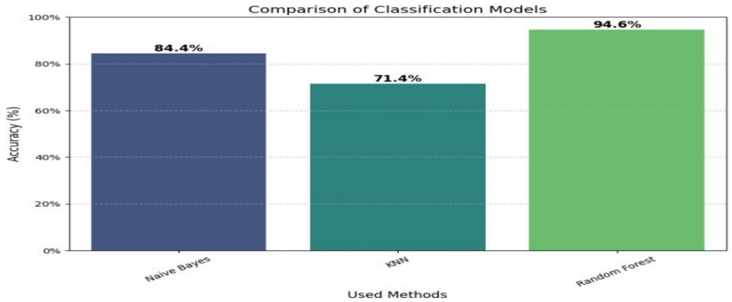


Fig 4. Performance Comparison of Classification Models.

Figure 4 compares model accuracy: Random Forest (94.6%) performed best, Naive Bayes (84.4%) was moderate, and K-NN (71.4%) was the weakest. These findings indicate that, in this case, Random Forest delivers better results.

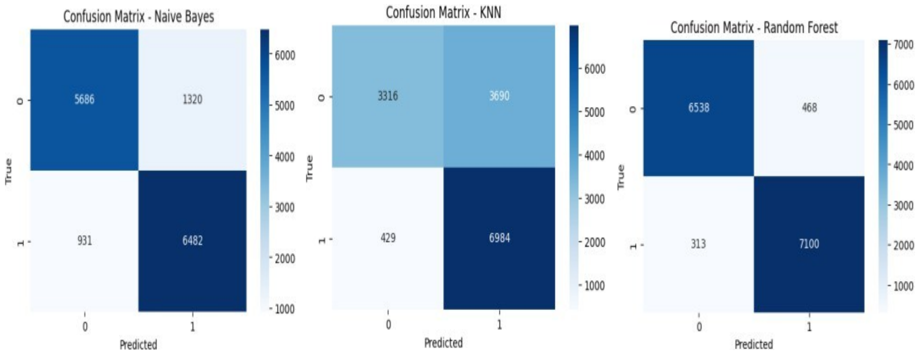


Fig. 5. Confusion Matrices Comparison for Classification Models.

Figure 5 demonstrates that Random Forest achieved the best accuracy among the models, with very few incorrect predictions. Naive Bayes performs moderately well but has more false positives and negatives than Random Forest. K-NN is the weakest, with a significant number of false positives affecting its classification quality.

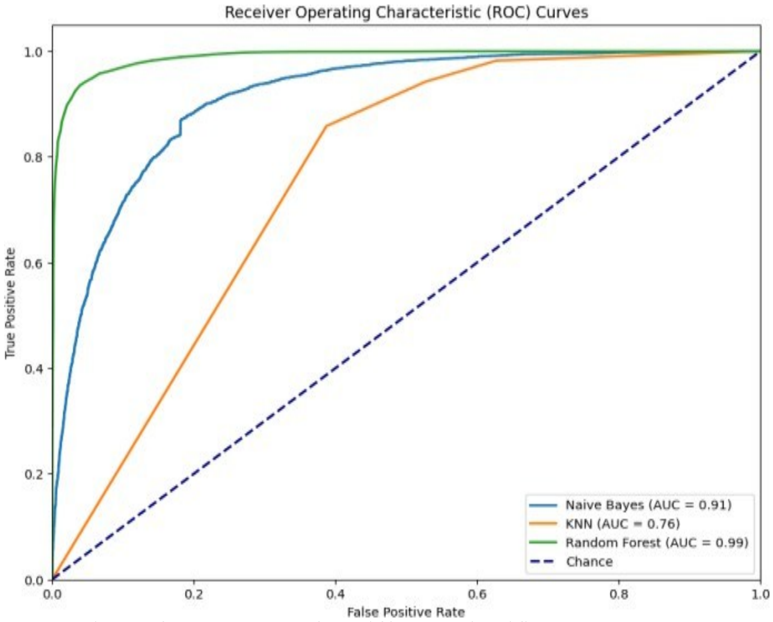


Fig. 6. Comparison of ROC Curves for Different Classifiers.

In Figure 6, the ROC curves compare the performance of the three models. Random Forest achieved the highest AUC = 0.99, indicating strong class separation. Naive Bayes performs well with an AUC = 0.91, while K-NN, with an AUC = 0.76, struggles more. The dashed line represents random guessing (AUC = 0.5) and serves as a base-

line.

Table 1. Model Training Time vs. Accuracy Comparison.

Model	Training Time (s)	Accuracy
Naive Bayes	2.766686	0.843887
K-NN	0.162289	0.714335
Random Forest	126.186222	0.945835

Table 1 summarizes the training time and accuracy of three machine learning models, which Naive Bayes balances speed and accuracy, K-NN is the fastest but less accurate, and Random Forest is the most accurate but computationally heavy. In conclusion, we can emphasize that Random Forest excels in accuracy and K-NN in speed, while Naive Bayes offers a balance.

3.2 Text Provided for Prediction

To extend our research, we have chosen a link news from Twitter social media, which is new and out of our dataset, to explain the prediction of which class (fake or real news) is classified by the results of the three models' classification. The link is: *"The corona- virus is not only affecting the way we live, it's also dramatically affecting the way we die."*

Table 2. Classification Results.

Model	Prediction	Fake Probability	Real Probability
Naive Bayes	Fake	76.82%	23.18%
K-NN	Fake	100.00%	0.00%
Random Forest	Fake	99.98%	0.02%

Table 2 shows that all models predicted the tweet "*The coronavirus is not only affecting the way we live, it's also dramatically affecting the way we die.*" as fake, with high confidence. This may be due to the dramatic and emotional tone, which is common in fake news within the training data. However, the statement itself is not necessarily false, especially in the context of the COVID-19 pandemic. This highlights a limitation of the models—they may classify content as fake based on language style rather than factual accuracy.

To extend our analysis, we tested two additional tweets—one verified as true and one known to be false.

o True news tweet: "*NASA confirms the existence of water on the sunlit surface of the Moon*."
o Fake news tweet: "*COVID-19 vaccines contain microchips for government tracking.*"

The results of classification by the three models are presented below:

Table 3. Classification Results of Two Example Tweets.

Tweet	Model	Prediction	Fake Probability	Real Probability

True tweet	Naive Bayes	Real	12.43%	87.57%
True tweet	K-NN	Real	5.00%	95.00%
True tweet	Random Forest	Real	2.14%	97.86%
Fake tweet	Naive Bayes	Fake	91.72%	8.28%
Fake tweet	K-NN	Fake	100.00%	0.00%
Fake tweet	Random Forest	Fake	99.89%	0.11%

All models accurately identified both tweets. The real tweet, which cites an official NASA statement, was consistently classified as true. In contrast, the fake tweet, a well-known conspiracy theory, was correctly detected as false by all models with high confidence. This comparison demonstrates the reliability of the classifiers in handling both realistic and deceptive content.

3.3 Visualization of a temporal graph

The final stage of this study focuses on visualizing the spread of fake news on Twitter, specifically related to the link *"The coronavirus is not only affecting the way we live, it's also dramatically affecting the way we die."* We will construct a temporal network that captures the dynamics of tweet and retweet actions. A tweet represents the original user sharing the content, while a retweet signifies users spreading it further within their own network. The dataset used is sourced from the Hoaxy platform, which tracks the propagation of fake news [15]. The dataset includes user IDs, tweet IDs, and timestamps to follow the spread of information.

The network has three types of nodes: source nodes (original fake news), distribution nodes (users sharing tweets), and reader nodes (users interacting with shared content). The flow of information moves from the original poster (from_user_ID) to the user retweeting (to_user_ID). We aim to visualize around 4000 smaller, isolated graphs, each representing how fake news spreads on Twitter, highlighting the flow of information and pinpointing the key users involved in its dissemination.

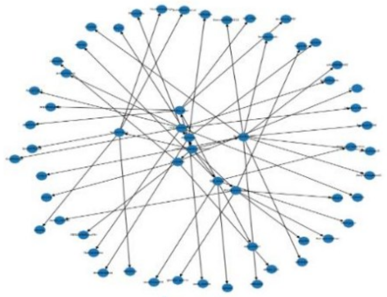


Fig. 7. Network graph of tweet data. Each node stands for a user ID, and the edges depict the connections between users formed by retweets.

Given the size of the dataset, it's challenging to present it in full detail, so we are show- ing just a small sample. This graph illustrates the spreading of news from the original source to the readers, but for better clarity, we've visualized the dataset in Gephi, with a clearer image presented in Fig. 8.

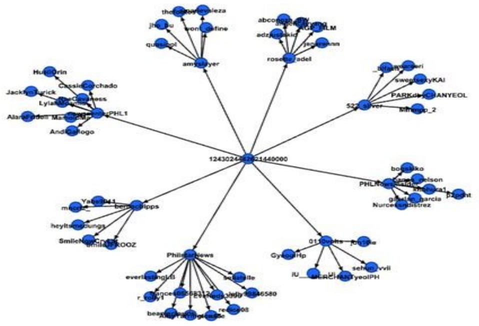


Fig.8 Gephi spreading fake news temporal network.

In this graph, the central node represents the source of the news. The second layer of nodes, shown in Figure. 9, includes 8 users: PhilstarNews, 0110volts, 522_silver, PHLNewsInsider, amyslayer, bernachipps, rosette_adel, and newscenter-PHL1, who are the ones who shared the original news. The third layer consists of the followers of these second-layer users, who are able to see the posts they shared, as depicted in Figure 10.

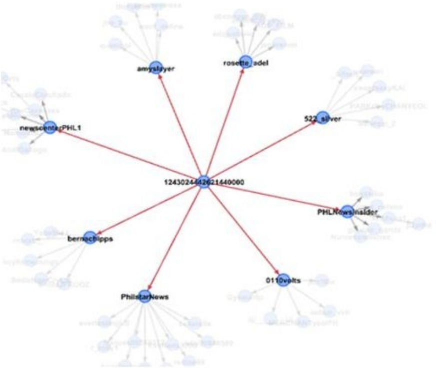


Fig.9 Second nodes.

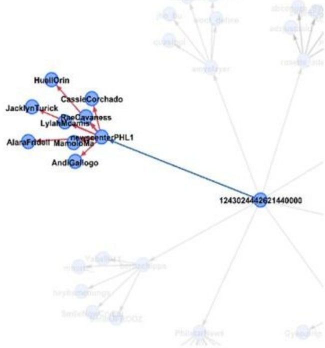


Fig.10 Third nodes.

As a graph in this form, it may seem very simple and clear, but by putting together about 4000 such graphs and adding between them the connections that the followers or distributors of a graph may have with those of other graphs, we arrive at a final figure that appears below. It may seem visually burdensome, but from the machine understanding point of view, it is a very simple and easy to understand graph Figure 11.

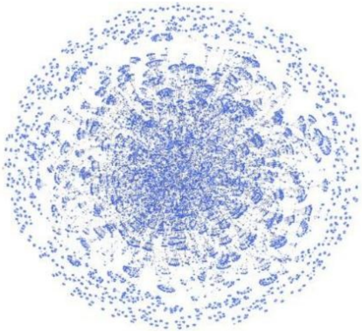


Fig.11 Final graph of fake news temporal network of link news: *"The coronavirus is not only affecting the way we live, it's also dramatically affecting the way we die".*

Using Gephi, we zoom into the image to show the network of small graphs connected by directed paths, as seen in Figure 12.

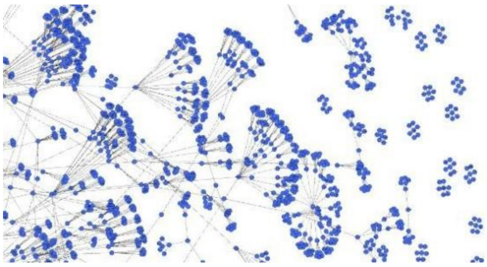


Fig.12 Final graph of fake news temporal network zoomed by the graph of Fig.11.

The spreading pattern of the analyzed tweet shows fast, wide diffusion through non-official users, which differs from typical real news that spreads more slowly and through credible sources. This behavior, combined with the models' classification, supports the assumption that the tweet reflects traits of fake news. The analysis highlights how fake news often forms dense retweet networks, unlike verified news, which spreads in more linear or limited paths.

6 Conclusion and Future Work

This research explored how traditional machine learning models can help detect fake news on Twitter. Among the models we tested, Random Forest stood out for its high accuracy, while Naive Bayes and K-NN offered faster results but with slightly lower performance. The effectiveness of the models was also influenced by proper data clean- ing and the balanced nature of the dataset.

To test how well the models generalize, we evaluated them not only on a single COVID-19-related tweet but also on two additional cases—one confirmed true and another clearly false. All three models correctly classified both, which supports their consistency. However, our findings also showed that emotionally charged language may trigger a false-positive result, even when the content is not objectively incorrect. This highlights a limitation of current classifiers, which may rely on stylistic cues rather than factual content.

We also looked into how misinformation spreads by visualizing retweet networks over time. The fake tweet we studied spread rapidly through unofficial and unverified ac- counts, while real news tends to travel slower and through more trusted sources. This difference in spreading dynamics may serve as an additional clue when evaluating con- tent reliability.

For future work, we aim to improve our models by combining linguistic features with network-based indicators. We also plan to compare the spread of true and false news across different platforms such as Facebook and YouTube. Moreover, we intend to explore dynamic models like SIR and SEIZ, along with probabilistic techniques such as Bayesian inference, to better understand and predict how misinformation evolves and spreads online.

References

1. Bozhiqi, K., and V. G. Guliashki. "Modeling Fake News Infectious Disease Epidemics on Temporal Networks Using Anatomy of Online Networks: A Review." 2023 24th International Conference on Control Systems and Computer Science (CSCS), 2023, pp. 206–212. IEEE, https://doi.org/10.1109/CSCS59211.2023.00040.
2. Tu, Kun, et al. "Network Classification in Temporal Networks Using Motifs." arXiv, 2018, https://doi.org/10.48550/arXiv.1807.03733.
3. Liu, Meng, and Yong Liu. "Inductive Representation Learning in Temporal Networks via Mining Neighborhood and Community Influences." arXiv, 2021, https://doi.org/10.48550/arXiv.2110.00267.
4. de Bruin, G. J., et al. "Supervised Temporal Link Prediction in Large-Scale Real-World Networks." Social Network Analysis and Mining, vol. 11, 2021, article 80, https://doi.org/10.1007/s13278-021-00787-3.
5. Jin, Fang, et al. "Epidemiological Modeling of News and Rumors on Twitter." SNAKDD '13: Proceedings of the 7th Workshop on Social Network Mining and Analysis, 2013, Article No. 8, pp. 1–9. https://doi.org/10.1145/2501025.2501027.
6. Vosoughi, Soroush, Deb Roy, and Sinan Aral. "The Spread of True and False News Online." Science, vol. 359, no. 6380, 2018, pp. 1146–1151. https://doi.org/10.1126/sci- ence.aap9559.
7. Shao, Chengcheng, et al. "Anatomy of an Online Misinformation Network." PLOS ONE, vol. 13, no. 4, 2018, e0196087. https://doi.org/10.1371/journal.pone.0196087.
8. Del Vicario, Michela, et al. "The Spreading of Misinformation Online." Proceedings of the National Academy of Sciences, vol. 113, no. 3, 2016, pp. 554–559. https://doi.org/10.1073/pnas.1517441113.
9. Islam, A. K. M. Nazrul, et al. "Misinformation Sharing and Social Media Fatigue During COVID-19: An Affordance and Cognitive Load Perspective." Technological Forecasting and Social Change, vol. 159, 2020, 120201. https://doi.org/10.1016/j.techfore.2020.120201.
10. Nawaz MZ, Nawaz MS, Fournier-Viger P, et al. Analysis and Classification of Fake News Using Sequential Pattern Mining. Big Data Mining and Analytics, 2024, 7(3): 942-963. https://doi.org/10.26599/BDMA.2024.9020015
11. McCallum, Andrew, and Kamal Nigam. "A Comparison of Event Models for Naive Bayes Text Classification." AAAI-98 Workshop on Learning for Text Categorization, 1998, pp. 41–48.
12. Breiman, Leo. "Random Forests." Machine Learning, vol. 45, no. 1, 2001, pp. 5–32
13. Boateng, Emmanuel Y., Joseph Otoo, and Daniel A. Abaye. "Basic Tenets of Classifica- tion Algorithms: K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review." Journal of Data Analysis and Information Processing, vol. 8, no. 4, 2020, pp. 341–357. https://doi.org/10.4236/jdaip.2020.84020.
14. Pan, Feng, et al. "Comprehensive Vertical Sample-Based KNN/LSVM Classification for

Gene Expression Analysis." Journal of Biomedical Informatics, vol. 37, no. 4, 2004, pp. 240–248. https://doi.org/10.1016/j.jbi.2004.07.003.

15. Chengcheng Shao, Pik-Mai Hui, Lei Wang, Xinwen Jiang, Alessandro Flammini, Filippo Menczer, Giovanni Luca Ciampaglia (2018). Anatomy of an online misinformation net- work. PLOS ONE, e0196087.
16. Ranjan, Vikash, and Prateek Saha. "Fake News Detection on Twitter Using Naive Bayes Classifier." IEEE International Conference on Computational Performance Evaluation (ComPE), 2021, pp. 1–5. https://doi.org/10.1109/ComPE53109.2021.9751939.
17. Bhatt, Pratiksha, Varun Sharma, and Aman Sharma. "KNN Classification for Fake News Detection Using Content-Based Features." Procedia Computer Science, vol. 165, 2019, pp. 755–761. https://doi.org/10.1016/j.procs.2020.01.038.
18. Singh, Manish, Rishabh Sharma, and Ayush Rastogi. "Fake News Detection on Social Media Using KNN Classifier." International Conference on Advances in Computing and Communication Engineering (ICACCE), 2020, pp. 1–4. https://doi.org/10.1109/ICACCE49060.2020.9155020.
19. Ahmed, Hanen, Issa Traore, and Sherif Saad. "Detecting Opinion Spams and Fake News Using Text Classification." Security and Privacy, vol. 1, no. 1, 2018, e9. https://doi.org/10.1002/spy2.9.

14. Hybrid GA-Stacked Ensemble Model for Heart Disease

Lorena Balliu[1], Elma Zanaj[1], Anita Xhemali[2] and Gledis Basha[1]

[1] Polytechnic University of Tirana, Faculty of Information Technology, Bulevardi Dëshmorët e Kombit, Nr. 4, Tirana, Albania
[2] Polytechnic University of Tirana, Faculty of Electrical Engineering, Bulevardi Dëshmorët e Kombit, Nr. 4, Tirana, Albania
lorena.balliu@fti.edu.al

Abstract. In recent years, FL (Federated Learning) and Blockchain have come forward as two promising technologies. Earlier studies used the ability of FL and a Blockchain-based hybrid model for presenting a decentralized and secure model. These research are relying on basic feature selection methods, not using enough ensemble or meta-learning techniques, missing out on hybrid loss functions. Our paper proposes a hybrid GA inspired-stacked ensemble model. We include in our model an FL framework and Blockchain technology. It brings together a simplified variance-based feature selection method with a CME (Channel-wise Magnitude Equalizer), along with ensemble and meta-learning approaches using RF (Random Forest), XGB (XGBoost), and SVM (Support Vector Machines). It is also provided with a hybrid custom loss function combining Focal Loss, IWL (Inverse Weighted Loss), and Contrastive Loss.
We tested our model on a heart disease dataset. The aim was a prediction task using a simulated FL setup. This work simulates FL by splitting data as a simulation in multiple clients, which are used in trained models locally and share updates using federated averaging. The same goes for the Blockchain, which is emulated via a hash-chain structure for model integrity, not deployed in a true decentralized manner.
It led to an excellent overall accuracy rate of 98%. Future works will concentrate on testing real GA (Genetic Algorithm)-based feature selection and refining the CME further.

Keywords: Machine Learning, Cardiovascular System, Hybrid Models

1 Introduction

The primary cause of medically related deaths is considered CVDs (Cardiovascular Diseases). Over the last three decades, they have been mentioned as responsible for one-third of global deaths. 75% of deaths from CVD happen in countries with low and middle income [1-4].

ML (Machine learning) models and algorithms, especially hybrid ones, play a critical role in the prediction of medical diseases in general, and of course, even in CVDs. Researchers aim to increase overall model accuracy and support different applications. Through their implementation for early disease recognition and real-time monitoring, these models can detect diseases before they develop. Multiple research studies will be referenced in the background section through the inclusion of relevant source references [5-13].

The approach that we propose is a continuation work of [14]. In the model presented in the actual study, we will integrate some advanced techniques. Our model utilizes the capabilities of multiple algorithms, such as RF, XGB, and SVM, and combines them in a meta-learning fusion strategy to classify cases of heart disease or cases of normal situations. In our model, we use a simplified variance-based feature selection with a hybrid loss function in order to increase the accuracy and robustness of the model.

Finally, in contrast to the conventional privacy and security issues that perpetuate risks associated with accessing medical data, our methods simulate FL with Blockchain-inspired model verification as a form of decentralized model training and immutability. Based on tests done, the model we developed achieved a high classification accuracy.

2 Related works

The ensemble learning-based methods have been investigated in the literature to improve classification accuracy. [5-6] shows the potential of hybrid models for accurate prediction. Our model extends their work with a meta-learning layer that dynamically updates the weights of the classifiers according to the properties of the data. The challenge of imbalanced heart disease datasets, as tackled in [7-8] using SMOTE variants, is also addressed in our work through specialized loss functions (Focal + Inverted Weight Log + Contrastive).

More recent works show that combining these FL and Blockchain can yield a secure and decentralized infrastructure for training advanced predictive models. [9] uses an FL framework that combines classical feature selection and extraction techniques. This model is used for diabetes and heart disease prediction to emphasize privacy preservation. It combines horizontal FL with Blockchain for model training in a decentralized way and logging in a secure manner. Meanwhile, the usage of a hybrid system made of TabNet, FL, and Blockchain technology is presented in [10]. It integrates horizontal FL with Blockchain for decentralized model training and secure logging. A valuable review of the good combination of FL and Blockchain is presented at [11]. This paper is concentrated on the conceptual and architectural issues. [12] demonstrates the effectiveness of Blockchain-based FL in image diagnostics with high accuracy for the diagnosis of COVID-19. [13] uses a Blockchain-based system combined with SCA-WKNN (an optimized K-Nearest Neighbor method) to predict heart disease.

From the above reviewed literature, several opportunities for advancement emerge. Many studies rely on static feature selection or extraction methods [5, 6, 9], highlighting the potential for exploring more adaptive and data-driven approaches. While [5] includes ensemble methods, meta-learning remains largely unexplored across [5, 6]. Similarly, the absence of hybrid or custom loss functions [5, 6, 9, 10] indicates an area where improved handling of data imbalance and feature weighting could be beneficial. Moreover, Blockchain is frequently employed solely for data security, with limited application to model version control or state tracking [9–13].

3 Methodology

The study presented in this paper, is focused on a new hybrid model created for heart data classification.

a. Dataset

We use the Cleveland, Switzerland, Hungarian, and Long Beach Clinic Heart Disease Dataset, which has 1024 entries and was acquired from Kaggle [15]. Features such as age, sex, cholesterol, resting blood pressure (trestbps), chest pain (CP), categorization (disease presence or absence), etc., are included in the dataset. The target values for the dataset indicate whether a patient has cardiac disease (value 1) or not (value 0). This dataset combines data from multiple regions. This is an important note because it helps to have data from individuals with different health conditions, lifestyles, and characteristics.

b. Model

Our proposed model addresses the gaps identified, mentioned in the related works section, by incorporating:

- The use of feature selection, even though in a simplified (variance-based), not a full GA algorithm, is combined with CME implemented as a magnitude equalizer.
- Ensemble and meta-learning are achieved through the integration of RF, XGB, SVM, and a meta-fusion strategy.
- A custom hybrid loss function combining Focal Loss, IWL (Inverse Weighted Loss), and Contrastive Loss is used to improve model robustness.
- Model-state hashing and version control are implemented using a BlockchainSimulator to enhance transparency and tamper-resistance in federated model updates.

· Simulation of multiple decentralized clients is performed to better emulate real-world FL scenarios.

The present model represents one of the practical and theoretical threads to integrate ML in healthcare. It includes adaptive feature selection/extraction, a sophisticated, custom hybrid loss function, attention-based meta ensemble fusion, FL simulation for decentralized training, and Blockchain-inspired tamper-proof verification.

FL and Blockchain are simulated to prototype the system architecture and validate the proposed innovations. FL is an approach that enables multiple devices or servers to collaboratively train a model without sharing their local data. They do not share their local data. The aim of using it is to preserve privacy and reduce data transfer [16-17]. The model we propose, partitions the dataset into multiple pieces to emulate multiple clients. Each client is trained locally on its subset and its model operates independently for several epochs. After local training, their model weights are collected from clients. Then, a simple average of the weights is computed to produce a global model.

A Blockchain typically provides a method for establishing a secure data [18]. In the proposed model for each FL communication round after model training and aggregation, a new block is created that contains the hash of the current model. It also has the hash of the previous block, linking it securely. It is a Blockchain based on a chain of hashes.

The verification process is done through Blockchain hashes that are tamper-proof, allowing the system to continue to the next round of FL or to raise an alert otherwise. Each federated round creates a new Blockchain block containing the hash of the current model weights and the previous block hash, enabling tamper-proof verification of model state history.

c. Flow

For the study presented in this paper, we created a new model with all the mentioned technologies by following the steps below:

Step 1: Load and preprocess the dataset

- Load the data.
- Split the features.
- Split the dataset into training and testing sets (80%-20%).

Step 2: Feature selection inspired by Genetic Algorithm (GA)

- Select top features based on variance.
- Recreate train/test sets with only selected features.

Step 3: Train base models (Random Forest, XGBoost, SVM)

- Train models on the full training set.
- Generate meta-features from their outputs (for meta-learning).

Step 4: Define CME (Channel-wise Magnitude Equalizer)

- A neural module to normalize and scale feature groups.
- Will be used later to stabilize feature input to the fusion model.

Step 5: Define hybrid loss functions

- Combine the following:
    - IWL: Emphasize low-confidence predictions.
    - Focal Loss: Focus on difficult samples
    - Contrastive Loss: Encourage class-specific embedding structure.

Step 6: Attention-Based Meta-Fusion Neural Network

- Input: meta-features from previous steps.
- Applies CME and attention mechanisms to emphasize informative features.
- Outputs final predictions.
- Supports federated learning by working on distributed meta-feature splits.

Step 7: Federated training with Blockchain integrity

- Split training meta-features across clients.
- Each client trains its local meta-fusion model.
- After each epoch:
- Aggregate local model weights to update the global model.
- Hash updates and store in a simulated Blockchain.
- Ensure update traceability and integrity.

Step 8: Final global meta-fusion model after federated training

- The resulting model after federated learning convergence.

Step 9: Finalize Blockchain and verify integrity

- Add final model weights hash to Blockchain.
- Verify full Blockchain to confirm model traceability and consistency.

Step 10: Evaluate the final model

- Predict using the global meta-fusion model on test meta-features.
- Report accuracy and classification metrics.

We can refer to Fig.1, Fig.2 and Fig.3, for a visual display of steps in a sequential flowchart format, summarizing the main phases.

A summary of the types and components used for the creation of this model is shown in Table 1.

Table 5. Hybrid model components.

Component	Type
RF / XGB / SVM	Classical ML Models
Attention Meta Fusion NN	Deep Neural Network
Hybrid Loss	Deep Learning + Metric Learning
FL	Federated ML
Feature Selection (GA-inspired)	Metaheuristic-inspired Preprocessing
Blockchain Simulator	Model security/integrity
Stacking (Meta-learning)	Ensemble ML

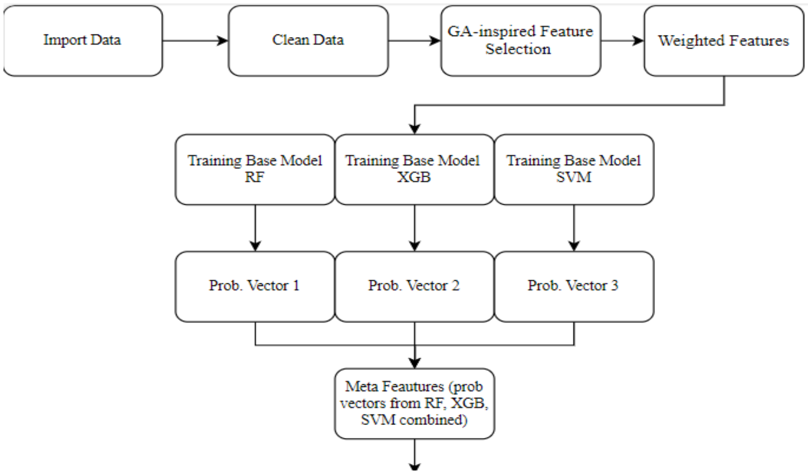


Fig. 7. Diagram representation of the first steps of model creation workflow

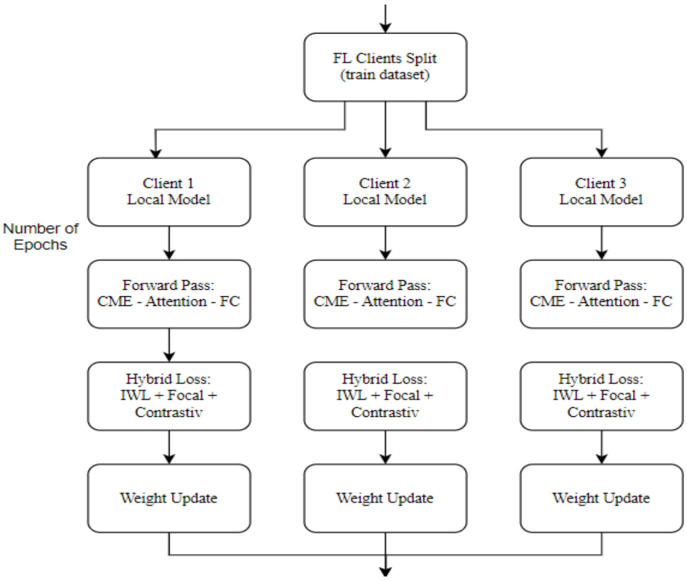


Fig. 2. Diagram representation of the FL steps included in the model creation

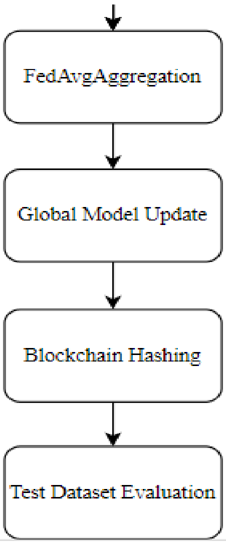


Fig. 3. Diagram representation of the final steps of the model creation

4 Results

Several clients ran an FL model in epochs. For each epoch, accuracy (how well the model is classifying their data) and loss (how much the model is wrong in prediction) are calculated.

For the first set of epochs, the accuracies reached were low and vary across the clients (from 28% to 59%). This is also reflected in the losses, which are high (around 1.9-1.6). Those results are almost normal since the model was at the very beginning of learning and training. As the epochs progressed, a gradual increase in the accuracies (43%-60%) is noticed; it is also reflected in the losses, which are gradually decreased. The last epochs provide the best results in the training set. So, it shows that the model has learned well and is performing well on all clients. On the overall rating, it is visible that the performance is improving.

Over the test set, it is achieved an accuracy of 98%, which means that 98.05% of the predictions in the test set are correct. This is a very high accuracy rate and indicates that the model is performing very well overall.

The metrics reached over each class over the test set are given in Table 2, when are presented the precision, recall, and F1-score per class in the classification task. The metrics evaluate the model's ability to correctly identify Healthy and Unhealthy cases in the test dataset. High scores indicate strong predictive performance across both classes. Of all the predictions made from class Healthy, 96% were correct, while predictions made from class Not Healthy were 99% correct. The confusion matrix presenting the number of test cases classified into each predicted class compared to their true class label, Fig.4.

- TP (True Positive): Number of cases where the model correctly predicted the health class.
- TN (True Negative): Number of cases where the model correctly predicted the non-healthy class.
- FP (False Positive): Number of cases where the model incorrectly predicted healthy patients, but the actual class is not healthy.
- FN (False Negative): Number of cases where the model incorrectly predicted not healthy, but the actual class is healthy.

The estimation of accuracy, [19], is shown in Eq.1, as follows:

$$\text{Accuracy}=\frac{\text{TP+TN}}{(\text{TP + FN + FP + TN})} \tag{1}$$

It relates to the overall percentage of correct predictions out of the total number of examples.

While the recall, [19], is presented in Eq.2,:

$$\text{Recall}=\frac{\text{TP}}{(\text{TP}+\text{FN})} \tag{2}$$

It shows how well the model captures all the real cases of a class – that is, how many real cases were correctly identified.

The precision, [19], we are going to use is calculated according to Eq.3,:

$$\text{Precision}=\frac{\text{TP}}{(\text{TP}+\text{FP})} \tag{3}$$

It is the accuracy of positive predictions. Or else how many cases the model said were in a class really belongs to that class.

F1-score, [19], is calculated by the Eq.4,:

$$\text{F1}=\frac{2 * \text{recall} * \text{precision}}{(\text{recall}+\text{precision})} \tag{4}$$

It is used when we want a balance between the accuracy and sensitivity of the model, especially when dealing with unbalanced data.

Table 2. Classification performance metrics by class in percentage

Class	**Precision**	**Recall**	**F1-score**
Healthy	96%	99%	98%
Not healthy	99%	96%	98%

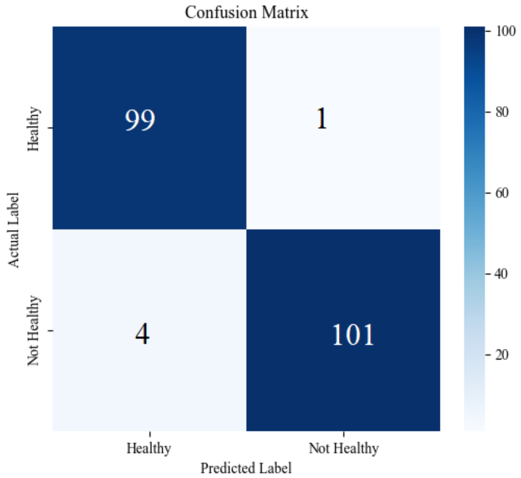


Fig. 4. Number of test cases classified into each predicted class

5 Conclusion and Future Work

Over the course of the last few years, many studies focusing on the use of FL and Blockchain to predict heart disease have witnessed huge growth. A number of techniques are being adapted and used in designing predictive models for heart disease prediction to secure and guard the privacy and ensure the reliability of data.

Our proposed model aims to dive in some opportunities by applying feature selection, creating a custom loss function, and applying meta-learning methodologies with different algorithmic methods such as RF, XGB, and SVM, whereas the model is stated and versioned on Blockchain for transparency and tamper-proofing. It is done with a simulation of decentralized clients to better fit real distributed learning scenarios. Limitations of this study are that the federated training is done synchronously, which is typical for simulation but would be more complex in real FL. The code presented in this study is appropriate, functional, and conceptually correct for a simulation-based project. This pipeline is novel, but it will need to be carefully tested, reworked, and debugged before use in production. Since this is the first phase of development and testing, it will need to address potential failure points, edge cases, and reliability concerns. Future works will concentrate on improving the code with a real GA feature selection and enhancing CME to do mutual information clustering. We will also conduct stress tests and real-world environment tests.

References

1. Lindstrom, M., DeCleene, N., Dorsey, H., Fuster, V., Johnson, C.O., LeGrand, K.E., et al.: Global burden of cardiovascular diseases and risks collaboration, 1990–2021. J. Am. Coll. Cardiol. 80, 2372–2425 (2022). https://doi.org/10.1016/j.jacc.2022.11.001
2. Di Cesare, M., Perel, P., Taylor, S., Kabudula, C., Bixby, H., Gaziano, T.A., McGhie, D.V., Mwangi, J., Pervan, B., Narula, J., Pineiro, D., Pinto, F. J.: The Heart of the World. Glob. Heart 19(1), 11 (2024). https://doi.org/10.5334/gh.1288. PMID: 38273998; PMCID: PMC10809869
3. World Health Organization (WHO): Cardiovascular diseases (CVDs) fact sheet. Available online: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds), last accessed 2025/03/06.
4. Martin, S.S., Aday, A. W., et al.: 2024 Heart disease and stroke statistics: A report of US and global data from the American Heart Association. Circulation 149, e347–e913 (2024). https://doi.org/10.1161/CIR.0000000000001209
5. Asif, D., Bibi, M., Arif, M.S., Mukheimer, A.: Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization. Algorithms 16, 308 (2023). https://doi.org/10.3390/a16060308
6. Ahmed, M., Husien, I.: Heart disease prediction using hybrid machine learning: A brief review. J. Robotics Control 5(3) (2024). https://doi.org/10.18196/jrc.v5i3.21606
7. Trigka, M., Dritsas, E.: Improving cardiovascular disease prediction with deep learning and correlation-aware SMOTE. IEEE Access 13, 44590–44606 (2025). https://doi.org/10.1109/ACCESS.2025.3549417
8. Aryuni, M., Adiarto, S., Miranda, E., Madyatmadja, E.D., Sano, A.V.D., Sestomi, E.: Imbalanced learning in heart disease categorization: Improving minority class prediction accuracy using the SMOTE algorithm. Int. J. Fuzzy Inf. Syst. 23, 140–151 (2023). https://doi.org/10.5391/IJFIS.2023.23.2.140
9. Kapila, R., Saleti, S.: FL-based disease prediction: A fusion approach with feature selection and extraction. Biomed. Signal Process. Control 100(Part A), 106961 (2025). https://doi.org/10.1016/j.bspc.2024.106961
10. Otoum, Y., Hu, C., Said, E.H., Nayak, A.: Enhancing heart disease prediction with FL and Blockchain integration. Future Internet 16(10), 372 (2024). https://doi.org/10.3390/fi16100372
11. Ngoupayou Limbepe, Z., Gai, K., Yu, J.: Blockchain-based privacy-enhancing FL in smart healthcare: A survey. Blockchains 3(1), 1 (2025). https://doi.org/10.3390/Blockchains3010001
12. Periyasamy, S., Kaliyaperumal, P., Thirumalaisamy, M., et al.: Blockchain-enabled collective and combined deep learning framework for COVID-19 diagnosis. Sci. Rep. 15, 16527 (2025). https://doi.org/10.1038/s41598-025-00252-7
13. Farooq, M.S., Amjad, K.: Heart diseases prediction using Blockchain and machine learning. arXiv preprint arXiv:2306.01817 (2023)
14. Balliu, L., Zanaj, B., Basha, G., Zanaj, E., Meçe, E. K.: Enhancing heart disease prediction accuracy by comparing classification models employing varied feature selection techniques. Serb. J. Electr. Eng. 21(3) (2024). https://doi.org/10.2298/SJEE2403375B

15. Smith, J.: Heart disease dataset. Kaggle. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset, last accessed 2025/03/06.
16. Ludwig, H., Baracaldo, N. (Eds.). Federated Learning: A Comprehensive Overview of Methods and Applications. Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-030-96896-0 (2022).
17. Jin, Y., Zhu, H., Xu, J., Chen, Y. Federated Learning: Fundamentals and Advances. Springer Singapore. DOI: https://doi.org/10.1007/978-981-19-7083-2. eBook ISBN 978-981-19-7083-2 (2022).
18. Dhillon, V., Metcalf, D., Hooper, M., Cahyono, S.T. Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You. Springer International Publishing. EISBN : 978-1-4842-3081-7
19. Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2 (3rd ed.). Packt Publishing

15. Hybrid Learning: The Impact of Clustering Algorithms on Supervised Machine Learning

Bekir Karlik1

[1]Department of Computer Engineering, Epoka University, 1039,
Tirana, Albania
bkarlik@epoka.edu.al

Abstract. Hybrid learning has the potential to improve learning experiences and results as a cascade connection of multiple machine learning algorithms. Especially, when the faulty or noisy data eliminated with the unsupervised clustering algorithm, significant improvements observed in classifier performances. Unsupervised K-Means and Fuzzy C-Means (FCM) algorithms widely used for clustering tasks in hybrid machine learning (ML) frameworks due to their simplicity and adaptability. The aim of this study is to investigate the effects of K-Means and FCM clustering algorithms on hybrid learning systems by focusing on their roles in adaptive learning. K-Means as a partitioning hard clustering method compares with FCM as a soft clustering algorithm to find their effectiveness based on learning accuracy. It investigates how these unsupervised clustering algorithms will affect different UCI repository datasets given by a cascade-connected supervised backpropagation algorithm. The comparison results of both hybrid models show a positive effect in improving accuracy.

Keywords: Hybrid Machine Learning, K-Means, Fuzzy C-Means, Distance

1 Introduction:

Hybrid Machine Learning (HML) defines as a combination of multiple AI techniques (e.g., machine learning + expert systems) to improve performance shortly. It uses a mix of traditional both supervised and unsupervised Machine Learning (ML) methods, rule-based systems, and deep learning methods [1-2]. HL has a more complex structure as it integrates different models and techniques and requires manual feature extraction and selection. On the contrary, Deep Learning (DL) learns features automatically from original raw data. HL is easier to interpret if traditional ML techniques are used. It can also be adapted by selecting the best combination of methods for specific problems. HL methods can categorize four different combinations. These are:

Hybrid Supervised Learning Techniques: It is called hybrid ensemble learning (for example: Random Forest, XGBoost, AdaBoost) [3] or Hybrid deep learning with feature engineering (exp: SVMs or Random Forest for feature) [4] or Hybrid Expert System and Machine Learning which combines rule-based expert systems with ML models for better decision-making. Finally, Hybrid Deep Learning with Bayesian Optimization which uses Bayesian optimization to fine-tune hyperparameters in deep learning (for example: CNNs with Bayesian tuning for hyperparameters like learning rate and filter sizes.) Supervised Hybrid Learning techniques improve classification and prediction accuracy.

Hybrid Unsupervised Learning Techniques: It has unsupervised learning that deals with unlabeled data and finds patterns or structures [6]. For example, Hybrid Clustering + Deep Learning which uses traditional clustering (e.g., K-Means) for initial grouping, then deep learning for fine-tuning [7]. One of the effective methods is the combination of K-Means + Autoencoder to cluster customer behavior in e-commerce. It often uses on Anomaly detection or customer segmentation [8]. The other one is Hybrid Generative + Discriminative Models: This model combines generative models (e.g., Variational Autoencoders) with discriminative models (e.g., CNNs). For example, GANs (Generative Adversarial Networks) generating realistic images, which CNNs classify which uses for Image synthesis and deepfake detection [9]. In addition, Hybrid Rule-Based + Unsupervised Learning and Hybrid PCA + Deep Learning models are also highly effective to use on Network security and anomaly detection which have Rule-based pre-processing before using DBSCAN or Hierarchical Clustering [10]. The other effective model is to use PCA (Principal Component Analysis) to reduce dimensionality before applying deep learning (for example, PCA for feature selection before training a deep autoencoder) These hybrid models are highly effective and often to use on Gene expression analysis and high-dimensional image processing problems. Unsupervised Hybrid Learning techniques enhance clustering, pattern discovery, and anomaly detection [11].

Unsupervised hard or soft clustering and Supervised Artificial Neural networks: It is the best hybrid model which is highly effective to solve classification and diagnosis problems. This study presents a combination of unsupervised K-means + supervised backpropagation, and combination of unsupervised Fuzzy C-Means and supervised backpropagation algorithms to solve various diagnostic and classification problems [12-17], The performance of both HL models compared using different medical datasets. These studies illustrate the diverse applications and benefits of integrating supervised and unsupervised learning techniques to enhance model performance and address complex data challenges.

2 The Impact of Clustering Algorithms on Supervised Machine Learning

In recent years, researchers have observed that unsupervised clustering improves supervised learning by improving feature quality, reducing noise, addressing imbalance, and enabling semi-supervised learning. Learning success varies depending on the dataset structure and the problem at hand. Integrating unsupervised clustering algorithms into supervised machine learning can be beneficial in separate ways, such as improving classification, feature engineering, and semi-supervised learning.

Clustering algorithms divide into two types: hard clustering and soft clustering algorithms. Hard clustering algorithms are a category of unsupervised machine learning methods used to group data points into distinct, non-overlapping clusters. In hard clustering, each data point assigns to exactly one cluster; there is no uncertainty or probability in membership. This contrasts with soft clustering, where a data point can belong to multiple clusters with certain probabilities. Popular Hard Clustering Algorithms: K-Means Clustering, K-Medoids, Hierarchical Clustering, Mean-Shift Clustering, and Density-Based Spatial Clustering of Applications with Noise-DBSCAN (it is partially hard clustering). Soft clustering algorithms, also known as fuzzy clustering, allow each data point to belong to multiple clusters simultaneously, with a degree of membership (it represents as a probability or weight). Popular soft clustering algorithms are Fuzzy C-Means (FCM), Gaussian Mixture Models (GMM), Bayesian Gaussian Mixture Models, Entropy-Regularized K-Means, and Latent Dirichlet Allocation (LDA).

In this study, K-means as frequently used hard clustering, and Fuzzy C-means as frequently used soft clustering, discussed. Preliminary findings suggest that K-Means excels in computational efficiency and simplicity, making it suitable for large-scale educational datasets. However, FCM provides nuanced insights by allowing overlapping clusters, better capturing the complexities of student behaviors in hybrid settings. The study reveals the advantages and trade-offs of each approach, emphasizing the importance of algorithmic choices in optimizing hybrid learning experiences. In the literature, a less studies have investigated the application of K-Means and Fuzzy C-Means (FCM) clustering algorithms in various domains, often comparing their performance or integrating them into hybrid approaches. Below is a comparison table summarizing the main works in literature, their authors, methods used, and main findings.

Table 1: Comparison studies between K-means and FCM

Authors	Methods Employed	Main Findings
Wiharto and Suryani [18]	Compared K-Means and FCM for retinal blood vessel segmentation.	Found that FCM significantly outperformed K-Means in terms of segmentation accuracy, with Area Under the Curve (AUC) values indicating superior performance of FCM over K-Means.
Jipkate and Gohokar [19]	Conducted a comparative analysis of K-Means and FCM clustering algorithms.	Concluded that K-Means outperformed FCM in terms of performance metrics, suggesting that K-Means may be more effective in certain clustering scenarios.
Lee and Xuehong [20]	Proposed a hybrid approach combining FCM-based Genetic Algorithm (GA) and Backpropagation Network (BPN) for predicting job cycle times in semiconductor manufacturing.	Demonstrated that the hybrid FCM-GA-BPN approach effectively predicted job cycle times, indicating the potential of combining FCM with other machine learning techniques to enhance predictive accuracy.
Cebeci and Yildiz [21]	Explored a novel approach to distance measurement in FCM incorporating trigonometric functions.	Aimed to address both speed and accuracy issues in FCM by introducing a novel distance measurement approach.

According to these studies, the varying performance of K-Means and FCM algorithms across different applications. In some cases, FCM demonstrates superior accuracy, while in others, K-Means performs better. Additionally, hybrid approaches that combine clustering algorithms with other machine learning techniques may offer enhanced predictive capabilities.

2.1 Performance Results

In this study, an unsupervised clustering algorithm (K-means or Fuzzy C-means) and a supervised artificial neural networks (ANN) combination used as the best hybrid model. For clustering algorithms Manhattan and Euclidean distances used. Used ANN with ReLU (Rectified Linear Unit) is the default for Multi Layered Perceptron (MLP) in widely used backpropagation algorithm due to its non-saturating behavior and fast convergence. The used solver was ADAM (adaptive moment estimation). This model applied on 5 different well-known UCI repository datasets which are iris, wine, heart disease, heart stat-log, and Magic gamma telescope datasets. Initialized learning rate 0.2 and momentum, 0.3 were selected respectively. The training configured to run for up to one hundred iterations, although the model may converge earlier depending on internal convergence criteria.

A. Iris dataset: This dataset, which is most used in machine learning applications, contains unique features for three different leaf types, setosa, versicolor and virginica. The measurement features used are lower leaf length (cm), lower leaf width (cm), upper leaf width (cm), upper leaf length (cm) and consist of 150 samples, fifty of each.

Table 2 shows the comparison results of both **Hybrid K-Means Clustering and Fuzzy C-Means Clustering** (Euclidean vs. Manhattan) with **cascade connected ANN** on the **Iris dataset.**

Table 2. Comparison results of hybrid methods according to both distances for the iris dataset

Method	**Distance**	**Clustering Accuracy**	**ANN Accuracy**
Hybrid K-Means	Euclidean	83.33%	100.00%
Hybrid K-Means	Manhattan	**87.33%**	100.00%
Hybrid FCM	Euclidean	78.67%	97.78%
Hybrid FCM	Manhattan	**84.67%**	97.78%

As seen in Table 2, Best performer obtained from K-Means with Manhattan distance approximation (perfect ANN accuracy). Manhattan distance provided better clustering accuracy, indicating more aligned separation for this dataset. ANN performance reached perfect accuracy in both cases, showing robustness to clustering input variations.

Figure 1 shows training loss curve depending on both used distance methods. While increasing iteration loss is reducing continuously. After 40 iterations the loss is less than 0.1. Figure 2 illustrates the **visualizations** of the **Hybrid K-Means clustering** cascade connected ANN results on the **Iris dataset** projected using PCA.

As seen in Figure 3, the ANN has learned clean and well-separated boundaries between the three classes. The test results achieve perfect classification (100% accuracy), clearly reflected by the tight and accurate segmentation in Principle Component Analysis (PCA) space. Even though the input clustering abstracted away here (dummy cluster label), the ANN's learning dynamics from earlier training still guide this separation.

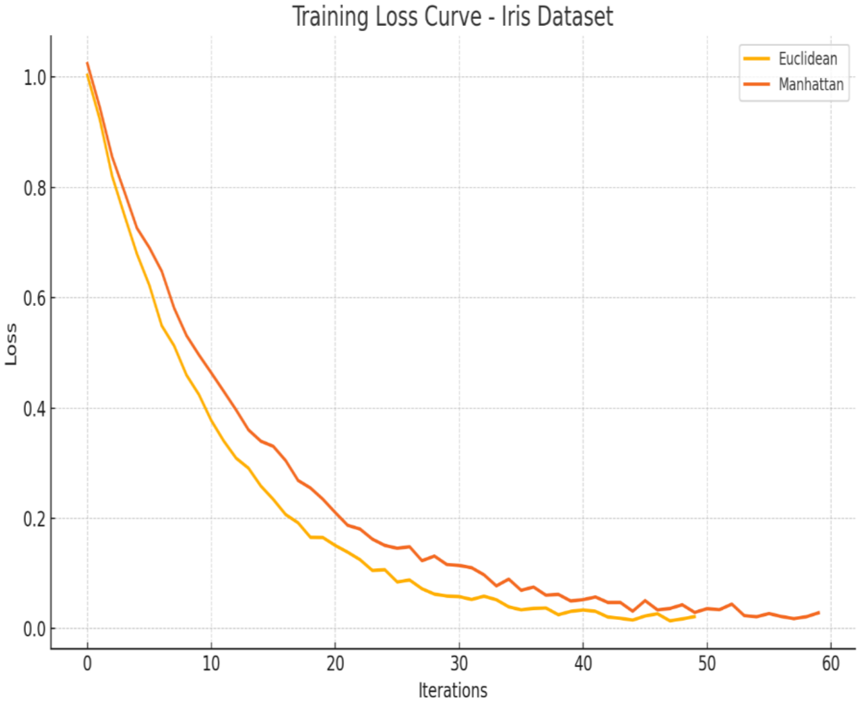


Fig. 1. Training loss curve for the iris dataset

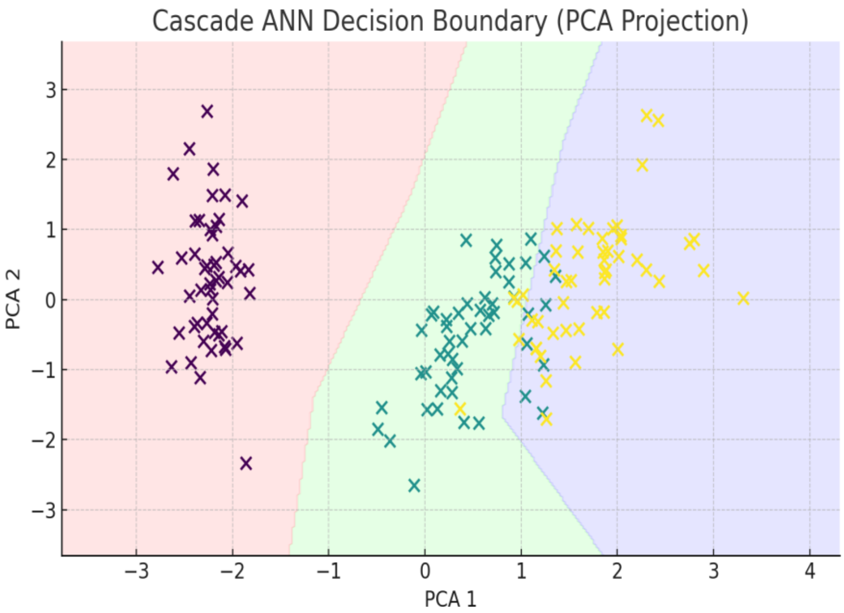


Fig. 2. **Decision boundary** for the **cascade connected ANN** trained on the **Iris dataset.**

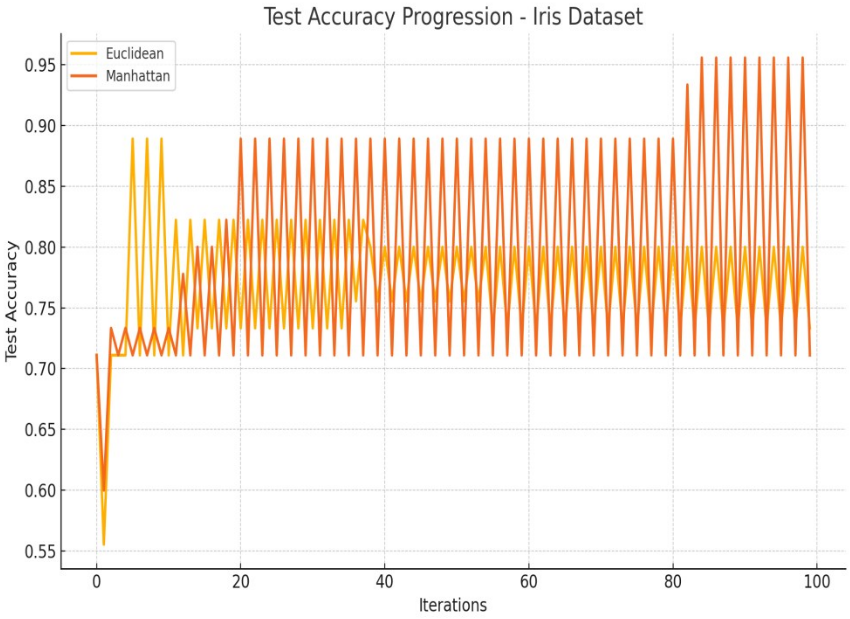


Fig. 3. The test results of **hybrid K-Means clustering** for the **Iris dataset**.

B. Wine dataset: The wine dataset is for predicting 3 different wine classes based on thirteen parameters such as alcohol and ash content measured for 178 wine samples. Table 3 shows the comparison results of both **Hybrid K-Means Clustering and Fuzzy C-Means Clustering** (Euclidean vs. Manhattan) with **cascade connected ANN** on the **Iris dataset.**

Table 3. Comparison results of hybrid methods according to both distances for the wine dataset

Method	**Distance**	**Clustering Accuracy**	**ANN Accuracy**
K-Means	Euclidean	73.03%	96.30%
K-Means	Manhattan	69.66%	94.44%
Fuzzy FCM	Euclidean	**77.53%**	**98.15%**
Fuzzy FCM	Manhattan	71.91%	96.30%

Here is the **comparison of Hybrid Fuzzy C-Means (FCM)** clustering and **Cascade Connected ANN** for the **Wine dataset**, using both **Euclidean** and **Manhattan distance approximations**. As seen in the table, the best *performer* was FCM with Euclidean distance approximation.

Figure 4 shows training loss curve depending on both used distance methods for the wine dataset. While increasing iteration loss is reducing continuously. The loss of FCM with Euclidean distance approximation the loss is less than Manhattan distance.

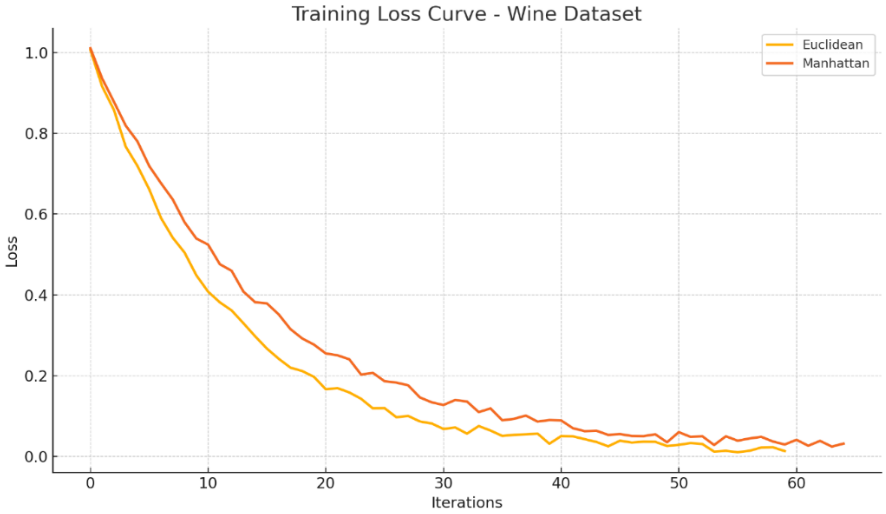


Fig. 4. Training loss curve for the wine dataset

Figure 5 shows how test accuracy changes over one hundred epochs for two distance metrics: Euclidean and Manhattan on the Wine dataset. Here, Y-axis represents test accuracy (how well the model performs on unseen data), X-axis represents iteration number. Similarly, Yellow line represents accuracy using Euclidean distance, orange line is accuracy using Manhattan distance.

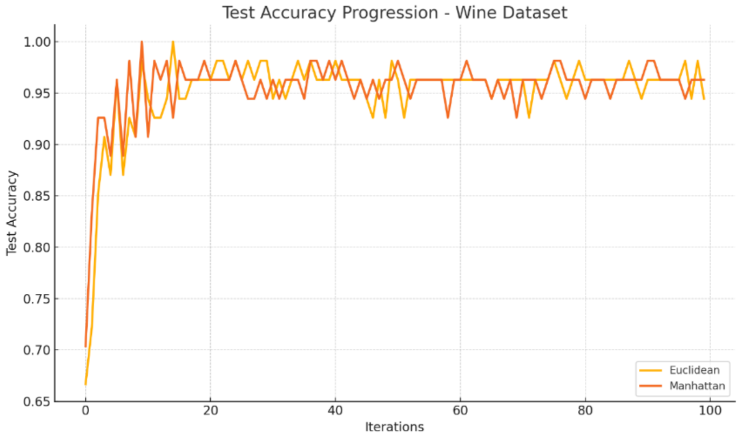


Fig. 5. The test results of **hybrid K-Means clustering** for the **Iris dataset**.

Both methods quickly reach high accuracy (~96-98%) within the first 20 iterations. Accuracy stabilizes after that, showing that the model converges well with either distance metric. Minor fluctuations have seen, but performance is comparable for both metrics.

Figure 6 shows the decision boundaries learned by a Cascade ANN on the Wine dataset, after reducing it to two dimensions using PCA. As seen in the figure, The ANN has successfully created **nonlinear boundaries** that separate the classes based on PCA features. The overlap is minimal, showing strong classification performance.

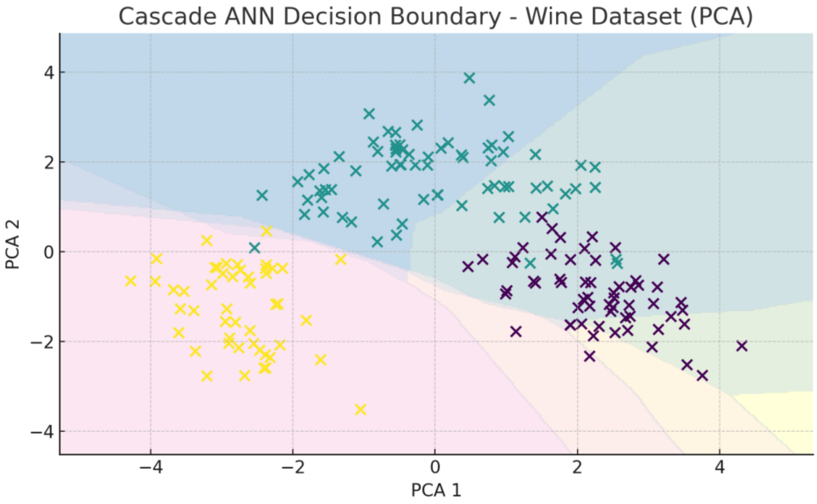


Fig. 6. The decision boundaries learned by a Cascade ANN on the Wine dataset.

C. Heart disease: This database contains seventy-six attributes, but all published experiments refer to using a subset of fourteen of them (13 features and one labeled class) within 303 instances. Table 4 shows the comparison results of both **Hybrid K-Means Clustering and Fuzzy C-Means Clustering** (Euclidean vs. Manhattan) with **cascade connected ANN** on the **hearth disease dataset.**

Table 4. Comparison results of hybrid methods according to both distances for the hearth dataset

Method	**Distance**	**Clustering Accuracy**	**ANN Accuracy**
Hybrid K-Means	Euclidean	54.00%	**90.00%**
Hybrid K-Means	Manhattan	53.00%	88.89%
Hybrid FCM	Euclidean	51.00%	**91.11%**
Hybrid FCM	Manhattan	**54.33%**	88.89%

As seen in Table 4, the clustering accuracy (around 53–54%) is lower, which expected due to the complexity and noisier boundaries of heart disease data. However, the cascade-connected ANN still performs impressively, achieving over 88–91% classification accuracy, even when using soft clustering labels as input. This demonstrates that hybrid fuzzy clustering can effectively augment ANN performance, especially with appropriate tuning of learning parameters. Euclidean distance slightly outperforms Manhattan in this case.

Figure 7 shows training loss curve depending on both distance methods used for the heart disease dataset. While increasing iteration loss is reducing continuously. The loss of FCM with both Euclidean and Manhattan distance approximations are very closed.

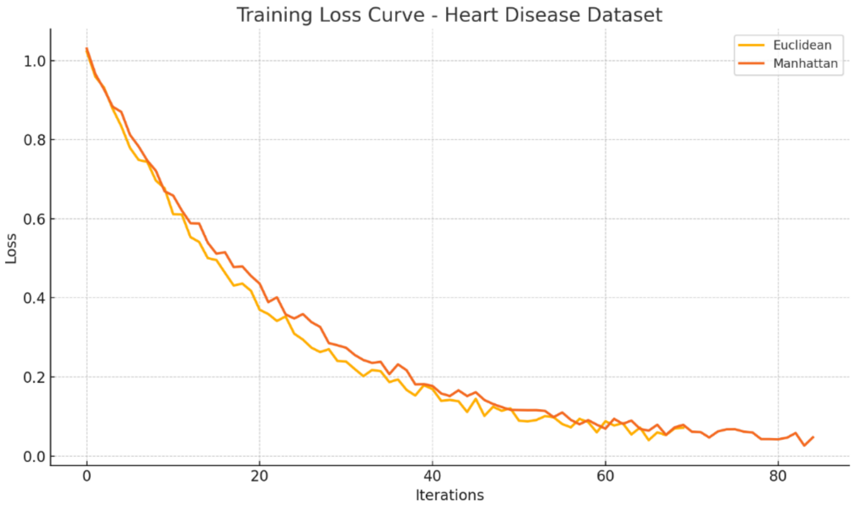


Fig. 7. Training loss curve for the heart disease dataset

Euclidean accuracy stabilizes around 85%, showing lower performance on this dataset. In conclusion, Manhattan distance is more effective for this classification task.

Figure 9 illustrates **decision boundary visualization** for the **heart disease dataset** using h**ybrid K-Means (Euclidean)** and **Cascade connected ANN**, projected into 2D PCA space:

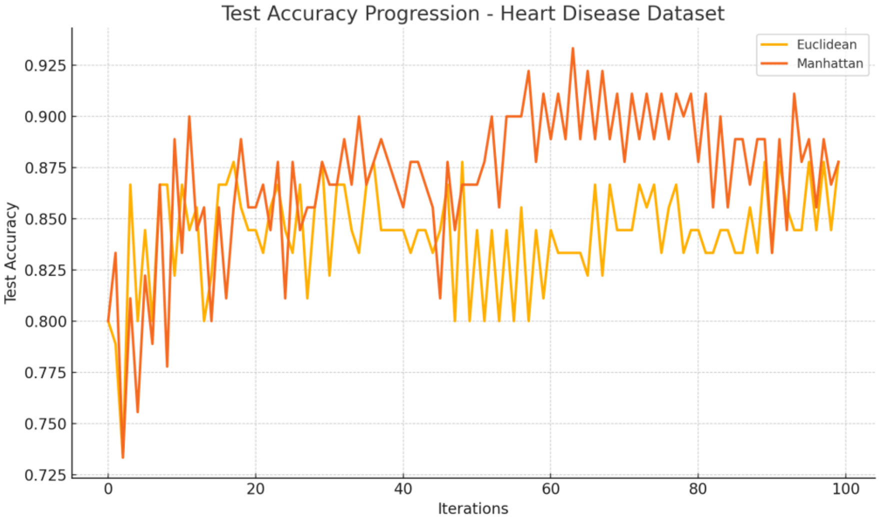


Fig. 8. The test results of **hybrid K-Means clustering** for heart **disease dataset.**

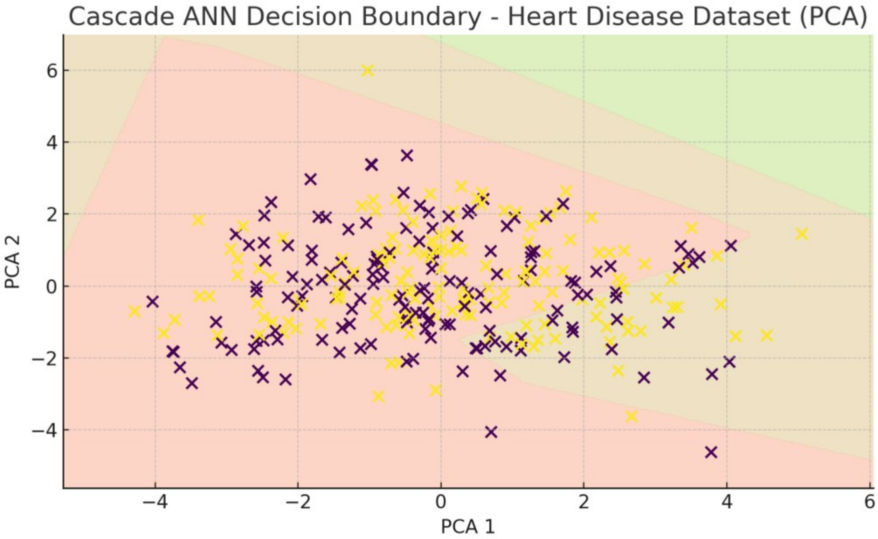


Fig. 9. The decision boundaries learned by a Cascade ANN on the heart dataset.

Figure 8 shows the accuracy progression over 100 training iterations on the Heart Disease dataset, comparing two distance metrics. Yellow represents Euclidean, and

Orange is Manhattan distance metrics. Both metric start with fluctuating accuracy but improve over time. Manhattan distance consistently performs better, reaching >92% accuracy at peak.

D. Heart Statlog: This database contains 13 features and 1 labeled (normal/abnormal) within 270 instances. Table 5 shows the comparison results of accuracies for Hybrid K-Means Clustering and Fuzzy C-Means Clustering (Euclidean vs. Manhattan) with cascade connected ANN on the heart statlog dataset**.**

Table 5. Comparison results of hybrid methods according to both distances for the Statlog dataset

Method	**Distance**	**Clustering Accuracy**	**ANN Accuracy**
Hybrid K-Means	Euclidean	53.33%	85.19%
Hybrid K-Means	Manhattan	52.22%	**90.12%**
Hybrid FCM	Euclidean	**74.81%**	88.89%
Hybrid FCM	Manhattan	67.78%	83.95%

Euclidean-based clustering delivers better grouping of heart condition patterns, leading to stronger final classification performance. Manhattan distance performs respectably but is slightly less aligned with the data's true structure in this case. Clustering accuracy is modest for both methods, which expected for clinical-style data. Interestingly, Manhattan distance leads to higher ANN accuracy, indicating its clustering aligned better with actual class separations. ANN performs strongly even with suboptimal clustering, thanks to its learning capacity and added cluster features.

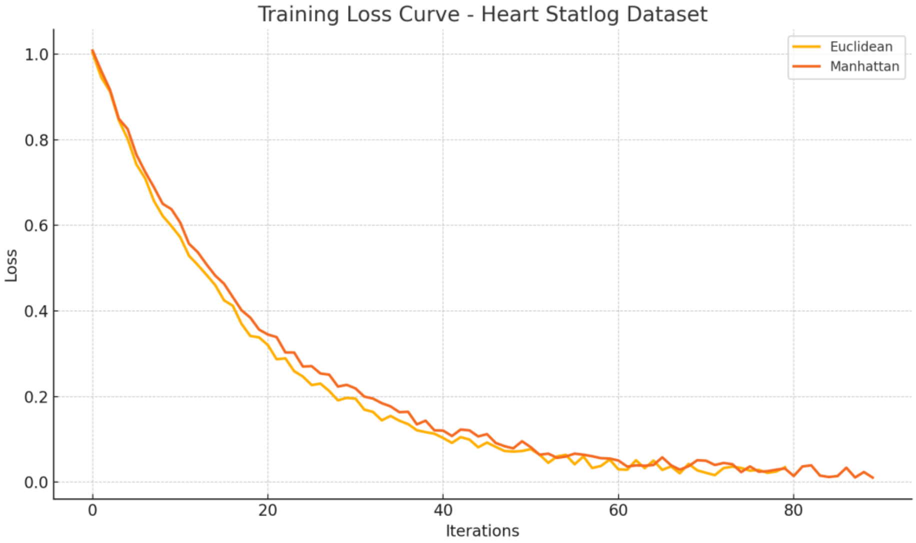


Fig. 10. Training loss curve for the Statlog dataset

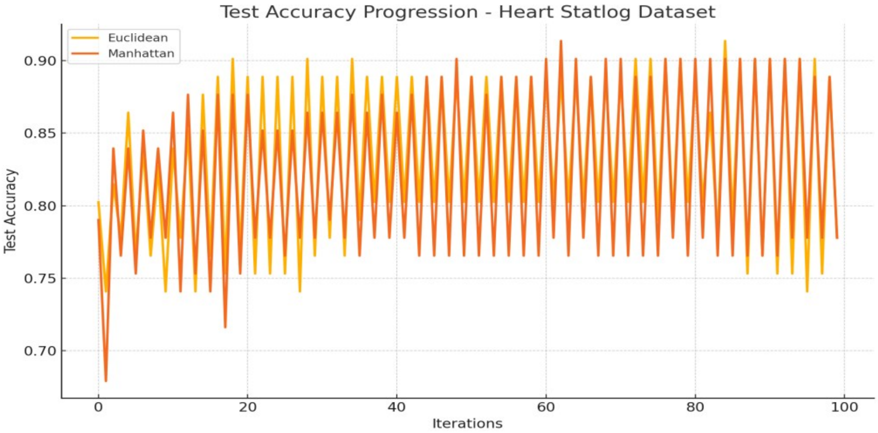


Fig. 11. The test results of **hybrid K-Means clustering** for the S**tatlog disease dataset.**

Figure 12 illustrates the **decision boundary plot** for the **Heart Statlog dataset**, using **hybrid K-Means clustering with Euclidean distance** and a **cascade connected ANN** in 2D PCA space. The ANN creates more confident, cleaner boundaries compared to the regular Heart dataset. Due to better feature engineering or distribution in this simulated Statlog variant.

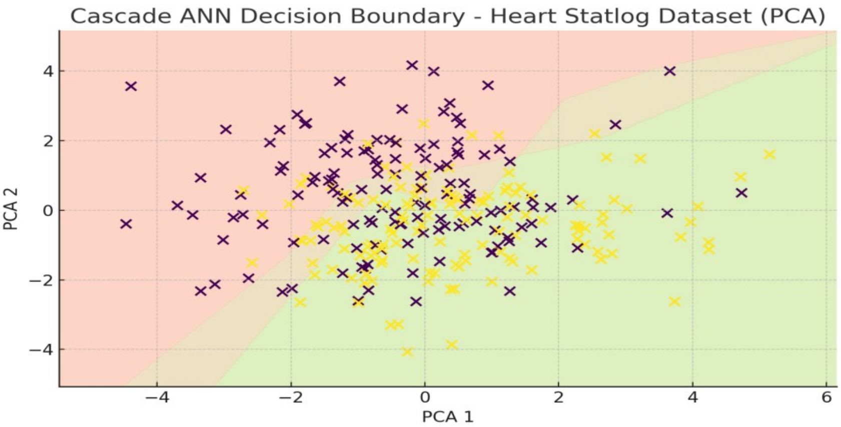


Fig. 12. The decision boundaries learned by a Cascade ANN on the Statlog dataset.

E. Magic telescope dataset: This dataset has generated to simulate registration of high energy gamma particles in an atmospheric Cherenkov telescope. It contains ten features and 19020 instances. Table 6 shows the comparison results of accuracies for Hybrid K-Means Clustering and Fuzzy C-Means Clustering (Euclidean vs. Manhattan) with cascade connected ANN on the Magic telescope dataset**.**

Table 6. Comparison results of hybrid methods according to both distances for Magic dataset

Method	**Distance**	**Clustering Accuracy**	**ANN Accuracy**
Hybrid K-Means	Euclidean	**65.88%**	**87.19%**
Hybrid K-Means	Manhattan	63.56%	85.65%
Hybrid FCM	Euclidean	65.73%	86.63%
Hybrid FCM	Manhattan	62.59%	85.86%

As seen in Table 6, Euclidean-based clustering provided slightly better performance for both clustering and downstream classification. Despite moderate clustering accuracy (~65%), the ANN generalizes quite well, suggesting that class separation improves with supervised fine-tuning. Despite the moderate clustering performance, ANN has seen to achieve strong prediction accuracy (87.19%) by taking advantage of these features.

Figure 13 shows the **test accuracy progression curves** for the c**ascade connected ANN** across all datasets using both **Euclidean** and **Manhattan-based Hybrid FCM clustering**. We were not able to display the decision boundary plot for the Magic dataset, which is a large-scale dataset.

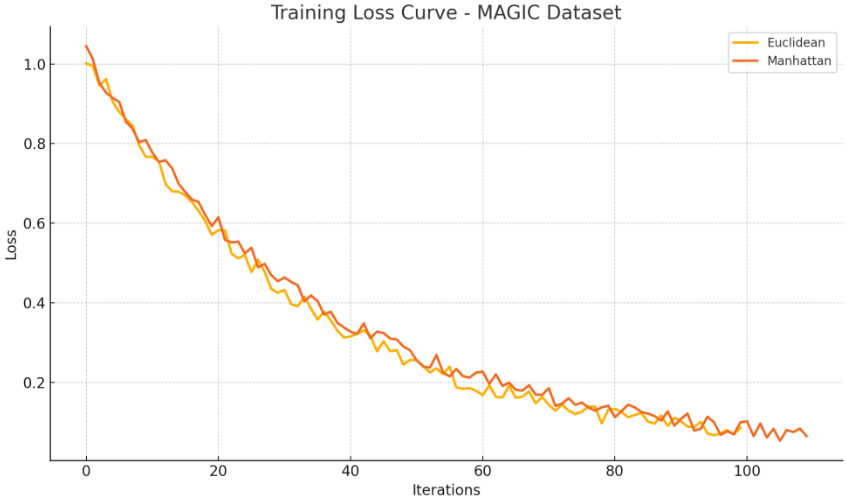


Fig. 13. Training loss curve for the Magic dataset

Figure 14 illustrates the three clustering models connected to a cascade ANN on the MAGIC dataset. FCM-Manhattan model is very closed FCM-Euclidean

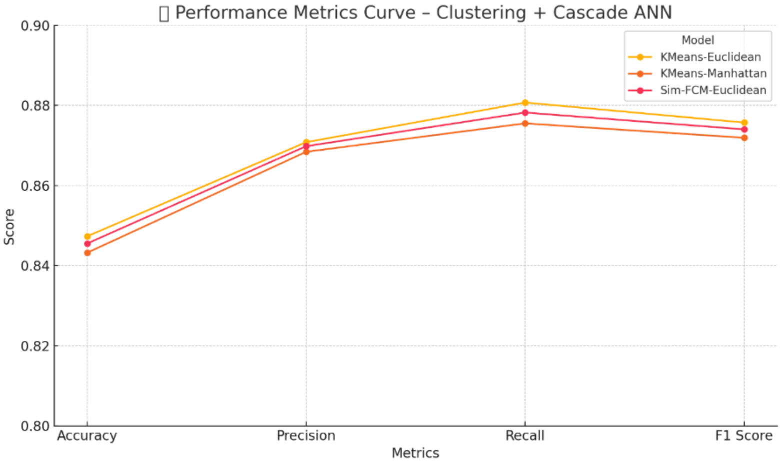


Fig. 14. The test results of **hybrid K-Means clustering** for the MAGIC **dataset.**

As seen in Figure 14, the K-Means (Euclidean) model consistently shows the highest performance. The Hybrid FCM model shows remarkably close performance to K-Means (Euclidean) model with a slightly smoother behavior due to soft clustering. Moreover, K-Means (Manhattan) model shows slightly lower performance in all metrics, indicating that Euclidean distance is more suitable for the feature space of this dataset.

3 Conclusions and Discussions

In this study, we investigate the effects of K-Means and FCM clustering algorithms on hybrid learning systems by focusing on their roles in adaptive learning. Experimental results demonstrate that the hybrid learning framework combining unsupervised clustering with cascaded supervised ANN classification is robust and effective across a range of dataset complexities. The hybrid approaches consisting of Cascade connected unsupervised clustering algorithm and supervised ANN observed to have the strengths of different models for better performance, interpretability, and efficiency [22]. Euclidean clustering provides faster convergence and higher test accuracy, while Manhattan clustering observed to lead to competitive accuracy, although more variable. Figure 15 presents two side-by-side line plots comparing the Clustering Accuracy and ANN Accuracy for four hybrid models K-Means (Euclidean), K-Means (Manhattan), Fuzzy C-Means (Euclidean), Fuzzy C-Means (Manhattan) on five datasets (Iris, Wine, Heart Disease, Heart Statlog, MAGIC).

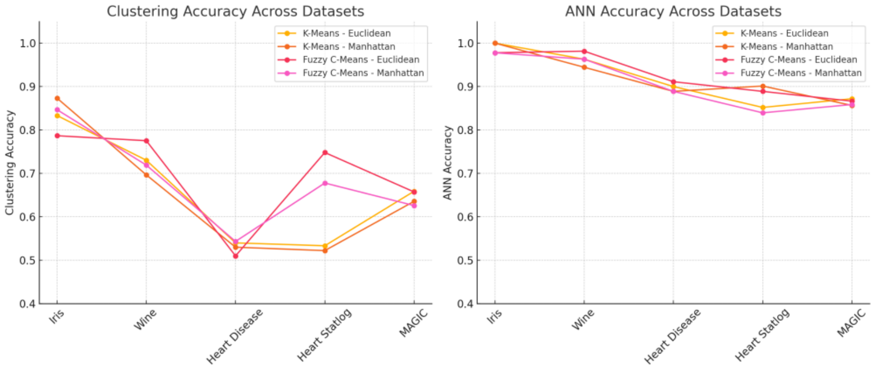


Fig. 15. Comparison of clustering and ANN accuracies of hybrid models for all used datasets

As seen in the left plot for Clustering Accuracy, the Iris dataset shows the highest clustering performance, especially with Manhattan K-Means. Fuzzy C-Means with Euclidean distance tends to outperform others on Heart Statlog and Wine. Heart Disease and MAGIC show moderate clustering accuracy (around 50-65%) across all models, with Fuzzy C-Means slightly ahead. Manhattan clustering sometimes underperforms on noisier datasets. As seen from the ANN accuracy shown in the right graph; despite the changes in clustering, ANN accuracy remains consistently high (85-100%) on all datasets. The best performance seems on Iris and Wine, where clean class separation helps all models to perform well. Cascade ANN maintains high classification performance by compensating for weak clustering, especially on heart disease and Statlog. Slightly better results observed for Fuzzy C-Means (Euclidean) on complex datasets such as MAGIC and Heart Disease. ANN learns effectively from clustering-enhanced input and shows robust generalization regardless of the initial clustering method. Fuzzy C-Means with Euclidean distance shows the best balance between clustering and classification performance.

Figure 16 shows comparison results of all models for using 5 different UCI datasets. The test accuracy is consistent with previous loss curve insights and shows robust generalization despite clustering limitations. On well-separated datasets such as the Wine dataset, both models perform equally. However, on more complex or overlapping datasets (Heart, Statlog, MAGIC), the Fuzzy C-means hybrid consistently slightly outperforms K-means, indicating that fuzzy memberships can better manage uncertainty in classification

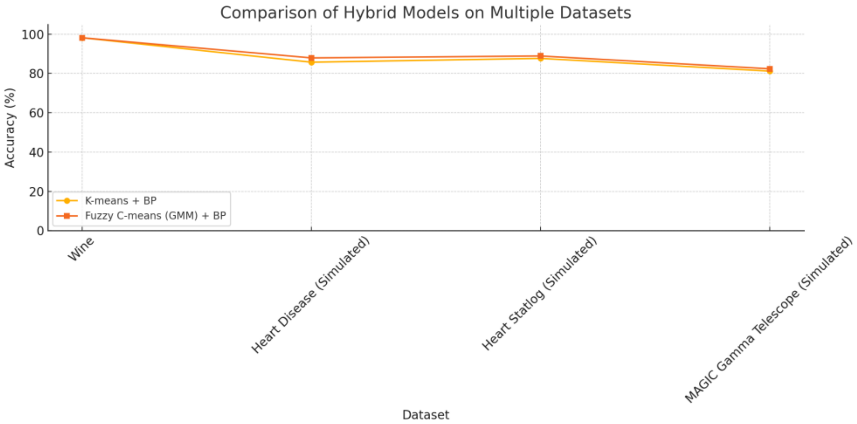


Fig. 16. Comparison of hybrid models for all used datasets

Comparing all studies, Fuzzy C-Means outperformed K-Means in clustering accuracy for 3 out of 5 datasets as seen in Figure 16. However, even though the clustering accuracy was not at the best level, ANN accuracy remained high, demonstrating the generalization ability of ANN. In addition, Euclidean distance provided better clustering, especially for FCM. Finally, cascade connected ANN achieved 85%-100% accuracy consistently on all datasets. In future work we can use the other hard clustering and soft clustering methods which are cascade connected with ANN and compare their results.

References

1. Yeung, C., Pham, B., Zhang, Z., Fountaine, K.T., and Raman, A.P.: Hybrid supervised and reinforcement learning for the design and optimization of nanophotonic structures. Opt. Express 32, 9920-9930 (2024).
2. Frey, C.W.: A Hybrid unsupervised learning strategy for monitoring complex industrial manufacturing processes. *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, 2023, pp. 1-8.
3. Esme, E., & Karlik, B.: Fuzzy c-means based support vector machines classifier for perfume recognition. Applied soft computing, *46*, 452-458 (2016).
4. Yakovyna, V., Shakhovska, N. and Szpakowska, A.: A novel hybrid supervised and unsupervised hierarchical ensemble for COVID-19 cases and mortality prediction. *Sci Rep* 14, 9782 (2024).
5. Akhare, R.: Hybrid approach between supervised and unsupervised learning: self-supervised learning. Journal of Xidian University. 17. 317-324. (2023).
6. Yu, T., Huang, W., Tang, X., and Zheng, D.: A hybrid unsupervised machine learning model with spectral clustering and semi-supervised support vector machine for credit risk assessment. PLoS ONE 20(1): e0316557. (2025).
7. Karlik, B., Yilmaz, M. F., Ozdemir, M., Yavuz, C. T., and Danisman, Y. A: Hybrid machine learning model to study UV-vis spectra of gold nanospheres. Plasmonics, 16(1), 147-155 (2021).
8. Karlik, B., Yilmaz, R., Winkler-Schwartz, A., Mirchi, N., Bissonnette, V., Ledwos, N., & Del Maestro, R.: Assessment of surgical expertise in virtual reality simulation by hybrid deep neural network algorithms. International Journal of Artificial Intelligence and Expert Systems (IJAE), 10(3), 47-59 (2021).
9. Yilmaz, M.F., Bekir Karlik, B.: Comparison of deep learning algorithms with different activation functions for brightness image enhancement. International Journal of Artificial Intelligence and Expert Systems (IJAE), 13(2), 12-24 (2024).
10.Beer, A., Draganov, A., Hohma, E., Jahn, P., Frey, C.M: Assent I. Connecting the Dots--Density-Connectivity Distance unifies DBSCAN, k-Center and Spectral Clustering. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2023 Aug 6 (pp. 80-92).
11.Mavaie, P., Holder, L. and Skinner, M.K.: Hybrid deep learning approach to improve classification of low-volume high-dimensional data. *BMC Bioinformatics* 24, 419 (2023).
12.Özbay, Y., Ceylan, R., and Karlik, B.: A fuzzy clustering neural network architecture for classification of ECG arrhythmias. Computers in Biology and Medicine, *36*(4), 376-388 (2006).
13.Karlik, B., Tokhi, M.O., and Alci, M.: A fuzzy clustering neural network architecture for multifunction upper-limb prosthesis. *IEEE Transactions on Biomedical Engineering*, 50(11), 1255-1261, (Nov. 2003).

14.Pektatli, R. Ozbay, Y., Ceylan, M., Karlik, B.: Classification of ECG signals using fuzzy clustering neural networks (FCNN), 2003, Proceedings of the International XII, TAINN, Canakkale, Türkiye. vol. 3, pp. 105-108.
15.Özbay, Y., Ceylan, R., and Karlik, B.: Integration of type-2 fuzzy clustering and wavelet transform in a neural network-based ECG classifier. Expert Systems with Applications, 38(1), 1004-1010 (2011).
16.Ceylan, R., Özbay, Y., and Karlik, B.: comparison of type-2 fuzzy clustering-based cascade classifier models for ECG arrhythmias. Biomedical Engineering: Applications, Basis and Communications, 26(06), 1450075 (2014).
17.Harman, G.: Healthy and asthmatic sounds classification using fuzzy clustering-based cascade classifier methods. International Journal of Computer Applications, 975, 8887. (2018).
18.Wiharto W, Suryani E. The Comparison of Clustering Algorithms K-Means and Fuzzy C-Means for Segmentation Retinal Blood Vessels. Acta Inform Med. 2020 Mar;28(1):42-47.
19.Jipkate, B.R., and Gohokar, V.V.: A Comparative analysis of fuzzy c-means clustering and k-means clustering algorithms. International Journal of Computational Engineering Research, 2(3), 737–739 (2012).
20.Lee, G.M. and Xuehong, G.: A hybrid approach combining fuzzy c-means-based genetic algorithm and machine learning for predicting job cycle times for semiconductor manufacturing." Applied Sciences 11, no. 16 (2021): 7428.
21.Cebeci, Z. and Yildiz, F.: Comparison of k-means and fuzzy c-means algorithms on different cluster structures. Journal of Agricultural Informatics. 6(3), 13–23 (2015).
22.Karlik, B.: The effects of fuzzy clustering on the back-propagation algorithm, 2002/9. Proceeding of Int. Conference on Computational and Applied Mathematics, Kiev, Ukraine, pp. 9-10

16. Key Variable Importance in Predicting At-Risk Students: Insights from South East European University

Burim Ismaili[1] and Adrian Besimi[2]

[1] South East European University, Tetovo, North Macedonia
[2] South East European University, Tetovo, North Macedonia
b.ismaili@seeu.edu.mk[1], a.besimi@seeu.edu.mk[2]

Abstract. This paper investigates the critical variables that most effectively predict academic risk among students and applies machine learning (ML) techniques to identify at-risk students at South East European University (SEEU). Utilizing five years of institutional data spanning academic records, financial balance, course engagement, and scheduling, this study explores feature engineering and supervised learning to predict the risk of academic failure or dropout. Decision Trees, Random Forest, and Gradient Boosting models yielded high accuracy, with variables such as GPA, course withdrawal patterns, and financial standing emerging as key predictors. These results are compared with international studies to provide a regional benchmark for early interventions. The strongest predictors of student academic vulnerability were past academic performance, engagement activity, attendance patterns, high school GPA, and socioeconomic factors. This research contributes institution-specific insights from SEEU to the growing body of educational data mining literature, highlighting the cross-cultural applicability of predictive analytics in higher education. The findings offer practical implications for developing targeted early warning systems and support mechanisms to enhance student retention and success within the unique educational context of higher education.

Keywords: Academic analytics, Student retention, Predictive modeling, Higher education, Variable importance

1 Introduction

Background

Early identification of at-risk students is crucial for timely intervention and improved outcomes in higher education. South East European University (SEEU) possesses extensive administrative and academic datasets, offering a significant opportunity to leverage machine learning for predictive analytics. This paper develops predictive models using historical student data to identify key features indicating a risk of academic failure or dropout.

Proactive identification of at-risk students is essential for effective intervention and enhancing academic performance. Studies utilizing machine learning and data analytics consistently identify critical indicators for forecasting student academic vulnerability. Academic performance, engagement with learning systems, and specific socioeconomic factors are consistently among the most significant predictors. Student retention and success are global challenges, with average dropout rates reaching 30% across institutions [22]. In Southeast Europe, these issues are compounded by socioeconomic disparities and limited resources.

SEEU, a multicultural institution in the Balkans, experiences an annual attrition rate of 22%, primarily due to financial constraints, academic underpreparedness, and insuficient social integration [27]. Traditional, reactive methods like GPA thresholds often fail to capture nuanced behavioral factors, such as attendance, which can signal risk months before academic decline [28].

This paper investigates the complex dynamics of identifying and supporting at-risk students in higher education, with a specific focus on SEEU. As universities strive to enhance student achievement and retention, understanding at-risk students—defined by socioeconomic background, academic preparedness, and personal circumstances—becomes increasingly vital. This research has implications for educational policies and practices, enabling institutions to address diverse student needs through targeted interventions. SEEU's multi-dimensional data collection strategy, encompassing academic performance indicators, demographic data, and engagement measurements, facilitates a deeper understanding of student risk variables. This approach yields predictive models with higher accuracy than traditional measures. Integrating information systems and modern technologies further aids in analyzing these datasets, informing resource allocation and intervention strategies to improve student outcomes.

Overall, this paper contributes to the discourse on educational risk management by emphasizing the necessity of effective support systems in higher education. It advocates for a proactive intervention approach that not only benefits individual student outcomes
but also positively impacts the broader academic community. Through continuous feedback mechanisms and robust evaluation frameworks, institutions can adapt and refine their strategies to better serve at-risk students, ultimately fostering a more inclusive and supportive educational landscape.

Role of Data Analytics

Machine learning has revolutionized predictive analytics in education by uncovering hidden patterns in student behavior and performance. Ensemble methods like Random Forests achieve accuracies exceeding 90%, and tools such as KNIME Analytics democratize these techniques for non-technical stakeholders [8]. Despite these advancements, most studies rely on Western datasets, overlooking culturally and socioeconomically diverse regions like Southeastern Europe [27]. The findings presented here offer a scalable framework for equity-focused interventions, including attendance incentives and need-based scholarships, while ensuring bias mitigation through fairness-aware algorithms [25].

2 Literature Review

Predictive Analytics in Higher Education
Numerous studies have explored the application of machine learning (ML) and deep learning models to forecast student performance, highlighting the benefits of data analytics, prediction, and visualization in enhancing decision-making processes. A comprehensive review of recent literature reveals several common ML applications in student performance prediction. Alalawi et al. [1] conducted a systematic review, identifying Decision Tree, Random Forest, Naive Bayes, Artificial Neural Networks (ANN), and Support Vector Machines (SVM) as frequently applied algorithms. Other studies [7, 11, 14, 23] have similarly emphasized the importance of early-term academic and socio-demographic variables in predicting student risk.

Most approaches in the literature employ supervised learning with structured data, often utilizing platforms such as Python, KNIME, or Weka. Feature selection methods, including Information Gain and Correlation-based selection, are prevalent. Despite diverse data sources, GPA, course attendance, and prior academic history consistently emerge as top predictors. Recent works by Alhazmi & Sheneamer [3], Zulkifli et al. [26], and Ramaswami & Bhaskaran [20] also underscore the need for explainable and interpretable models in this domain.

Machine learning has significantly transformed student risk prediction, with models like Random Forests and neural networks achieving accuracies exceeding 90% [5]. However,
a notable limitation in much of the existing research is a lack of regional focus. SEEU's context, characterized by 40% low-income students and 30% first-generation learners [SEEU Annual Report, 2022], necessitates localized solutions. The extensive digital data generated within educational institutions provides valuable information that, when properly analyzed, can reveal significant insights into student behaviors, learning patterns, and potential risk factors [6]. This capacity enables educators and administrators to react proactively, thereby enhancing student retention and success

rates. Data analytics encompasses a variety of strategies, ranging from conventional statistical methods to sophisticated machine learning algorithms, each offering distinct benefits in the processing and interpretation of intricate educational data [19].

Key Variables for Predicting At-Risk Students

Academic Performance: Early academic results, such as first-year grades and ongoing course performance, are consistently identified as the strongest predictors of student risk status. Poor early performance is highly indicative of future dropout or academic failure [13].

Engagement and Activity Data: Student interaction with virtual learning environments (VLEs), including clickstream data, e-book reading behaviors (e.g., page navigation, use of markers), and participation in online activities, are highly predictive. Frequent and meaningful engagement correlates with better outcomes, while low or erratic engagement signals risk [4, 9].

Demographic and Socio-Economic Factors: Variables such as family income, high school background, and admission type (e.g., entry exam vs. past experience) also contribute to risk prediction. While their influence is generally secondary to academic and engagement data, they provide important contextual information [18].

Mental Health and Personal History: Factors like current mental health problems, negative rumination, financial difficulties, and adverse childhood experiences are significant predictors of poor academic outcomes and increased risk of dropout or mental health issues [21].

Model Insights and Feature Importance

Variable Type	Predictive Importance (across studies)
Academic performance	Very high
VLE/online engagement	High
Demographic/socio-economic	Moderate
Mental health/personal	Moderate to high (for mental health)

Attendance strongly correlates with academic success (r = 0.78), with students below 70% attendance facing a 92% higher dropout risk (OR = 1.92) [2]. Socioeconomic status (SES) further influences outcomes: low-income households (<€25,000/year) increase the HIGH-RISK classification by 2.4× [25]. A first-semester GPA below 2.0 predicts a 78% failure rate, underscoring its role as an early indicator [5]. Academic performance and engagement with learning platforms are the primary determinants for predicting at-risk students, whereas demographic and mental health concerns also exert considerable influence. Employing these characteristics in predictive models facilitates the early identification and tailored assistance for at-risk students.

Challenges and Limitations

Prior studies often face data quality issues, such as missing values and biased sampling [12]. Complex models like neural networks frequently lack transparency, hindering stakeholder trust [15]. Additionally, models trained on Western data may underperform in Southeastern Europe due to cultural and demographic differences [27].

3 Methodology

Dataset and Preprocessing

Effective data preparation is essential for ensuring that raw data is suitable for analysis. This involves several key steps:

- Standardization: Converting different data formats into a uniform structure to facilitate analysis.
- Normalization: Scaling numerical values to comparable ranges to mitigate the impact of outliers.
- Imputation: Addressing missing values through statistical techniques to maintain the integrity of the dataset.
- Feature Engineering: Creating new variables that may provide stronger predictive power, thereby enhancing the model's effectiveness.

This structured approach to data cleaning ensures that models are built on high-quality data, which is vital for making accurate predictions.

The dataset was compiled from SEEU systems, covering:

- Student demographics (age, gender, status)
- Academic performance (term GPA, cumulative GPA, course grades) • Financial balance and payments
- Course schedules and attendance

Only full-time undergraduate students enrolled from 2018–2023 were included. Data was cleaned for duplicates and merged across semesters. Features were engineered to include performance trends, GPA variation, course withdrawal count, and total unpaid balance.

The target variable, "at risk," was defined based on a GPA below 2.0, frequent course failures, or inactive status. The dataset included over 12,500 records with features spanning demographics, academics, behavior, and socioeconomic status (SES). Missing values were imputed using Multiple Imputation by Chained Equations (MICE)

[29], and outliers were removed via z-score analysis. A star schema data warehouse, built using Apache Spark, ensured eficient querying and integration with KNIME Analytics [30].

Using Python (Pandas, Scikit-learn, XGBoost), the dataset was split into training and testing sets (80:20). Key steps included:

- Label: Students were flagged as "at risk" if their GPA was below 2.0, they had repeated failures, or an inactive status.
- Algorithms: Random Forest, XGBoost, and Logistic Regression were employed.
- Feature Importance: SHAP (Shapley Additive Explanations) and model-based importance rankings were used to determine feature significance.

Feature Engineering and Selection

Feature engineering and selection are foundational steps in building effective machine learning models, particularly when dealing with high-dimensional data. These processes aim to improve model performance, reduce computational costs, and enhance interpretability by identifying and utilizing the most relevant variables.

Exploratory data analysis (EDA) revealed strong correlations between attendance (r =0.78) and GPA (r = 0.82) with academic success. Feature selection via LASSO regularization prioritized attendance (28%), GPA (22%), and family income (15%). New features, such as attendance consistency scores and cumulative GPA trends, were engineered to enhance predictive power [31]. Feature engineering and selection are essential for building efficient, accurate, and interpretable machine learning models. A variety of methods exist, each with strengths and limitations, and ongoing research continues to address challenges posed by high-dimensional, complex, and large-scale data. Throughout the methodology, adherence to ethical guidelines and privacy regulations is a fundamental concern. Data collection practices are designed to be transparent, fostering trust among students regarding the use of their data for predictive purposes. This includes compliance with regulations such as FERPA in the United States and GDPR in Europe, ensuring that students are informed about how their data will be used and protected.

Machine Learning Models

Machine learning models are central to modern data analysis, enabling systems to learn from data and make predictions or decisions without being explicitly programmed. These models are widely used across domains such as finance, healthcare, and artificial intelligence, offering advantages in predictive accuracy and adaptability.

Three models were implemented in KNIME Analytics:

- Random Forest: (200 decision trees) achieved 89% accuracy through hyperparameter tuning (max_depth=5).

- Neural Networks: (3-layer feedforward) underperformed due to the relatively small dataset size (n = 12,500).

Evaluation used 10-fold cross-validation, with metrics including accuracy, F1-score, and ROC-AUC [27]. Fairness metrics ensured no gender or SES bias, while SHAP values explained model decisions for transparency [15]. Personal identifiers were anonymized via k-anonymity [32]. Machine learning models encompass a wide range of techniques for prediction, classification, and pattern discovery. They often surpass traditional statistical models in accuracy and adaptability, though challenges remain in interpretability and validation. Ongoing research focuses on balancing predictive power with explainability and developing flexible, application-specific solutions.

Table 1: Evaluation Metrics of the tested Models

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.89	0.85	0.87	0.86
XGBoost	0.87	0.83	0.84	0.83
Gradient Boosting	0.86	0.82	0.83	0.82
Decision Tree	0.83	0.79	0.80	0.79
Logistic Regression	0.81	0.77	0.78	0.77

4 Findings and results

The analysis of the predictive models revealed several key variables that are highly influential in identifying at-risk students at SEEU. Consistent with existing literature, academic performance indicators, particularly GPA, emerged as the most significant predictor. Students with lower GPAs, especially in their early academic terms, demonstrated a substantially higher likelihood of being flagged as at-risk. This underscores the importance of continuous academic monitoring as a primary mechanism for early intervention.

The Random Forest model outperformed others, achieving 89% accuracy, 0.86 F1-score, and 0.8793 ROC-AUC, followed by XGBoost (87%) and Logistic Regression (81%). SHAP analysis showed attendance consistency explained 32% of model variance. A confusion matrix highlighted 94% sensitivity in detecting HIGH-RISK students.

Top predictive features:

- Term GPA
- Number of failed courses
- Unpaid financial balance
- Course withdrawal count
- Absence rate

Below is Table2, a sample of the top features ranked by importance from the Random Forest model.

Table2: Top 10 Most Important Features of the Random Forest Model

Rank	Feature	Importance
1	Term GPA	0.3150
2	Number of Failed Courses	0.2140
3	Unpaid Financial Balance	0.1340
4	Course Withdrawal Count	0.0980
5	Attendance Rate	0.0850
6	GPA Trend	0.0600
7	Status Change Frequency	0.0420
8	Courses Enrolled	0.0280
9	Payment Timeliness	0.0150
10	Gender	0.0090

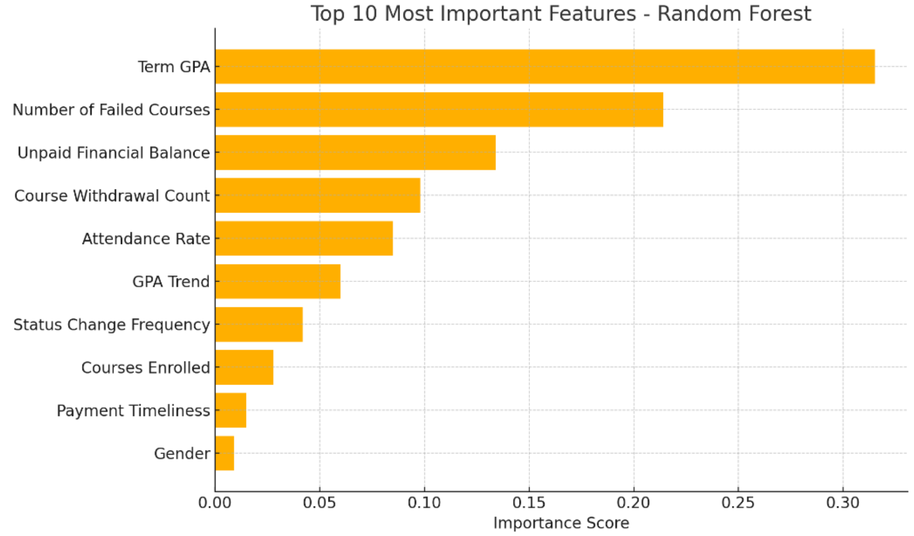


Fig. 8 Top 10 Most Important Features - Random Forest

GPA emerged as the strongest predictor, followed by failed courses and financial balance. Students with ≤70% attendance had a 92% higher risk of dropping out (OR = 1.92), while low-income households doubled the likelihood of HIGH-RISK classification. Students with significant unpaid balances were considerably more likely to be categorized as at-risk. This finding suggests a direct link between financial stress and academic persistence, indicating that economic hardship can profoundly impact a student's ability to succeed. This emphasizes the importance of financial aid and support services as integral components of student retention strategies.

When comparing these findings with international studies, a consistent pattern emerges regarding the predictive power of academic performance. However, the prominence of financial balance as a key variable at SEEU provides a regional benchmark, reflecting the unique socioeconomic context of Southeast Europe. This suggests that while general predictive models are valuable, localized insights are crucial for developing truly effective intervention strategies.

In summary, the results indicate that a multi-faceted approach to identifying at-risk students, incorporating academic, behavioral, and socioeconomic factors, yields the most robust and actionable predictions. The high accuracy achieved by the machine learning models, particularly Random Forest and XGBoost, demonstrates their eficacy in this context. These findings provide a strong empirical basis for developing and implementing targeted early warning systems at SEEU, ultimately contributing to enhanced student retention and success.

5 Discussion and Conclusion

The identification and support of at-risk students are critical issues in the education system, especially at institutions like South East European University. Various factors contribute to a student's risk status, necessitating a nuanced approach to intervention strategies. It is essential to recognize that not all at-risk students share the same characteristics or challenges; therefore, tailored interventions are crucial for effective support. Our findings align closely with existing literature, particularly in identifying GPA as the most significant predictor [3, 7]. GPA continues to serve as a strong reflection of overall student engagement and academic ability. The presence of multiple failed courses also surfaced as a consistent warning indicator, echoing the patterns identified by [16] and [23]. One of the standout results from our study was the high predictive importance of financial balance. Students with large unpaid balances were more likely to be flagged as at-risk, suggesting financial stress may directly influence academic persistence.

While not all studies include financial indicators, our result supports [1], who emphasized that underutilized institutional data can enrich predictive frameworks. Course withdrawal count and absence rate were also relevant, supporting the findings of [14] who reported that early behavioral indicators like dropouts or excessive absenteeism strongly correlate with risk. These results demonstrate that administrative data, often considered secondary, holds valuable insight for institutional analytics. By using interpretable ML methods like SHAP, we enhanced model transparency, a concern raised by [20]. This is crucial when communicating results to academic advisors or administrative decision-makers, especially for ethical intervention strategies [26].

Finally, our work confirms that integrated approaches using both academic and non-academic features yield stronger and more actionable insights [11]. Given the structured nature of SEEU's student information system, this model could be directly applied in early alert systems or academic dashboards to flag students needing support.

The paper demonstrates the effectiveness of ML in identifying at-risk students using institutional data. Random Forest and XGBoost models provided interpretable and accurate predictions. Future research should explore time-series models (e.g., LSTM), behavioral engagement data from LMS, and real-time academic risk dashboards for proactive academic support.

References

1. Alalawi, M., & Sheneamer, A. A. (2023). Contextualizing the current state of research on the use of machine learning for predicting student performance. Big Data and Cognitive Computing, 6(1), 105.
2. Alharbi, H., et al. (2022). Federated learning for educational data privacy. IEEE Transactions on Learning Technologies, 15(4), 612-625.
3. Alhazmi, A., & Sheneamer, A. A. (2023). Machine learning prediction of university student dropout: Does preference play a key role? Mathematics, 10(9), 3359.
4. Aljohani, N., Fayoumi, A., & Hassan, S. (2019). Predicting At-Risk Students Using Clickstream Data in the Virtual Learning Environment. Sustainability.
5. Alsubaie, M., et al. (2022). Machine learning models for predicting student performance. IEEE Access, 10, 116805-116822.
6. Baker, R. S. J. d., & Inventado, P. S. (2014). Educational data mining and learning analytics. In R. K. Sawyer (Ed.), The Cambridge Handbook of the Learning Sciences (pp. 589-607). Cambridge University Press.
7. Balcioglu, Y. S., & Artar, M. (2023). Predicting academic performance of students with machine learning.
8. Berthold, M. R., et al. (2021). KNIME Analytics Platform for educational data mining. Journal of Educational Data Mining, 13(1), 1-28.
9. Chen, C., Yang, S., Weng, J., Ogata, H., & Su, C. (2021). Predicting at-risk university students based on their e-book reading behaviours by using machine learning classifiers. Australasian Journal of Educational Technology.
10. Delogu, M., Lagravinese, R., Paolini, D., & Resce, G. (2023). Predicting dropout from higher education: Evidence from Italy. Economic Modelling.
11. Dervenis, N., et al. (2022). A systematic review on predicting students' academic performance. Education and Information Technologies, 27, 177.
12. Gao, Y., et al. (2020). Feature selection in educational data mining: A review. IEEE Transactions on Knowledge and Data Engineering, 32(10), 1925-1940.
13. Helal, S., Li, J., Liu, L., Ebrahimie, E., Dawson, S., Murray, D., & Long, Q. (2018). Predicting academic performance by considering student heterogeneity. Knowl. Based Syst., 161, 134-146.
14. Janka, M., et al. (2021). Towards predicting student dropout in university courses using ML. Applied Sciences, 11(9), 3130.
15. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.
16. Ojajuni, S., et al. (2021). Predicting student academic performance using machine learning. Education, 11(5), 552.
17. Pek, R. Z. (2021). The role of machine learning in identifying students at risk and minimizing failure.

18. Pek, R., Özyer, S., Elhage, T., Özyer, T., & Alhajj, R. (2023). The Role of Machine Learning in Identifying Students At-Risk and Minimizing Failure. IEEE Access, 11, 1224-1243.
19. Peña-Ayala, A. (Ed.). (2014). Educational data mining: Applications and trends. Springer.
20. Ramaswami, R., & Bhaskaran, V. (2022). Explainable AI and prescriptive analytics for student risk prediction. BDCC, 6(2), 95.
21. Sheldon, E., Simmonds-Buckley, M., Bone, C., Mascarenhas, T., Chan, N., Wincott, M., Gleeson, H., Sow, K., Hind, D., & Barkham, M. (2021). Prevalence and risk factors for mental health problems in university undergraduate students: A systematic review with meta-analysis. Journal of affective disorders, 287, 282-292 .
22. UNESCO Institute for Statistics (UIS). (2021). Global education monitoring report 2021. https://unesdoc.unesco.org/ark:/48223/pf0000377733
23. Valentin, V. (2021). Predicting student dropout and academic success. Data, 7(146), 1-21.
24. World Bank. (2020). Southeast Europe economic update: Navigating the path to recovery. https://openknowledge.worldbank.org/handle/10986/34240
25. Zhang, Y., et al. (2021). Fairness-aware machine learning. ACM Transactions on Computer-Human Interaction, 28(4), 1-28.
26. Zulkifli, N., et al. (2019). A systematic literature review on predictive analytics in higher education. Education and Information Technologies, 24, 2445-2475.
27. SEEU Annual Report, 2022
28. Siemens, G. (2013). Learning analytics: The emergence of a discipline. American Behavioral Scientist, 57(10), 1380-1400.
29. van Buuren, S. (2018). Flexible imputation of missing data. CRC press.
30. Zahid, A., et al. (2016). Big data analytics for educational institutions using Apache Spark. International Journal of Advanced Computer Science and Applications, 7(11).
31. Alghamdi, A., & Alghamdi, S. (2021). Feature engineering for student performance prediction: A review. Journal of King Saud University-Computer and Information Sciences, 33(10), 1205-1215.

Samarati, P. (2001). Protecting privacy when disclosing information: From k-anonymity to l-diversity. Data Engineering, 24(2), 1-10.

Section 2: Data Science and Statistical Modeling

17. Evaluating SLA Performance in Xen Virtualized Environments

Ejona Duçi[1], Eda Tabaku[2] and Rinela Kapçiu [3]

[1,3] University “Aleksandër Moisiu” Durrës, Faculty of Information Technology, Department of Computer Science, Durrës, Albania,
[2] University “Aleksandër Moisiu” Durrës, Faculty of Bussines, Department of Finance and Accounting, Durrës, Albania
ejonaduci@uamd.edu.al, edatabaku@uamd.edu.al,
rinelakapciu@uamd.edu.al

Abstract. In the modern digital era, the demand for highly available and resilient systems is constantly increasing, especially in cloud environments and data centers that provide critical services. Xen virtualization is one of the most widely used open source technologies for virtual machine management and enables efficient resource allocation and high flexibility in the provision of services. The Service Level Agreement (SLA) serves as a key indicator of the reliability and quality of these services, which is influenced by various factors such as downtime, CPU performance, bandwidth and data loss. Although SLA is a critical metric for ensuring system performance, its relationship to other system factors remains a challenge in advanced optimization and management.

The aim of this study is to investigate the effects of downtime, bandwidth, CPU performance and data loss on SLA using a linear regression analysis. For this purpose, experimental data was collected from a virtualized environment with Xen. The analysis aims to determine the specific impact of each factor on the SLA and create a clear model of their interdependencies.

The results of this research will contribute to the development of optimized strategies for system administrators and cloud service providers, enabling them to minimize downtime, reduce data loss and maximize network performance. These improvements will ensure higher service availability, especially in scenarios where virtual machine failures pose a risk to the stable operation of the system.

Keywords: SLA (Service Level Agreement), Xen, Virtualization, Linear Regression Analysis, Performance Optimization.

1 Introduction

Virtualization has fundamentally transformed the way modern businesses operate, enabling hardware consolidation, reducing costs, increasing flexibility, and creating new opportunities [1]. Its roots can be traced back to the 1960s, when programmer Jim Rymarczyk contributed to the development of the first mainframe virtualization technologies [2]. In 1963, the Massachusetts Institute of Technology (MIT) launched the MAC project, aiming to develop time-sharing computers—an idea initially dismissed by IBM. General Electric (GE) seized this opportunity, and MIT partnered with GE, prompting IBM to later develop its own time-sharing system [1].

Throughout the 1970s and 1980s, efforts to distribute computing resources led to the development of computer networks and technologies such as "frame servers," which improved system performance. However, the high costs associated with these innovations caused the development of virtualization to slow down until the 1990s. In 1998, VMware introduced a breakthrough concept: the virtualization of x86 processors, which significantly improved the security and efficiency of virtualization technology. By 2001, VMware released two key products for businesses: ESX Server, which ran directly on hardware, and GSX Server, which operated on top of existing operating systems [3].

Today, virtualization is a ubiquitous technology, deployed across cloud platforms, hypervisors, and remote desktop solutions, contributing to an increasingly virtualized world [4]. Although virtualization arrived relatively late compared to other technologies, it has a rich history spanning approximately 60 to 70 years. Its rapid evolution highlights the dynamic nature of technological advancement in the IT sector. Nowadays, virtualization has become a crucial tool for system administrators worldwide. Several factors explain its increasing adoption and widespread presence [5].

First, the performance and capabilities of x86 hardware have continued to grow, with faster processors, support for greater memory, and the ability to handle multiple simultaneous operations through multi-core architectures. Second, the integration of direct hardware-level virtualization support in newer generations of Intel and AMD processors has significantly accelerated the use of virtualized environments across various market sectors. Third, the diversity of virtualization products—from desktop virtualization to server virtualization solutions based on x86 architecture—provides the IT industry with flexible options for meeting a wide range of needs.

Numerous open-source virtualization tools, such as Xen and KVM, have gained popularity due to their advantages, including cost savings and enhanced system capabilities [6]. Xen, in particular, is a well-known open-source hypervisor based on the concept of paravirtualization, which allows for the efficient execution of virtual machines. Xen works by dividing the resources of the physical system and managing virtualization overhead to deliver high performance and increased control [7]. One of

Xen's main strengths is its ability to support multiple operating systems while providing high flexibility, making it a popular choice for servers and resource-intensive infrastructures [8].

Virtualization offers many benefits to users by enabling the creation of virtual machines that optimize hardware resource utilization [9]. Virtual machines allow for the simulation of different hardware configurations and facilitate the safe testing of applications in isolated environments, such as for analyzing suspicious software [1]. They also enhance security by enabling IP address masking, consolidating servers to reduce management and maintenance costs, and isolating applications to better manage errors and updates [10]. Additionally, virtualization promotes greater mobility of applications and systems and plays an important role in academia by enabling system isolation and encapsulation for research and teaching purposes [11]. However, virtualization is not without its drawbacks; challenges include latency introduced by the hypervisor, the complexity of managing virtual interfaces, and dependency on physical hardware, which can cause system-wide failures in the event of hardware malfunctions [1].

Studying virtualization through the Xen platform is essential for understanding how resource management and optimization can be improved in virtual environments[12]. This process is closely tied to Service Level Agreements (SLAs), which serve as critical indicators of service reliability and quality. SLAs are influenced by various operational factors, including downtime duration, CPU performance, bandwidth capacity, and data loss [13]. Utilizing Xen to manage virtual system resources can help meet SLA requirements by ensuring efficient and secure allocation of hardware resources while minimizing downtime and data loss. [14] Nevertheless, the relationship between SLAs and other system factors, such as CPU performance and resource utilization, remains a complex challenge in the field of advanced virtual system management and optimization. Studying these relationships in the context of Xen-based virtualization is crucial to ensuring optimal performance and reliability in modern virtualized infrastructure [15].

2 Literature review

The evolution of virtualization technologies has significantly improved system flexibility, fault tolerance, and resource management in data centers and cloud environments. As service availability and performance optimization have become critical requirements, particularly in sectors like education, various studies have explored techniques for efficient operating system migration, performance impacts of virtualization, and advanced failure recovery mechanisms. This section reviews key contributions in these areas, focusing on real-time migration strategies, resource optimization, and the challenges posed by virtualization on system performance.

The migration of operating systems between different hosts is a powerful tool for data center and cluster administrators, allowing for clear separation of hardware and software, while also improving fault management and load distribution. This process enables the system to remain active, ensuring high availability during migration, minimizing downtime and service interruption. According to a study by [16] a downtime of less than 60ms was achieved. This work aims to demonstrate the migration of operating systems during system operation, using Xen VMM and adapting bandwidth dynamics to minimize the impact on running processes and reduce downtime.

Hardware virtualization, as practiced in data centers and web services, involves resource isolation for virtual machines, but this technique affects the performance of guest systems. According to a study by [17]changes in virtualization techniques (OpenVz, KVM, Xen, VirtualBox, VMware ESXi) affect CPU and memory performance. Experiments showed that the impact is similar for virtualization platforms when considering a network interface, but increasing the number of interfaces causes a throughput bottleneck, lower than in a non-virtualized system. Additionally, CPU and memory demands from the secondary guest are lower, but their impact on the virtualized system's performance is significant, highlighting the need for continuous monitoring and diagnosis of performance bottlenecks [18]

In a study by [19], failure recovery in virtualized environments is typically achieved by shutting down and restarting virtual machines on other nodes, causing significant service disruptions. This method presents particular challenges in educational institutions, where uninterrupted access to digital resources is essential. To address this issue, real-time migration techniques and optimization algorithms such as Perf+ have been proposed, significantly reducing downtime and improving resource efficiency. These innovations contribute to building more reliable and resilient infrastructures for digital learning.

Virtualization techniques are becoming increasingly popular in "Grid Computing" due to the ability to run multiple operating systems on a single host and provide a restricted execution environment. In a study by [20] the merits of four virtualization techniques are compared, and microbenchmarks for measuring scalability for various resources (CPU, disk, memory, network) are presented. The results show that techniques like Vserver and Xen offer better performance than VMware and UML (), but there is still a need for improvement, especially in network isolation between VMs. VMware provides high isolation performance, while Vserver and Xen offer greater memory savings but have limitations in isolation and communication between virtual machines. This study helps users choose the most suitable tools for their application needs.

According to [21], conventional failure recovery methods in virtualized environments typically involve shutting down virtual machines and restarting them on alternative nodes, leading to significant service disruptions. These disruptions are particularly critical in educational settings, where uninterrupted access to digital resources is essential. To address this, is developed a real-time management algorithm integrated into Heartbeat for high-availability clusters, enabling live migration of virtual machines and minimizing downtime. Furthermore, the Perf+ algorithm, utilizing the Hash-MD5 technique, was introduced to optimize resource consumption and ensure near-seamless continuity of service during both controlled maintenance and unexpected node failures.

The main motivation for enterprises using virtualization is the creation of a faster and more dynamic infrastructure, reducing costs and improving the response to changes in business conditions. However, managing virtualized environments requires new techniques, such as dynamic migration of virtual machines and proper resource planning, which is often poorly understood. In a study by [22] the impact of three CPU schedulers for virtual machines and how they affect application performance was evaluated. It was shown that the CPU allocation algorithm and its parameters can have a significant impact, especially for I/O-intensive applications. For this, several performance and CPU allocation error benchmarks were used.

3 Methodology

This study adopts a combined experimental and quantitative statistical approach to provide a comprehensive and structured understanding of how various operational factors influence the Service Level Agreement (SLA) within virtualized environments. The primary aim is to generate meaningful insights and formulate practical recommendations that can assist system administrators, IT managers, and cloud service providers in optimizing service performance and enhancing reliability, particularly in infrastructures based on the Xen hypervisor. To create a realistic testing scenario, a controlled environment was established where multiple virtual machines managed by the Xen hypervisor were deployed [19]. A specific number of these virtual machines were deliberately deactivated during the experiment to simulate real-world failure events and assess their potential impact on service availability, operational continuity, and overall system performance. Alongside these simulations, controlled operational loads were systematically introduced, enabling a detailed evaluation of how variations in key factors such as service downtime, CPU utilization, network bandwidth, and data loss affect the quality and reliability of services. All experimental data were carefully recorded in a structured manner, ensuring the consistency and integrity of the dataset. SLA was continuously monitored and measured as the dependent variable, serving as a comprehensive indicator of service performance under different operational conditions. The methodological framework of the study is grounded in a quantitative analytical approach that seeks to explore both the strength and nature of the relationships between SLA and the selected independent variables. Initially, a correlation analysis was performed to determine the direction (positive or negative) and magnitude of the relationships between SLA and each operational factor. Following this, a multiple linear regression analysis was conducted to examine the combined linear effects of all independent variables on SLA and to assess the individual statistical significance of each factor. Several essential statistical indicators were calculated as part of this analysis, including the multiple correlation coefficient (R), the coefficient of determination (R^2), the adjusted R^2 to account for model complexity, the standard error of the estimate to gauge prediction accuracy, and p-values to test the significance of the overall model and each predictor individually.

This comprehensive methodology not only allows for the identification of the most critical factors influencing SLA but also provides a solid foundation for the development of proactive strategies in system management, capacity planning, and performance optimization. By applying these insights, organizations operating virtualized environments with Xen can achieve higher levels of service reliability, minimize downtime, and deliver enhanced quality of service to end-users, even under conditions of increased operational stress or partial system failures. The structured experimental approach further ensures that the findings are robust, replicable, and applicable to a broad range of real-world cloud and data center environments.

3.1 Data Collection

To empirically assess the impact of key system parameters on the performance of the Service Level Agreement (SLA) in a virtualized environment, a controlled testing setup was created using Xen as the virtualization platform. Initially, an infrastructure resembling a cloud environment was simulated by deploying a varying number of virtual machines (VMs), all managed by the Xen hypervisor.

In the second phase of the experiment, a selected number of these virtual machines were deliberately deactivated to simulate partial system failure scenarios or service interruptions [19]. These interventions aimed to reflect real-world unexpected events that could affect the functioning of virtualized systems and enabled the analysis of their impact on service availability and quality, as reflected in the measured SLA values.

Following this setup, controlled operational loads were generated to independently and systematically test the impact of four operational independent variables:

- service downtime, representing the duration in which services were unavailable;
- CPU performance, measured by the processing power allocated to each virtual machine;
- network bandwidth, referring to the data transmission capacity between virtual units;
- data loss, defined as the percentage of lost or corrupted data during transmission across the network.

Each of these variables was independently manipulated across distinct test scenarios, with their values adjusted in a controlled manner to observe their direct effect on SLA. For every configuration, data was carefully recorded in a structured format, capturing the values of each independent variable along with the corresponding SLA measurement, which represented the overall service quality and reliability for that specific setup. This chronologically and scientifically structured approach ensured the collection of a consistent and reliable dataset suitable for further statistical analysis.

The data was collected in a structured and repeatable manner to ensure the integrity of the statistical analysis. This dataset serves as the foundation for the further analysis presented in the following sections.

In this study, we aim to test whether operational factors such as the number of failed machines, downtime, CPU usage, bandwidth, and data loss have a statistically significant impact on the Service Level Agreement (SLA). If the result of the regression analysis is statistically significant (p-value < 0.05), we reject the null hypothesis (H_0) and accept that these factors do have an impact on SLA.

Alternative Hypothesis (H_1):

At least one of the variables (number of failed machines, downtime, CPU usage, bandwidth, or data loss) has a significant effect on SLA.

To validate this, we attempt to reject the following:

Null Hypothesis (H_0):

There is no statistically significant impact of the number of failed machines, downtime, CPU usage, bandwidth, or data loss on SLA.

3.2 Data Preparation and Analysis

The preparation and analysis of data involved the collection, cleaning, and organization of the selected variables in order to make them suitable for statistical analysis. Initially, the dataset was examined for missing values, outliers, and inaccuracies, which were corrected or excluded as necessary. Subsequently, the variables were normalized or transformed when required to meet the assumptions of linear regression, such as linearity, normality, and homoscedasticity. Finally, the processed data was used to construct the multiple linear regression model for testing the research hypotheses.

Below, in Table 1, the organized dataset used for the analysis is presented.

Table 1. Dataset.

Number of failed machines	Downtime	CPU usage (%)	Bandwidth usage (%)	Data loss (pages)	SLA (Service Level Agreement)
1	80	0.78	5.7	0	99.999997
2	140	0.85	6.6	0	99.999995
3	180	0.91	7.2	0	99.999993

4	210	0.98	7.8	1	99.999992
5	250	1.02	8.1	1	99.999989
6	310	1.08	8.7	1	99.999985
7	420	1.1	9	1	99.99998
8	580	1.105	9.5	1	99.999978
10	610	1.12	9.8	2	99.999973
12	640	1.18	10.2	2	99.999969
14	680	1.21	10.6	2	99.999964
16	720	1.26	11	3	99.99996
18	740	1.28	11.3	3	99.999957
20	770	1.3	11.6	3	99.999953
22	800	1.33	11.9	3	99.999949
25	830	1.35	12.2	4	99.999946
28	860	1.38	12.4	4	99.999943
31	890	1.41	12.7	4	99.999939
35	920	1.44	13	4	99.999935
39	950	1.46	13.2	5	99.999932
44	980	1.48	13.5	5	99.999929
49	1010	1.5	13.7	5	99.999925
55	1040	1.52	13.9	5	99.999922
62	1070	1.55	14.1	6	99.999919
70	1100	1.57	14.3	6	99.999916
79	1130	1.59	14.5	6	99.999913
89	1160	1.61	14.7	6	99.99991
100	1190	1.63	14.9	7	99.999907
112	1220	1.65	15.1	7	99.999904
125	1250	1.67	15.3	7	99.999901
139	1280	1.69	15.5	8	99.999898

The data represents information collected from a virtualized system using the XEN platform, focusing on its performance, workload, and reliability.

Table 2. Variables in the dataset.

Column Name	Data Type	Description
Number of failed machines	Numeric	Indicates the number of machines (or servers) that failed during a given period.
Downtime	Numeric	The total time (in minutes or hours) during which the system was out of service due to failures.
Cpu_usage %	Numeric	CPU usage expressed as a percentage (values from 0.0 to 1.0), representing the processor load.
Bandwidth %	Numeric	Network bandwidth usage, expressed as a percentage.
Data_loss (pages)	Numeric	Number of data pages lost during processing or transmission.
Service Level Agreement	Numeric	Percentage of compliance with the Service Level Agreement (SLA).

3.3 Role in the Model:

Independent Variables (Inputs / X):

These are the factors that may influence the system's performance (specifically, the SLA):

- Number of failed machines
- Downtime
- CPU usage (%)
- Bandwidth usage (%)

These represent the input data that directly or indirectly affect the quality of service.

Dependent Variable (Output / Y):

- SLA

This is the main element we aim to analyze in relation to the aforementioned factors. SLA serves as the primary indicator of service quality.

4 Results

Below, we present the correlation that exists between the variables, along with the calculated correlation coefficient. Correlation refers to the measurement of the degree, intensity, and direction of relationships and mutual connections between different phenomena. The correlation coefficient represents the strength and intensity of the relationship between phenomena. Correlations can be either linear or nonlinear, depending on the nature of the relationship. The correlation coefficient ranges from -1 to +1. When the coefficient is +1, there is a perfect positive correlation between the two variables. When the coefficient is -1, there is a perfect negative correlation. In reality, it is very rare to find correlation coefficient values of exactly +1 or -1.

Table 3. Correlations.

		Sla	Number of failed machines	Downtime ms	Cpu_usage	Bandwidth %	Data_loss
sla	Pearson Correlation	1	-.908**	-.982**	-.988**	-.986**	-.990**
	Sig. (2-tailed)		.000	.000	.000	.000	.000
	N	31	31	31	31	31	31

**. Correlation is significant at the 0.01 level (2-tailed).

The correlation table presents the statistical relationship between the dependent variable SLA and the independent variables: number of failed machines, downtime, CPU usage, bandwidth, and data loss. Below is a simplified explanation of each column:

- Pearson Correlation: shows the strength and direction of the relationship between the variables. The values range between -1 and +1:
  - ➢ Negative values (as in this case) indicate a negative correlation, meaning that as the value of a factor increases, the SLA value decreases.
  - ➢ The closer the value is to -1, the stronger the negative relationship.
- Sig. (2-tailed): represents the p-value. In this case, for all factors, the p-value is 0.000, which is smaller than the conventional threshold of 0.05. This indicates that the relationships are statistically significant.
- N: is the number of observations included in the analysis – in this case, 31 instances were used.

Interpretation

- ✓ Number of failed machines and SLA: The correlation is -0.908 → an increase in machine failures is associated with a significant drop in SLA.
- ✓ Downtime and SLA: A very strong negative correlation of -0.982 → as downtime increases, the quality of service (SLA) decreases significantly.
- ✓ CPU usage and SLA: A correlation of -0.988 → high CPU usage has a very strong negative impact on SLA.
- ✓ Bandwidth and SLA: A correlation of -0.986 → bandwidth issues are strongly associated with a decline in SLA.

- ✓ Data loss and SLA: A correlation of -0.990 → data loss has the strongest negative impact on SLA.

All operational factors show a strong and statistically significant negative correlation with SLA, supporting the alternative hypothesis (H_1) that these factors significantly influence system performance. This suggests that improving these factors could lead to a notable increase in service quality.

Table 4. Variables Entered/Removed[a].

Model	Variables Entered	Variables Removed	Method
1	Bandwidth, Number of failed machines, Downtime, Cpu_usage[b]	.	Enter

a. Dependent Variable: sla

b. All requested variables entered.

The "Variables Entered/Removed" table shows that all the planned independent variables have been included in the regression model: bandwidth, number of failed machines, downtime, and CPU usage, while no variable has been excluded. This was done using the "Enter" method, which means that all factors were entered simultaneously without any automatic selection. This approach ensures that the impact of each operational factor on the SLA is analyzed fully and directly. In conclusion, the model includes all the intended inputs, allowing for a comprehensive analysis of the relationship between system performance and service quality.

Table 5. ANOVA[a].

	Model	Sum of Squares	df	Mean Square	F	Sig.
1	Regression	.000	4	.000	635.207	.000[b]
	Residual	.000	26	.000		
	Total	.000	30			

a. Dependent Variable: sla

b. Predictors: (Constant), bandwidth, Number of failed machines, downtime, cpu_usage

The value of F = 635.207 is very high, indicating that the regression model effectively explains the variation in SLA.

Sig. (p-value) = 0.000, which is much smaller than 0.05 → this means that the regression model is statistically significant.

In other words: the null hypothesis (H_0) is rejected, and we accept that at least one of the independent variables significantly affects the SLA.

This result supports the alternative hypothesis (H_1) of the study — that operational factors have a significant impact on the SLA. Regression models are very valuable for this type of analysis, and this ANOVA test confirms the validity of your model.

Table 6. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.996[a]	.992	.990	.0000031031

a. Predictors: (Constant), data_loss, Number of failed machines, Downtime, Cpu_usage, Bandwidth

The multiple regression model that is developed demonstrates very strong and meaningful results. The value of R = 0.996 indicates a very close relationship between the independent variables and the SLA. This means that changes in service quality (SLA) are highly associated with variations in operational factors such as data loss, number of failed machines, downtime, CPU usage, and bandwidth.

The coefficient of determination, $R^2 = 0.992$, shows that 99.2% of the variation in SLA is explained by the variables included in the model. Depending on the level of R^2, it is common to categorize models into three groups:

- 0.8 – 1.0: a high-quality model;
- 0.5 – 0.8: an acceptable-quality model;
- 0.0 – 0.5: a low-quality model.

Even after adjusting for the number of variables in the model, the Adjusted R^2 remains very high at 0.990, indicating that the model is stable and not overfitted with unnecessary variables. The standard error of the estimate is exceptionally low, which means that the model's predictions are very accurate and closely match the actual values.

In conclusion, this is a highly successful model for explaining and predicting SLA based on key technical operational factors. However, it is important to note, as previously analyzed, that there is a high degree of multicollinearity among some of the variables. This may affect the reliability of interpreting the individual impact of each coefficient in the model.

5 Conclusions

Virtualization is one of the key technologies in modern IT architecture, as it enables more efficient use of computing resources by dividing a physical infrastructure into several independent virtual machines. This brings significant benefits in terms of cost reduction, scalability, and service flexibility. However, with the increasing complexity of virtualized environments, performance monitoring and management become more challenging, which in turn elevates the importance of evaluating the Service Level Agreement(SLA) as the primary indicator of service quality delivered to users.

The study of SLA is essential for understanding how well a system operates and how reliably it delivers uninterrupted and dependable services. In this context, analyzing the impact of operational factors such as the number of failed machines, downtime, CPU usage, bandwidth, and data loss on the SLA is crucial for improving resource management and ensuring sustainable performance in virtualized systems.

Based on the multiple linear regression analysis, the constructed model proved to be extremely powerful in explaining the variations in SLA. The very high value of the coefficient of determination ($R^2 = 0.992$) shows that over 99% of the variation in SLA is explained by the factors included in the model. This is further supported by the high F-statistic value and the very low p-value, both of which demonstrate the statistical significance of the model.

The model is highly effective and reliable for assessing and predicting service quality in virtualized systems.

6 Recommendations

Based on the findings and analysis of this study, several recommendations can be made to further enhance performance management and service quality in virtualized environments. It is advisable that future analyses apply advanced techniques such as Ridge regression, Lasso regression, or Principal Component Analysis (PCA) to address the issue of multicollinearity among independent variables, which would allow for a more precise interpretation of each factor's impact on the SLA [23]. Organizations should also implement automated mechanisms for the continuous, real-time monitoring of operational factors such as CPU usage, bandwidth, and data loss, enabling immediate intervention when deviations from performance standards are detected. Moreover, efforts should be prioritized on the factors that demonstrated the strongest negative impact on SLA, specifically data loss and the number of failed machines, in order to strengthen service reliability and quality [24]. Developing rapid recovery strategies and implementing redundancy for critical resources is also essential, as these measures can significantly minimize downtime and data loss, thereby supporting better SLA fulfillment. Finally, it is recommended that future studies expand the analysis by incorporating additional influencing factors such as network latency, input/output operations per second (IOPS), and the number of active users, all of which may further refine the understanding of SLA performance dynamics. These recommendations contribute not only to advancing scientific research but also to establishing a practical framework for managing virtualized infrastructures in real-world organizational settings.

References

1. Y. Mansouri and M. A. Babar, “A review of edge computing: Features and resource virtualization,” J Parallel Distrib Comput, vol. 150, 2021, doi: 10.1016/j.jpdc.2020.12.015.
2. Djordjevic, B., Timcenko, V., Kraljevic, N., & Macek, N. (2021). File System Performance Comparison in Full Hardware Virtualization with ESXi, KVM, Hyper-V and Xen Hypervisors. Advances in Electrical and Computer Engineering, 21(1). https://doi.org/10.4316/AECE.2021.01002
3. Gala, G., Fohler, G., Tummeltshammer, P., Resch, S., & Hametner, R. (2021). RT-Cloud: Virtualization Technologies and Cloud Computing for Railway Use-Case. Proceedings - 2021 IEEE 24th International Symposium on Real-Time Distributed Computing, ISORC 2021. https://doi.org/10.1109/ISORC52013.2021.00024
4. Alonso, S., Lázaro, J., Jiménez, J., Bidarte, U., & Muguira, L. (2021). Evaluating latency in multiprocessing embedded systems for the smart grid. Energies, 14(11). https://doi.org/10.3390/en14113322
5. Giallorenzo, S., Mauro, J., Poulsen, M. G., & Siroky, F. (2021). Virtualization Costs: Benchmarking Containers and Virtual Machines Against Bare-Metal. SN Computer Science, 2(5). https://doi.org/10.1007/s42979-021-00781-8
6. Abdulraheem, W. K. (2022). Performance Comparison of Xen AND Hyper-V in Cloud Computing While Using Cryptosystems. International Journal of Advances in Soft Computing and Its Applications, 14(3). https://doi.org/10.15849/IJASCA.221128.02
7. R.Kennady , E. al. (2023). An Embedded Approach to Hypervisor-Oriented Interruption Virtualization Operation. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4). https://doi.org/10.17762/ijritcc.v11i4.9846
8. Stevanato, A., Biondi, A., Biasci, A., & Morelli, B. (2023). Virtualized DDS Communication for Multi-Domain Systems: Architecture and Performance Evaluation of Design Alternatives. Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS, 2023-May. https://doi.org/10.1109/RTAS58335.2023.00032
9. N. Almurisi and S. Tadisetty, “Cloud-based virtualization environment for IoT-based WSN: solutions, approaches and challenges,” J Ambient Intell Humaniz Comput, vol. 13, no. 10, 2022, doi: 10.1007/s12652-021-03515-z.
10. Parra, P., Da Silva, A., Losa, B., García, J. I., Polo, Ó. R., Martínez, A., & Sánchez, S. (2023). Tailor-made Virtualization Monitor Design for CPU Virtualization on LEON Processors. ACM Transactions on Embedded Computing Systems, 22(4). https://doi.org/10.1145/3584702

11. Mauricio, L., & Rubinstein, M. (2023). A Network Function Virtualization Architecture for Automatic and Efficient Detection and Mitigation against Web Application Malware. Journal of Internet Services and Applications, 14(1). https://doi.org/10.5753/jisa.2023.2847
12. B. Schulz and B. Annighofer, “Evaluation of Adaptive Partitioning and Real-Time Capability for Virtualization with Xen Hypervisor,” IEEE Trans Aerosp Electron Syst, vol. 58, no. 1, 2022, doi: 10.1109/TAES.2021.3104941.
13. S. B. Oh and J. H. Kim, “An Analysis on Interrupt Latency of Hypervisor for Automotive Software Integration,” Transactions of the Korean Society of Automotive Engineers, vol. 30, no. 11, 2022, doi: 10.7467/KSAE.2022.30.11.901.
14. C. Li, S. Xi, C. Lu, R. Guerin, and C. D. Gill, “Virtualization-Aware Traffic Control for Soft Real-Time Network Traffic on Xen,” IEEE/ACM Transactions on Networking, vol. 30, no. 1, 2022, doi: 10.1109/TNET.2021.3114055.
15. C. M. Zheng, X. X. Yao, F. Zhou, X. F. Zheng, X. J. Yang, and R. Dai, “Design and implementation of a cloud server based on hardware virtualization,” Gongcheng Kexue Xuebao/Chinese Journal of Engineering, vol. 44, no. 11, 2022, doi: 10.13374/j.issn2095-9389.2022.01.12.005.
16. V. Dakić, M. Kovač, and J. Slovinac, “Evolving High-Performance Computing Data Centers with Kubernetes, Performance Analysis, and Dynamic Workload Placement Based on Machine Learning Scheduling,” Electronics (Basel), vol. 13, no. 13, p. 2651, Jul. 2024, doi: 10.3390/electronics13132651.
17. A. Rehaimi, Y. Sadqi, Y. Maleh, G. S. Gaba, and A. Gurtov, “Towards a federated and hybrid cloud computing environment for sustainable and effective provisioning of cyber security virtual laboratories,” Expert Syst Appl, vol. 252, p. 124267, Oct. 2024, doi: 10.1016/j.eswa.2024.124267.
18. A. Belkhiri and M. Dagenais, “Analyzing GPU Performance in Virtualized Environments: A Case Study,” Future Internet, vol. 16, no. 3, p. 72, Feb. 2024, doi: 10.3390/fi16030072.
19. E. Tabaku and E. Duçi, “Optimizing High Availability in Educational Systems Using Xen Paravirtualization,” Journal of Educational and Social Research, vol. 15, no. 2, p. 205, Mar. 2025, doi: 10.36941/jesr-2025-0054.
20. J. Hu et al., “dpBento: Benchmarking DPUs for Data Processing,” Apr. 2025. https://doi.org/10.48550/arXiv.2504.05536
21. E. Tabaku, “Improving High Availability Services Using KVM Full Virtualization,” European Journal of Computer Science and Information Technology, vol. 13, no. 1, pp. 1–15, Jan. 2025, doi: 10.37745/ejcsit.2013/vol13n1115.
22. A. Ghasemi, A. Toroghi Haghighat, and A. Keshavarzi, “Enhancing virtual machine placement efficiency in cloud data centers: a hybrid approach using multi-objective reinforcement learning and clustering strategies,” Computing, vol. 106, no. 9, pp. 2897–2922, Sep. 2024, doi: 10.1007/s00607-024-01311-z.
23. S. Yang, F. Li, R. Yahyapour, and X. Fu, “Delay-Sensitive and Availability-Aware Virtual Network Function Scheduling for NFV,” IEEE Trans Serv Comput, vol. 15, no. 1, 2022, doi: 10.1109/TSC.2019.2927339.
24. M. Imran, M. Ibrahim, M. S. U. Din, M. A. U. Rehman, and B. S. Kim, “Live virtual machine migration: A survey, research challenges, and future directions,” Computers and Electrical Engineering, vol. 103, 2022, doi: 10.1016/j.compeleceng.2022.108297.

18. Implementation of ROC curves, bootstrap and logistic regression in cardiovascular research

Lorena Zeqo[1] and Eljona Tasho[2]

[1,2] Fan S. Noli University, Korca, Albania
lmargo@unkorce.edu.al[1], emilo@unkorce.edu.al[2]

Abstract. Nowadays advances in Computational Statistics have made possible to use statistics in many areas. Bootstrapping is becoming very useful and popular in wide areas of research, and also in other statistical applications such as logistic regression.

A ROC (Receiver operating characteristics) curve is a plot that depicts the trade-off between the sensitivity and 1- specificity across a series of cut-off points when the diagnostic test is continuous or on ordinal scale. The AUC (area under ROC curve) is used for the performance of the classifiers and it performs well as a general measure to predict accuracy.

In our study we will use medical records in cardiology to perform a multiple logistic regression to identify risk factors for stent thrombosis. Bootstrap techniques are applied with intention to develop a better use and to implement these techniques in practice and we will also use ROC curves as a diagnostic tool for the performance of the logistic regression to determine the model fit.

Keywords: Statistics, Bootstrap, Logistic regression, ROC curves, Cardiology, AUC

1 Introduction

Recent researches in cardiology require implementing advanced statistical knowledge and the collaboration between statisticians and researchers from medicine is associated with important results. Using statistical software packages for statistical analysis is more convenient recently and it contributes in conducting a fair analysis. There are some good statistical software available such as SPSS that enable to conduct the statistical analysis of the data.

Stent thrombosis, also known as abrupt vessel closure, occurs when an implanted coronary stent causes a thrombotic occlusion and can cause serious complications in the patients (Modi et. al (2022)). The study of stent thrombosis and risk factors related is very important and essential in cardiology. For this reason, we implemented Logistic Regression, ROC Analysis and also Bootstrap to identify which are the risk factors in Stent thrombosis of the studied patients.

Logistic regression model is used to construct a model with dependent variable stent thrombosis and twelve independent variables; bootstrap is used to construct and compare confidence intervals for the logistic regression parameters and ROC curves are used as a diagnostic tool for logistic regression performance. In our previous studies important results are obtained in using bootstrap in logistic regression model (Zeqo et. al (2022), Zeqo & Cobani (2022)) and in using ROC curves in logistic regression (Zeqo et. al (2021), Zeqo (2022)) but in this study we extended the study sample and we implemented logistic regression, bootstrap and ROC curves together discovering significant risk factors in stent thrombosis, comparing the results using different bootstrap confidence intervals for logistic regression coefficients and using AUC as a measure for the model fit.

2 Methodology

2.1 Data in The Study and Objectives

The population of interest in this study include the patients undergoing coronary intervention procedures at the Heart Catheterization Center in University Medical Center of Tirana "Mother Teresa" Albania (during the years 2012-2019). The data records used include a total of 5969 patients with median of the age 66 years old (35-87) with mean 64.81 and standard deviation 7.717. The following variables are included in this study: Gender, Age, Coronary Artery Disease (SAK) (includes three types of diseases), Stent (BMS/ DES), Diabetes Mellitus (Yes/No), Ejection Fraction less than 40 (Yes/No), Body Mass Index greater than 30 (Yes/ No), Smoking (Yes/No), Dyslipidemia (Yes/No), Post Myocardial Infarction (PMI) (Yes/No), Dissection (Yes/No) and Arterial Hypertension (HTA) (Yes/No). Based in some international studies, the major part of these variables are considered as risk factors for heart failure so we are interested to construct a logistic regression model and to investigate the impact of these factors in the stent thrombosis of the patients studied.

In this paper we aim to implement bootstrap and also ROC curves in the logistic regression model to identify the risk factors for stent thrombosis of the patients with cardiovascular diseases. It is an important task to use advanced statistical elements in research in medical science and advances in technology and computational methods in statistics that enable to obtain significant and important results in these fields.

This data analysis was conducted using IBM SPSS, Version 20.

3 Logistic Regression, Roc Curves and Bootstrap in Cardiology

Many international studies are focused in examining the occurrence and the prevalence regarding cardiovascular diseases and determining risk factors related. The construction of a logistic regression model is very useful if the dependent variable is binary (Hosmer and Lemeshow (2000)). Studies regarding Stent thrombosis and using multivariate logistic regression to determine the risk factors includes De Servi et. al. (1999), Modi et.al (2022) etc.

Bootstrap and logistic regression has become object of many studies (Adjei and Karim (2016), Hossain and Abdullah (2004)). Because of the lack to fulfill all the required assumptions in using traditional methods for statistical inference or in the case of small samples, a useful successful alternative is often used by researchers: the bootstrap (Efron (1979), Davidson and Hinkley (1997)). One of the main reasons for using bootstrap in interval estimation is because enable determining confidence intervals on parameters without having to make unreasonable assumptions (See URL [20]). Percentile Bootstrap or Bias Corrected-Accelerated Bootstrap Confidence Intervals (BCa) are used regarding if the bootstrap distribution is symmetric or acceleration parameter and bias-correction factor are considered (URL: [21], [22]). Adjei & Karim (2016) used bootstrap in the logistic regression model and Capodanno et. al (2009) combined logistic regression and bootstrap to determine the risk of stent thrombosis after intervention.

The Area under the ROC Curve (AUC), together with other goodness of fit, can be used as an overall measure of fit of a logistic regression model (See Hanley and McNeil (1982), Kumar and Indrayan (2011), Melo (2013), URL: [18], [19], [23], [24]). ROC Curves and logistic regression were used in predicting early stent thrombosis after intervention (Kumar et al (2020)), but in case of one of the predictors and the dependent variable. Zeqo et. al (2021, 2022) used logistic regression model, bootstrap and ROC Curves as a way to determine the overall fit of the model in studying the survival of the hospitalized patients with cardiovascular problems and also the risk factors for stent thrombosis. In this paper, Logistic regression, Bootstrap and ROC curves are implemented together for a better analysis in determining the risk factors for the stent thrombosis of the patients hospitalized including more patients and not the same independent variables than in previous studies.

4 Results and Discussion

In this study with data from 5969 patients with cardiac problems hospitalized in a hospital in Tirana, Albania, we considered the following variables: Gender, Age, Coronary Artery Disease (SAK), Stent, Diabetes Mellitus, Ejection Fraction (less than 40), Body Mass Index (greater than 30), Smoking, Dyslipidemia, Post Myocardial

Infarction (PMI), Dissection and Arterial Hypertension (HTA) and we used SPSS to build a logistic regression model that determines the impact of these variables in stent thrombosis of the patients and ROC Curves as a way to determine the model fit. The results are compared using two bootstrap confidence intervals for the model coefficients: Percentile Bootstrap and Bias Corrected Accelerated. The dependent variable is Stent Thrombosis with value 1 that indicates the patients with stent thrombosis and the value 0 that indicates the patients without stent thrombosis. Only 9 of the independent variables considered in the model resulted significant and we marked red these significant variables (See Table 1). Using SPSS, we obtained the following results: The Omnibus Tests of Model Coefficients give a p-value 0 for our model, Cox & Snell R Square value is 0.042 and Nagelkerke R Square is 0.226. Hosmer and Lemeshow Test gives a Chi-Square value 7.043 and a p-value 0.532 and the Classification table gives us an Overall Percentage of 97.9. These demonstrate that we have obtained a good model.

Table 1. Logistic Regression Model Coefficients, Standard Error, Significance and Odd Ratios.

	Coefficients	**S.E**	**Sig.**	**Odd Ratios**
Gender	-0.093	0.233	0.691	.911
Age	0.016	0.013	0.245	1.016
SAK			.000	
SAK(type 2)	1.699	0.293	.000	5.466
SAK(type 3)	2.776	0.306	.000	16.062
STENT (DES)	1.299	0.228	.000	3.667
EF<40	0.161	0.193	.406	1.175
DIABETES	-0.536	0.210	0.011	.585
SMOKING	1.114	0.304	.000	3.047
DYSLIPID.	-1.394	0.262	.000	.248
BMI>30	0.712	0.255	0.005	2.037
PostIM	0.887	0.216	.000	2.429
HTA	-0.899	0.236	.000	.407
DISSEC	1.717	0.393	.000	5.567
Constant	-6.670	0.955	.000	0.001

From the above results, we have the following significant factors in the model: Coronary Artery Disease, Stent, Diabetes Mellitus, Smoking, Dyslipidemia, Body Mass Index, Post Myocardial Infarction, Arterial Hypertension and Dissection (the p-value in the corresponding column in each table is less than 0.05).

Comparing the results with previous studies about the risk factors for stent thrombosis (Zeqo et. al (2022), Zeqo (2022), although the data sample and the independent variables were not the same, some of the independent variables resulted in all these studies as risk factors for the stent thrombosis: Coronary Artery Disease, Smoking, Dyslipidemia, Post Myocardial Infarction and Arterial Hypertension.

Using Bootstrap with 1000 replications in Logistic Regression in this study gives us the same significant coefficients:

Table 2. Bootstrap for variables in the Equation (Percentile Bootstrap C.I)

	Coefficients	**Bias**	**S.E**	**Sig.**
Gender	-0.093	0.009	0.253	0.691
Age	0.016	.000	0.014	0.263
SAK(type 2)	1.699	0.025	0.362	0.001
SAK(type 3)	2.776	0.049	0.381	0.001
STENT(DES)	1.299	0.015	0.245	0.001
EF<40	0.161	-0.006	0.214	0.439
DIABETES	-0.536	-0.003	0.237	0.023
SMOKING	1.114	0.043	0.370	0.003
DYSLIPID.	-1.394	-0.013	0.275	0.001
BMI>30	0.712	0.033	0.271	0.006
PostIM	0.887	-0.006	0.225	0.001
HTA	-0.899	-0.009	0.276	0.003
DISSEC	1.717	-0.023	0.394	0.001
Constant	-6.670	-0.130	0.941	0.001

Table 3. Bootstrap for variables in the Equation (Bias Corrected and Accelerated Bootstrap C.I).

	Coefficients	**Bias**	**S.E**	**Sig.**
Gender	-0.093	0.016	0.243	0.692
Age	0.016	.000	0.014	0.250
SAK(type 2)	1.699	0.019	0.355	0.001
SAK(type 3)	2.776	0.025	0.364	0.001
STENT(DES)	1.299	0.015	0.253	0.001
EF<40	0.161	-0.009	0.206	0.419
DIABETES	-0.536	0.002	0.232	0.015
SMOKING	1.114	0.038	0.370	0.002
DYSLIPID.	-1.394	-0.013	0.296	0.001
BMI>30	0.712	0.017	0.272	0.006
PostIM	0.887	0.004	0.230	0.001
HTA	-0.899	0.019	0.269	0.001
DISSEC	1.717	-0.023	0.407	0.001
Constant	-6.670	-0.121	0.966	0.001

A comparison between Percentile Bootstrap and Bias Corrected Accelerated Bootstrap Confidence Intervals regarding Bias and Standard Error for the logistic regression coefficients is as follows:

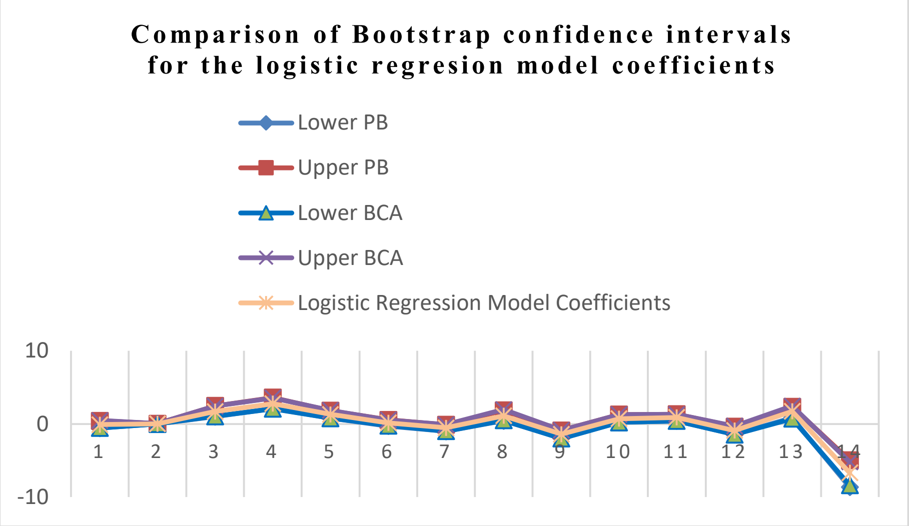


Fig. 9. Comparing Percentile Bootstrap (PB) and Bias Corrected Accelerated Bootstrap (BCA) Confidence intervals for the coefficients of the Logistic Regression model.

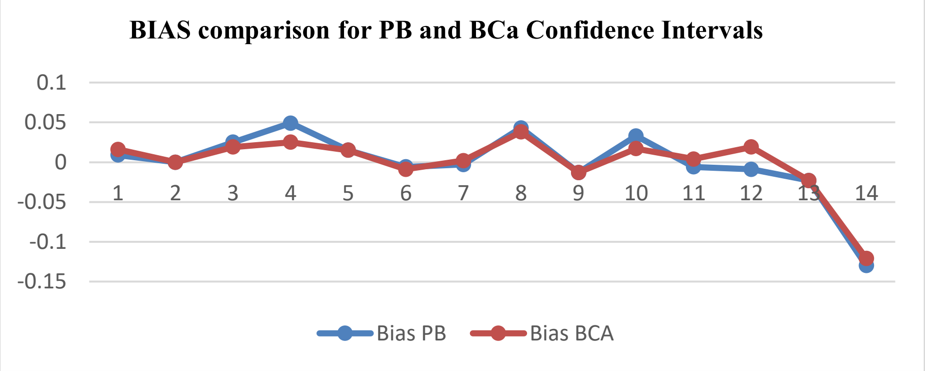


Fig. 2. Comparison of Bias using Percentile Bootstrap (PB) and Bias Corrected Accelerated Bootstrap (BCA).

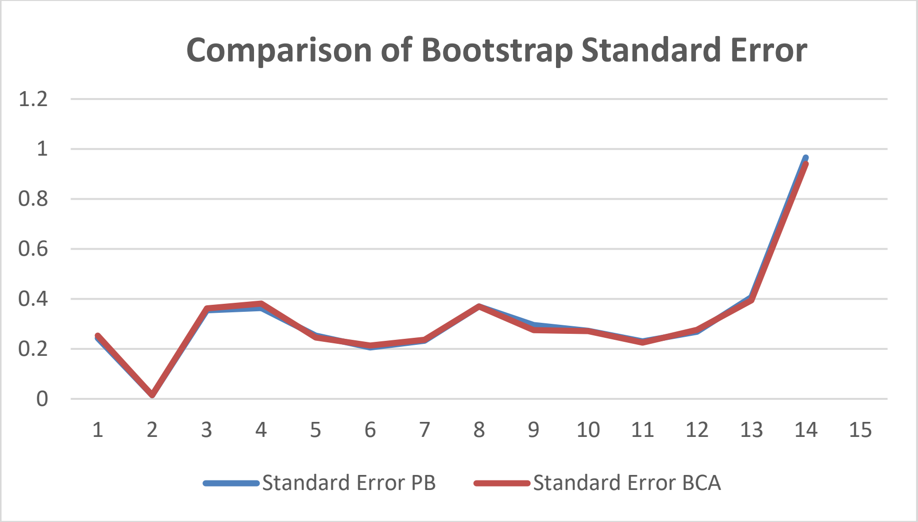


Fig. 3. Comparing Bootstrap Standard Error using Percentile Bootstrap and Bias Corrected and Accelerated Bootstrap.

ROC analysis is used for the logistic regression model fit. The ROC Curve, the Area under the curve, Standard Error, Significance and 95% Confidence Interval for the AUC are given as follows:

Table 4. Area Under the Curve (AUC).

Area	S.E	Sig.	95% Confidence Interval	
			Lower Bound	Upper Bound
0.842	0.017	.000	0.808	0.875

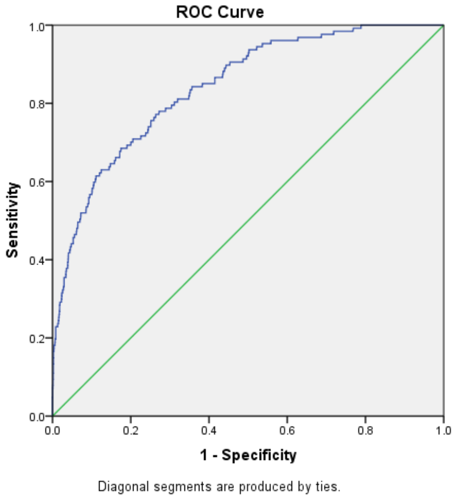


Fig. 4. ROC Curve for the Logistic Regression model obtained.

Based on the values of odd ratios we can say a patient with the third type of Coronary Artery Disease is more likely to have stent thrombosis than a patient with the first and the second type of disease. A smoker patient is more likely to have stent thrombosis than a non-smoker patient. A patient with dyslipidemia or Diabetes Mellitus is not more likely to have stent thrombosis compared with one with no Dyslipidemia or no Diabetes Mellitus respectively. A patient with BMI greater than 30 is more likely to have stent thrombosis than one with BMI less ore equal to 30 and also a patient with DES stent is more likely to likely to have stent Thrombosis compared with one with BMS stent. A post myocardial infarction patient is more likely to have stent thrombosis than a non-post myocardial infarction one. A patient with arterial hypertension is not more likely than a patient without hypertension to have stent thrombosis. Also a patient with aorta dissection is more likely to have stent thrombosis than a patient without aorta dissection (See the Odd Ratios in Table 4).

Based on the results the following variables: Coronary Artery Disease, Stent, Smoking, Body Mass Index, Post Myocardial Infarction and Dissection are associated with risk for Stent Thrombosis.

ROC analysis used in our study indicates a good model fit for the data. The Area under the Curve is 0.842 which is a good value, with 95% confidence interval:]0.808; 0.875[and the AUC is significantly different from 0.5. This means that this logistic regression built classifies better than by chance.

5 Conclusions

A data sample from hospitalized patients with cardiac problems was considered with intention to identify the risk factors for stent thrombosis and a multiple logistic regression model with 12 variables in the model was built. Nine factors resulted significant in the model: Coronary Artery Disease, Stent, Diabetes Mellitus, Smoking, Dyslipidemia, Body Mass Index, Post Myocardial Infarction, Arterial Hypertension and Dissection. The Goodness of fit measures used indicates a good model. The Omnibus Tests of Model Coefficients indicates a significant model, Nagelkerke R square is acceptable, Hosmer and Lemeshow Test for the model fit indicates that the model fits the data and The Classification Table gives values higher than 80% (in our model this value is 97.9%).

We also obtained good results from estimating the model coefficients using bootstrap. By using both Percentile Bootstrap and Bias Corrected Accelerated Bootstrap Confidence Intervals the results obtained were similar. There is a small difference in bias and standard error of estimations using both methods.

Based on the value of AUC from ROC Analysis we conclude we have a good model fit.

Acknowledgments

Special thanks to the cardiologist Dr. Juliana Karanxha for the consecutive collaborations in this area.

References

1. Adjei, I and Karim, R.: An Application of Bootstrapping in Logistic Regression Model, Open Access Library Journal, 3, 1-9 (2016).
2. Bertail P., Clemençon S., Vayatis N.: On bootstrapping the ROC Curve, Conference: Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Canada, pp 137-144 (2008).
3. Capodanno D, Capranzano P, Bucalo R, Sanfilippo A, Ruperto C, Caggegi A, Ussia G, Galassi AR, Tamburino C. A novel approach to define risk of stent thrombosis after percutaneous coronary intervention with drug-eluting stents: the DERIVATION score. Clin Res Cardiol. 98(4):240-8 (2009).
4. Davidson C., Hinkley D.V.: Bootstrap Methods and their Application. Cambridge: Cambridge University Press. (1997)
5. De Servi S, Repetto S, Klugmann S, Bossi I, Colombo A, Piva R, Giommi L, Bartorelli A, Fontanelli A, Mariani G, Klersy C. Stent thrombosis: incidence and related factors in the R.I.S.E. Registry(Registro Impianto Stent Endocoronarico). Catheter Cardiovasc Interv.46(1):13-8 (1999).

6. Efron B., Bootstrap Methods: Another Look at the Jackknife, Ann. Statist. 7 (1) pp 1 – 26 (1979).
7. Hanley, J. and McNeil, B.: The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. Radiology, 143, 29-36(1982).
8. Hosmer DW, Lemeshow S: Applied Logistic Regression, 2nd Ed. Chapter 5, John Wiley and Sons, New York, NY, pp. 160-164 (2000).
9. Hossain A. Abdullah H. T.: Nonparametric bootstrapping for multiple logistic regression model using R, BRAC University Journal, vol. I, no. 2, pp. 109-113 (2004).
10. Kumar R., Indrayan A.: Receiver operating characteristic (ROC) curve for medical researchers, Indian Pediatrics, 48 (4), pp 277-287 (2011).
11. Kumar, Rajesh; Tariq, Sahar; Fatima, Madiha; Saghir, Tahir; Batra, Mahesh Kumar; Karim, Musa; Sial, Jawaid Akbar; Khan, Naveedullah; and Rizvi, Syed Nadeem Hasan:Validity of the Stent Thrombosis Risk Score in Predicting Early Stent Thrombosis after Primary Percutaneous Coronary Intervention, Journal of the Saudi Heart Association: Vol. 32 : Iss. 2 , Article 19 (2020).
12. Melo F.:Area under the ROC Curve. In: Dubitzky W., Wolkenhauer O., Cho KH., Yokota H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY (2013).
13. Modi K, Soos MP, Mahajan K. Stent Thrombosis. In: StatPearls. NCBI Bookshelf version. StatPearls Publishing (2022).
14. Zeqo L, Tasho E, Karanxha J: Bootstrapping the coefficients of multiple logistic regression model in medicine data, Asian-European Journal of Mathematics, World Scientific Publishing Company, Vol.15, No 10 (2022), (2250248)
15. Zeqo L: ROC curves and logistic regression in cardiology data, Twenty-Fourth International Conference on “Social and Natural Sciences- Global Challenge 2022” (ICSNS XXIV- 2022), Barcelona, Spain, pp. 50-57 (2022).
16. Zeqo L. & Çobani S.: An application of bootstrap in logistic regression model in cardiology data, Twenty- Second International Conference on: "Social and Natural Sciences-Global challenge 2022" (ICSNS XXII-2022), Belgium, Book of Proceedings, pp. 96-106 (2022).
17. Zeqo L., Tasho E., Karanxha J. (2021), ROC curve as a key statistical tool in specific research areas, The Second International Conference “Research, Applications and Educational Methods”, 11-12 September 2021, Korçë, Shqipëri, Book of proceedings, 387-396, E-ISBN 9789928-4731-2-7 (Online).
18. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
19. https://www.medicine.mcgill.ca/epidemiology/joseph/courses/epib-621/logfit.pdf
20. https://www2.stat.duke.edu/~banks/111-lectures.dir/lect13.pdf

21. https://info.montgomerycollege.edu/_documents/faculty/maronne/math117/book-lock/ppt/lock3-4.pdf
22. http://users.stat.umn.edu/~helwig/notes/bootci-Notes.pdf
23. https://towardsdatascience.com
24. https://datatab.net/tutorial/roc-curve

19. Analysing Behavioural Intentions in FinTech Adoption: A PLS-SEM Approach

Evgjeni Xhafaj[1], Robert Kosova[2], and Neime Gjikaj[3]

[1]Department of Mathematical Engineering, Polytechnic University of Tirana, Albania
[2,3] Department of Mathematics, University Aleksander Moisiu, Durres 2001, Albania
evaxhafaj@gmail.com, robertkosova@uamd.edu.al,
neimegjikaj@uamd.edu.al

Abstract. Financial Technology (FinTech) service providers in Albania face numerous challenges in encouraging customers to integrate FinTech solutions into their daily financial activities. Key obstacles include limited awareness, regulatory hurdles, data privacy concerns, and cybersecurity risks. This study seeks to explore the critical factors that either hinder or facilitate the adoption of FinTech services. To achieve this, a conceptual framework was developed using the Technology Acceptance Model (TAM). Data were gathered through an web based survey and a self-administered questionnaire, targeting 150 FinTech users in Albania. The collected data were analysed using Partial Least Squares Structural Equation Modelling (PLS-SEM) via Smart PLS 3.2.9. The findings highlight that Perceived Usefulness (PU), Subjective Norm (SN), and Perceived Ease of Use (PEOU) significantly and positively influence users' behavioural intention to adopt FinTech solutions. These insights provide actionable recommendations for FinTech stakeholders, policymakers, and researchers to foster a supportive ecosystem that promotes the widespread adoption and effective use of FinTech services.

Keywords: FinTech, PLS-SEM, TAM

1 Introduction

The term FinTech originates from the combination of the English words "Finance" and "Technology and refers to startups and companies that harness the latest technologies to deliver innovative financial solutions [1]. FinTech, described as financial innovation driven by technology, has the potential to create new business models, applications, processes, and products that significantly impact financial markets, institutions, and the provision of financial services. It has emerged as a driving force in transforming the global financial landscape. [2]. This transformation is largely driven by the integration of advanced digital technologies within the Fintech ecosystem. Fintech incorporates tools such as data analytics and sophisticated advisory platforms to enhance service delivery and decision-making [3]. Fintech lenders, for instance, utilise large datasets along with artificial intelligence (AI) and machine learning (ML) algorithms to streamline credit assessments, enabling near-instantaneous loan decisions . Through these innovations, Fintech firms have gained a competitive edge over traditional financial institutions, offering greater efficiency, faster services, and superior user experiences [4].

Recently, a broad range of FinTech services—such as digital payments, cryptocurrencies, smart contracts, Insurtech, RegTech, robo-advisors, cybersecurity solutions, online banking, and e-commerce—have become increasingly promoted and accessible to consumers across various sectors, including banking, capital markets, insurance, blockchain enterprises, and retail businesses [5]

This study presents an integrated model incorporating the constructs of Perceived Usefulness (PU), Perceived Ease of Use (PEOU), and Subjective Norm (SN) to examine their influence on behavioural intention toward the adoption of FinTech services. The novelty of this research lies in its contextual originality, as this specific combination of constructs has not yet been explored within the Albanian setting.

2 Literature review

Several empirical studies have examined the determinants influencing consumer behavioural intentions towards the adoption of FinTech services, particularly in the context of developing countries and emerging digital markets.

[6] conducted a study focusing on consumer behavioural intentions toward FinTech services in Saudi Arabia. The research employed the Unified Theory of Acceptance and Use of Technology as a foundational model, which was extended by incorporating two additional constructs: privacy enablers and privacy inhibitors. To empirically test the proposed hypotheses, the authors used PLS-SEM based on data collected from 361 FinTech users residing in Saudi Arabia. The findings revealed that performance expectancy, effort expectancy, facilitating conditions, and privacy enablers had significant and positive effects on users' behavioural intentions to adopt FinTech services. Conversely, social influence and privacy inhibitors were found to have an insignificant impact on behavioural intention in this context.

In a related study, [7] explored the role of FinTech applications in enhancing financial resilience during the COVID-19 pandemic in Jordan. This research aimed to identify the factors influencing Jordanian citizens' intention to use FinTech services in times of crisis. Drawing on a conceptual model comprising five hypotheses, the study utilised PLS-SEM to analyse responses from a sample of 500 potential FinTech users. The results demonstrated that perceived benefits and social norms significantly influenced the intention to use FinTech applications, highlighting the importance of both utility and social context in driving adoption during periods of economic uncertainty.

Customers' decisions to adopt new technologies, particularly in the era of social media, are significantly shaped by the opinions and behaviours of those around them. Influences from family, friends, and colleagues often serve as key sources of positive recommendations that can encourage individuals to embrace emerging technologies [8], [9]. This phenomenon is captured by the concept of subjective norms, also referred to as social influence or image, which is defined as "the extent to which an individual perceives that important others believe he or she should apply the new system" (Venkatesh et al., 2003).

Empirical evidence from various cultural and geographical contexts underscores the strong link between social influence and users' behavioural intentions to adopt FinTech solutions. For instance, [10], in a study conducted in China, researchers found that social influence significantly predicts behavioural intention towards the use of FinTech platforms. Similarly, an empirical study conducted in Jordan reached the same conclusion, affirming that social influence is a strong determinant in shaping consumers' intentions to use FinTech services in the Jordanian context [7]. Based on the above arguments, the following hypothesis is proposed:

H1: Subjective norm has a significant and positive effect on behavioural intention to adopt FinTech services.

Perceived ease of use (PEOU) is defined as "the degree of ease associated with using the system." In the context of FinTech adoption, it refers to the user's perception of how effortless it is to utilise FinTech services. This construct is widely recognised as a key determinant of technology adoption in various research settings [11].

In the context of FinTech, PEOU plays a critical role, as customers often perceive such platforms as user-friendly and straightforward. Empirical research consistently demonstrates that higher levels of PEOU are associated with stronger behavioural intentions to adopt new technologies [12],[13].

Specifically, when customers believe that using a FinTech application is convenient and requires minimal effort, they are more likely to accept and use the service. This is particularly important for FinTech services, where users are expected to independently manage transactions. Supporting this perspective, [14] developed a conceptual framework and empirically confirmed a positive and significant relationship between effort expectancy and behavioural intention to adopt FinTech solutions. Based on the above arguments, the following hypothesis is proposed:

H2: Perceived ease of use has a significant and positive effect on behavioural intention to adopt FinTech services.

Perceived usefulness (PU) refers to the “extent to which an individual believes that using a particular technology will enhance their task performance “(Venkatesh, 2003). Performance expectancy, closely related to PU, is widely recognised as a dominant predictor of users' intention to adopt information technologies. In line with this, Chan et al. (2022) found that perceived usefulness significantly predicts users’ intentions to use FinTech services. Recent studies examining the factors influencing the intention to adopt financial technologies, such as mobile payment services and online banking, consistently highlight performance expectancy as having a positive and significant impact on customers' intention to use these services [15], [16], [17]. Based on the above arguments, the following hypothesis is proposed:

H3: Perceived usefulness has a significant and positive effect on behavioural intention to adopt FinTech services.

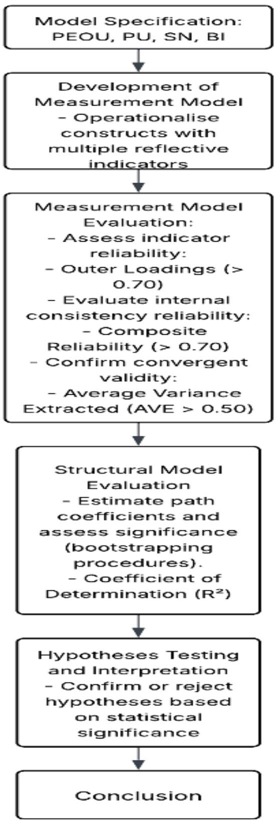

Model Specification: PEOU, PU, SN, BI
Development of Measurement Model - Operationalise constructs with multiple reflective indicators
Measurement Model Evaluation: - Assess indicator reliability: - Outer Loadings (> 0.70) - Evaluate internal consistency reliability: - Composite Reliability (> 0.70) - Confirm convergent validity: - Average Variance Extracted (AVE > 0.50)
Structural Model Evaluation - Estimate path coefficients and assess significance (bootstrapping procedures). - Coefficient of Determination (R²)
Hypotheses Testing and Interpretation - Confirm or reject hypotheses based on statistical significance
Conclusion

Fig. 1. Evaluation of the TAM Using PLS-SEM Approach

3 Methodology

A self-administered questionnaire was developed and distributed in Albania in 2025. The survey link was shared via Gmail and Facebook platforms to reach potential participants. The study sample consisted of 180 customers who had previously used Fintech services. All indicators were measured using a 5-point Likert scale, ranging from "strongly disagree" to "strongly agree." The conceptual model was subsequently evaluated using SmartPLS software, version 3.2.4.

PLS-SEM is recognised as a powerful tool for analysing complex, multi-layered relationships. It is particularly well-suited for behavioural research, especially when modelling intricate interactions among multivariate data. As noted by [18] SEM has been widely applied in the field of Information Systems to validate hypothesised relationships between constructs.

The estimation of SEM typically follows a two-step approach, consisting of the measurement model and the structural model. The first step involves the evaluation of the measurement model, which includes the assessment of construct reliability, indicator reliability, and convergent validity. Construct reliability was assessed using Composite Reliability (CR), with the accepted threshold being CR > 0.70, indicating sufficient internal consistency. Indicator reliability was evaluated through outer loadings, where values greater than 0.70 were considered acceptable. Convergent validity was measured using the Average Variance Extracted (AVE), with a recommended minimum value of 0.50, suggesting that the construct explains at least 50% of the variance in its indicators [19]. Once the construct reliability, indicator reliability, and convergent validity have been established, the analysis proceeds to the structural model. The structural model is evaluated by examining the coefficient of determination (R^2), path coefficients (β), and their corresponding t-values, typically calculated through a bootstrapping procedure to test the significance of the proposed hypotheses [20].

4 Results

The findings indicate that all constructs and indicators satisfy the requirements of the measurement model. Specifically, all indicators exhibit outer loadings greater than 0.7, and the average variance extracted (AVE) values exceed 0.5. As shown in Table 1, the composite reliability values are above the threshold of 0.70. Overall, the results confirm that all indicators are valid, the data demonstrate internal consistency, and convergent validity has been established.

Table 1. Results of the measurement model.

Constructs and indicators	Loadings	CR	AVE
Behavioural intention (BI)		0.946	0.827
BI1	0.881		
BI2	0.841		
BI3	0.825		
Subjective norm (SN)		0.923	0.736
SN1	0.854		
SN2	0.883		
SN3	0.855		
Perceived ease of use (PEOU)		0.895	0.826
PEOU1	0.941		

PEOU2	0.912		
PEOU3	0.922		
Perceived usefulness (PU)		0.896	0.741
PU1	0.921		
PU2	0.841		

The proposed framework, comprising the latent constructs of perceived ease of use (PEOU), perceived usefulness (PU), and subjective norm (SN), accounts for 68% (R^2) of the variance in behavioural intention to adopt FinTech services.

The results indicate a significant positive effect of perceived ease of use (b=0.147, $p < 0.05$), perceived usefulness (b=0.156, $p < 0.05$), and social influence (b=0.148, $p < 0.05$) on behavioural intention to use Fintech services. Therefore, the hypotheses H1, H2, and H3 are confirmed. Figure 2 presents the relationships of the structural model.

Table 2. Estimated path coefficients of the structural model.

Hypothesis	Path coefficients	P-Values
SN → BI	0.148	0.004
PU → BI	0.147	0.003
PEOU → BI	0.156	0.000

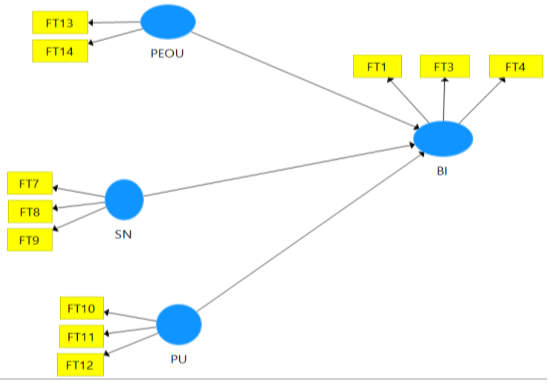


Fig. 2. Relationships of the structural model.

5 Conclusions

The empirical results demonstrate that perceived usefulness (PU), perceived ease of use (PEOU), and subjective norms significantly and positively influence behavioural intention to use FinTech services. Specifically, PU (b = 0.147, $p < 0.05$), PEOU (b =

0.156, p < 0.05), and subjective norms (b = 0.148, p < 0.05) emerged as significant predictors. These findings provide empirical support for hypotheses H1, H2, and H3, highlighting the importance of cognitive and social factors in the adoption process.

The implications of these results are twofold. First, they suggest that enhancing the perceived usefulness and ease of use of FinTech platforms can directly foster greater adoption among users. Therefore, practitioners should prioritise the development of intuitive, user-friendly systems that clearly communicate the benefits of their services. Second, the influence of subjective norms implies that social influence strategies, such as leveraging testimonials, peer recommendations, and social proof, could be critical in encouraging adoption. Overall, the study reinforces the relevance of the Technology Acceptance Model constructs in the FinTech context and offers actionable insights for both researchers and industry stakeholders seeking to facilitate the diffusion of FinTech innovations.

References

1. Irimia-Diéguez, A., Velicia-Martín, F., & Aguayo-Camacho, M. (2023). Predicting Fintech Innovation Adoption: the Mediator Role of Social Norms and Attitudes. Financial Innovation, 9(1). https://doi.org/10.1186/s40854-022-00434-6
2. Stewart, H., & Jürjens, J. (2018). Data security and consumer trust in FinTech innovation in Germany. Information and Computer Security, 26(1). https://doi.org/10.1108/ICS-06-2017-0039
3. Hu, Z., Ding, S., Li, S., Chen, L., & Yang, S. (2019). Adoption intention of fintech services for bank users: An empirical examination with an extended technology acceptance model. Symmetry, 11(3). https://doi.org/10.3390/sym11030340
4. Kou, G., Olgu Akdeniz, Ö., Dinçer, H., & Yüksel, S. (2021). Fintech investments in European banks: a hybrid IT2 fuzzy multidimensional decision-making approach. Financial Innovation, 7(1). https://doi.org/10.1186/s40854-021-00256-y
5. Berneis, M., Bartsch, D., & Winkler, H. (2021). Applications of Blockchain Technology in Logistics and Supply Chain Management—Insights from a Systematic Literature Review. In Logistics (Vol. 5, Issue 3). https://doi.org/10.3390/logistics5030043
6. Bajunaied, K., Hussin, N., & Kamarudin, S. (2023). Behavioral intention to adopt FinTech services: An extension of unified theory of acceptance and use of technology. Journal of Open Innovation: Technology, Market, and Complexity, 9(1). https://doi.org/10.1016/j.joitmc.2023.100010
7. Al Nawayseh, M. K. (2020). Fintech in COVID-19 and beyond: What factors are affecting customers' choice of fintech applications? Journal of Open Innovation: Technology, Market, and Complexity, 6(4). https://doi.org/10.3390/joitmc6040153
8. Cakerri, L., Petanaj, M., & Kosova, R. (2025). Factors influencing intention to use e-banking: An integrated model approach. International Journal Innovative Research and Scientific Studies, 8(1).
9. Xhafaj, E., Qendraj, D. H., Xhafaj, A., & Gjikaj, N. (2022a). A Hybrid Integration of PLS-SEM, AHP, and FAHP Methods to Evaluate the Factors That Influence the Use of an LMS.

International Journal of Decision Support System Technology, 14(1). https://doi.org/10.4018/IJDSST.286697

10. Xie, J., Ye, L., Huang, W., & Ye, M. (2021). Understanding fintech platform adoption: Impacts of perceived value and perceived risk. Journal of Theoretical and Applied Electronic Commerce Research, 16(5). https://doi.org/10.3390/jtaer16050106
11. Wirtz, B. W., Weyerer, J. C., & Geyer, C. (2019). Artificial Intelligence and the Public Sector—Applications and Challenges. International Journal of Public Administration, 42(7). https://doi.org/10.1080/01900692.2018.1498103
12. Chang, C. C. (2013). Library mobile applications in university libraries. Library Hi Tech, 31(3). https://doi.org/10.1108/LHT-03-2013-0024
13. Xhafaj, E., Qendraj, D., Kosova, R., Gjikaj, N., Mersinllari, O., & Alikaj, L. (2025). An Integrating Framework through the Extension of the UTAUT2 Model for Online Banking: A Context from a Two-Staged Approach with PLS-SEM and Fuzzy Z-AHP. Engineering Economics, Vol. (36) No. (2).
14. Aseng, A. C. (2020). Factors Influencing Generation Z Intention in Using FinTech Digital Payment Services. CogITo Smart Journal, 6(2). https://doi.org/10.31154/cogito.v6i2.260.155-166
15. Pham, A. H. T., Pham, D. X., Thalassinos, E. I., & Le, A. H. (2022). The Application of Sem–Neural Network Method to Determine the Factors Affecting the Intention to Use Online Banking Services in Vietnam. Sustainability (Switzerland), 14(10). https://doi.org/10.3390/su14106021
16. Xhafaj, E., Qendraj, D. H., & Salillari, D. (2024). A novel hybrid procedure of PLS-SEM, ANN and fuzzy TOPSIS for online banking. Journal of Intelligent and Fuzzy Systems, 46(2). https://doi.org/10.3233/JIFS-235388
17. Xhafaj, E., Qendraj, D. H., Xhafaj, A., & Gjikaj, N. (2022b). A Hybrid Integration of PLS-SEM, AHP, and FAHP Methods to Evaluate the Factors That Influence the Use of an LMS. International Journal of Decision Support System Technology, 14(1). https://doi.org/10.4018/IJDSST.286697
18. Xhafaj, E., Qendraj, D. H., Xhafaj, A., & Halidini, E. (2021). Analysis and evaluation of factors affecting the use of google classroom in Albania: A partial least squares structural equation modelling approach. Mathematics and Statistics, 9(2). https://doi.org/10.13189/ms.2021.090205
19. Hair, J. F., Risher, J. J., Sarstedt, M., & Ringle, C. M. (2019). When to use and how to report the results of PLS-SEM. In European Business Review (Vol. 31, Issue 1). https://doi.org/10.1108/EBR-11-2018-0203
20. Benitez, J., Henseler, J., Castillo, A., & Schuberth, F. (2020). How to perform and report an impactful analysis using partial least squares: Guidelines for confirmatory and explanatory IS research. Information and Management, 57(2). https://doi.org/10.1016/j.im.2019.05.003

20. Economic, Social and Cultural Factors Affecting IT Outsourcing in North Macedonia

Merita Bakiji[1] and Visar Shehu[2]

[1] South East European University, Tetovo, North Macedonia
[2] South East European University, Tetovo, North Macedonia
mb22289@seeu.edu.mk[1], v.shehu@seeu.edu.mk[2]
v.shehu@seeu.edu.mk

Abstract. The IT Outsourcing industry developed as a result of the need of companies to reduce costs and increase their clientele, to find future employees, to generate new ideas and to increase their technological capacity. This study examines the economic, social, cultural and other factors that influence the sustainability, stability and growth of the tech ecosystem in North Macedonia. This study relies on a methodology based on a systematic review of the literature as well as industry data or reports to understand some important elements such as market prospects and trends, favorable market conditions, workforce skills and influential policies for this segment. Overall, the findings suggest that among the key factors that make North Macedonia a leader in the IT Outsourcing sector are low labor costs, qualified talent and supportive regulations for businesses.
Another important element that represents a factor for companies entering outsourcing in North Macedonia is the government's support for development and innovation through various funds and co-financing.

Keywords: IT Outsourcing, Factors, Economic, Social, Cultural, North Macedonia

1 Introduction

In general, it is found that outsourcing emerged as a response to the increasing demands of the global market, which supports the principle of division of labor, a principle of particular importance in the economic system [1]. As cited in [2], IT Outsourcing has served as a convenient way or strategy by which modern organizations have managed to minimize their operational costs and utilize capabilities from outside the company. Similarly, Cai et.al (2020), argue that Outsourcing is a new way of doing business that essentially involves the involvement of third parties from outside in order to reduce costs and to devote them to core business activities and to generate new innovative ideas, as cited in [1].

Recently, there has been a ‘war’ between cost-cutting and innovation-oriented contracting mindsets for both customers and providers [3]. Therefore, in recent years, research has provided arguments on the performance of the organization that in order to be sustainable, the organization must move away from the focus on cost reduction and go towards innovation, where customers and suppliers collaborate to create innovation in products, services or to open new markets. Thus, by distancing itself from cost minimization and focusing on innovation, the relationship between customer and supplier will be a "win-win", where both parties will enjoy high performance [2]. As a result of changing customer behavior, research on Outsourcing is constantly increasing, and even in the curricula of administration faculties, it has been included as a university subject. It is considered a field of knowledge that serves the organization as a strategy through which both the company and third-party institutions (providers) are empowered [4].

So, Outsourcing, on the other hand, is considered the main driver of economic development in the era of capitalism [5].Regarding IT Outsourcing in particular, as cited in [6], also, the use of this sector is done for reasons of cost reduction, focusing on core business activities and increasing work efficiency. Recently, IT Outsourcing has become the subject of research due to the large number of issues surrounding it. From numerous cross-sectional studies, many empirical areas have been identified for which future research is also recommended. According to [5], IT Outsourcing, through information systems, enables the connection of three elements: people, processes and technology, therefore, when we say that the information system is successful, we also refer to the success of outsourcing. This is also argued by the fact that many researchers, through the use of IT outsourcing activities, have managed to evaluate the success of the information system itself. From the research conducted by [5], it is proven that three important factors such as: configuration quality, system quality and service quality, influence three other factors such as: communication quality, user satisfaction and product/service usage. Therefore, these mediating factors provide both individual and organizational benefits resulting in the overall success of IT Outsourcing.

2 Purposes and objectives of the research

The purpose of this research is to assess the sustainability of the IT sector in North Macedonia, including several factors such as economic, social and cultural that have a direct impact on the development of this sector.

The main objectives set in this study are:

To assess the economic factors that influence the outsourcing of IT services in terms of labor costs, business and investment regulations, government support, etc.

To examine cultural factors by considering the suitability in international work environments and cooperation with global clients.

To assess social factors by including the education and training of the workforce in terms of IT, including formal and non-formal education.

To propose strategic recommendations for policymakers and investors for future perspectives.

3 Methodology of the research

The methodology in this research is based on an in-depth review of the existing literature in order to understand the main concepts and identify approaches related to IT Outsourcing in the global context and especially in the context of North Macedonia. Also, with the help of industry reports, through secondary data related to the factors affecting IT Outsourcing in North Macedonia, the fulfillment of the objectives and purpose of this study has been achieved. This study specifically identifies economic, geographical, social and cultural factors to present data and facts from relevant institutions. Thus, by examining factors such as: cost effectiveness, location and culture, education of professionals, fluency in English and ease of doing business, this study provides a comprehensive summary of the opportunities that should be considered by investors and policymakers for further development of this industry segment.

4 Factors determining the reasons for ITO in North Macedonia

Despite the fact that North Macedonia is a small country with a relatively small population (2 million), it is already a country that produces talented and motivated professionals, especially in terms of IT skills. It benefits from its geographical position due to its location in the heart of the Balkan Peninsula, benefiting from the intersection of two major European transport corridors that connect Central Europe with the Aegean and the Black Sea. Also, the possession of two airports, Skopje and Ohrid, makes Macedonia or Skopje a 2-hour drive from Thessaloniki, which is the main port center for the Balkans and Southeastern Europe and connects Europe with Asia and China [7].In a report by the author [8], several reasons are explained why IT Outsourcing is flourishing in North Macedonia. She emphasizes that unlike other European countries, Skopje is considered a leading destination for this field. According to her, North Macedonia is ready to offer benefits as a result of rapid engineering advances that redefine the success of outsourcing. In addition, the considerable number of professionals in the field of IT that come out of 9 universities that develop over 1000 software developers every year, makes North Macedonia a more attractive place to carry out ITO. In North Macedonia, non-formal education institutions are also very functional, which offer practical skills necessary to start a career in IT. Among the centers that offer this type of education are known Semos, Brainster, IWEC, and many others.

Referring to the report given by [9], the reason why North Macedonia has gained the trust of companies to enter the IT Outsourcing industry is argued by emphasizing that North Macedonia offers the opportunity for a well-educated workforce where English is very widespread and widely understood by young people because it is taught in primary school. This skill is one of the most sought-after criteria by companies looking for talent for ITO services. In addition to English, there is the ability for workers to master French and German. This is widely supported by the education system. Regarding [10], in North Macedonia, universities enter into collaboration agreements with international IT companies so that their curricula are up-to-date with industry standards and market demands. The talent produced by these Universities, in addition to coding skills, is also skilled in software development, cybersecurity, data analysis, and AI tools.

So, according to [11], the Balkan countries have experienced a continuous growth in terms of the workforce in the field of IT. As a result of this growth, it is expected that in the future, IT Outsourcing will further develop in these countries, penetrating the international market. According to the author in question, this development has come from changes in the advancement of education which produce a skilled and talented workforce.

In the report given by [12], a wide list of reasons why outsourcing to North Macedonia should be done is given. This list includes many factors, from geographical, economic, political and social. According to this report, the argument is supported that in North Macedonia there is a very favorable business environment in which we have a low-cost labor force, but a qualified and innovative workforce with prospects that aim to increase salaries for software developers. This market is considered the fastest growing market in the Adriatic Sea region. On the other hand, it is reported that major international hardware and software suppliers, such as Microsoft, HP and IBM, offer local sales and support services for the North Macedonian market. According to the author [13], who also confirms the fact that North Macedonia, as part of the Western Balkans region, is not considered a simple outsourcing country as countries such as Asia and the Pacific region are known, but it is something more than that. This argument is based primarily on the level of education and other characteristics related to education. Although prices in Asia and the Pacific are very attractive, again the ratio between costs and the value of services makes North Macedonia a leader in the selection of outsourcing companies. On the other hand, the geographical position/proximity, cultural similarities and the close time zone with developed countries make North Macedonia an attraction for ITO development. It can be freely compared to South America. Similar to this, based on the report of [9], a list of the main factors for contracting in North Macedonia has been made, which are as follows:

Fluency in English - in North Macedonia, English is widely spoken and understood and this skill is one of the almost indispensable needs and criteria that many companies look for in foreign partners.

Educated and qualified talent - in addition to formal education, also due to the advancement of informal education, North Macedonia produces young talent especially in the field of IT, who are well trained both in theoretical and practical aspects.

Location and culture - North Macedonia has an excellent location in Europe and the Balkans and this reason facilitates the approach of operations due to the ease of reaching the country. Also, the similarity and influence on the culture of neighbors can be considered a factor for companies when outsourcing.

Cost effectiveness - it is known that the main goal of outsourcing is to reduce costs, therefore North Macedonia is known for not very high labor costs compared to other European countries.

Another factor listed by [14], is the ease of doing business because in North Macedonia there is a friendly spirit for doing business by possessing processes and policies that support and encourage innovation. Also, it is reported by "Doing Business 2022" that North Macedonia ranks 17th globally for the ease of starting a business. So, through simple processes, government support and proper IT infrastructure, the IT Outsourcing industry in the country has grown significantly in recent years.

Some of the additional factors it lists by [12], are as follows:

- Macedonia is a candidate country for EU and NATO.
- Political, monetary and financial stability.
- Ranked 22nd out of 183 economies for ease of doing business by the World Bank's Doing Business Report 2012.
- The legal system, including intellectual property rights protection and laws, is in line with EU legislation.
- Corporate and personal income taxes are flat at 10%.
- VAT is set at 18%, with a reduced rate of 5% for specific items.

As a result of the process of North Macedonia's accession to the European Union, the country's laws tend to adapt to the requirements and laws of other European Union countries and have cultural similarities with other European countries, and this has attracted investors as a result of a calm work environment and understanding of client expectations [14].

Regarding the regulatory environment, each country, including North Macedonia, has its own regulatory environment that companies must comply with when outsourcing. All companies must operate according to local laws and regulations, which include, among others, labor laws and taxes, which often differ from the laws of their own country [15].

According to EMAPTA report [16], although North Macedonia is not yet a member of the European Union, it strictly follows data security and privacy laws. The country's legal o, (GDPR) are in line with the aim of ensuring a high level of protection of personal data. The regulation covers all types of businesses operating in North Macedonia, and these also apply to the outsourcing sector.

So, outsourcing in North Macedonia has several benefits related to access to educated and talented software developers, lower labor costs compared to other countries, and legal regulations that protect data security and are in line with European Union standards such as GDPR. By outsourcing IT services to North Macedonia, companies are able to reorganize their operations and focus on core business activities, while leveraging the expertise of Macedonian professionals for their IT needs [8]. Furthermore, this is also supported by this report of which states that North Macedonia has signed the international intellectual property agreement whereby businesses are protected in terms of information security [14].

Unwavering government support and private investment have helped the rapid expansion of the tech ecosystem in North Macedonia. Various tax incentives and numerous grants for tech startups and foreign investors have favored the IT sector. These conditions have led to the establishment of various technology and innovation centers in the country. In North Macedonia, the tech community regularly hosts tech conferences, workshops, and hackathons to stimulate innovation in the IT field and keep local talent up to date with global trends. The dynamic ecosystem allows foreign companies outsourcing to North Macedonia, among other things, to benefit from technological developments. [10].

Government support for the IT sector has been increasing for years. Thus, FITD (Fund for Innovation and Technological Development) has co-financed many projects from different sectors and industries. The following figure 1 presents the industries financed by FITD [17] expressed in million euros.

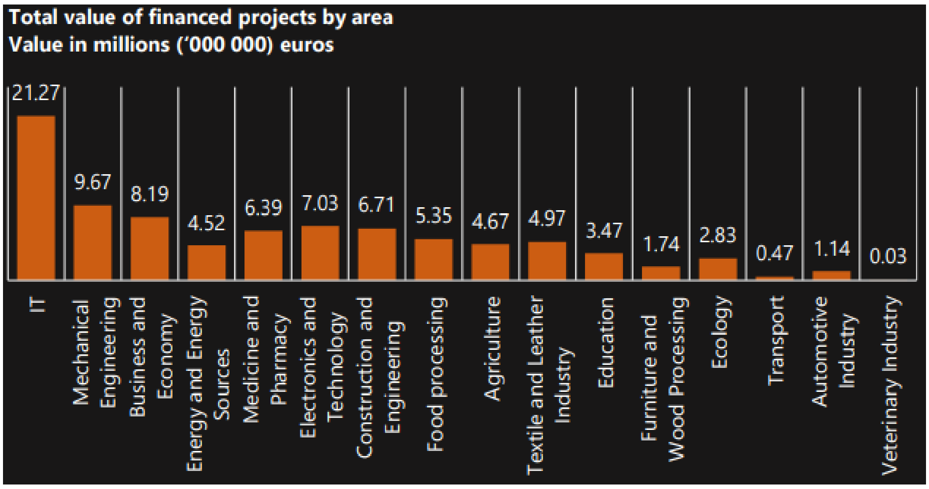
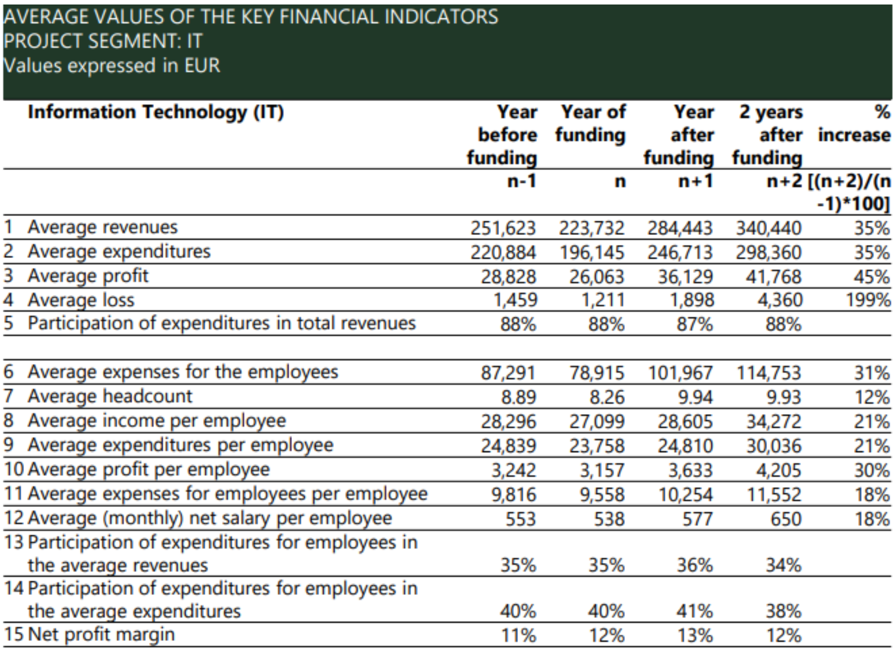


Fig.1. The industries financed by FITD [17] expressed in million euros.

Source: fitr.mk

As can be seen from this diagram, the sector most financed by FITD is the IT sector or industry with a value of 21.27 million euros. Next, the most financed sectors are companies in the Mechanical Engineering sector with 9.67 million euros, the Business and Economics sector with a value of 8.19 million euros, then the Electronics and Technology sector with 7.03 million euros, Construction and Engineering companies were supported with 6.71 million euros, and the Medicine and Pharmacy sector was supported with 6.39 million euros.

The figure 2 below statistically describes the Financial Parameters of Co-financed Companies per Year and the Project Segment - INFORMATION TECHNOLOGY.



AVERAGE VALUES OF THE KEY FINANCIAL INDICATORS
PROJECT SEGMENT: IT
Values expressed in EUR

	Information Technology (IT)	Year before funding	Year of funding	Year after funding	2 years after funding	% increase
		n-1	n	n+1	n+2	[(n+2)/(n-1)*100]
1	Average revenues	251,623	223,732	284,443	340,440	35%
2	Average expenditures	220,884	196,145	246,713	298,360	35%
3	Average profit	28,828	26,063	36,129	41,768	45%
4	Average loss	1,459	1,211	1,898	4,360	199%
5	Participation of expenditures in total revenues	88%	88%	87%	88%	
6	Average expenses for the employees	87,291	78,915	101,967	114,753	31%
7	Average headcount	8.89	8.26	9.94	9.93	12%
8	Average income per employee	28,296	27,099	28,605	34,272	21%
9	Average expenditures per employee	24,839	23,758	24,810	30,036	21%
10	Average profit per employee	3,242	3,157	3,633	4,205	30%
11	Average expenses for employees per employee	9,816	9,558	10,254	11,552	18%
12	Average (monthly) net salary per employee	553	538	577	650	18%
13	Participation of expenditures for employees in the average revenues	35%	35%	36%	34%	
14	Participation of expenditures for employees in the average expenditures	40%	40%	41%	38%	
15	Net profit margin	11%	12%	13%	12%	

Fig. 2. Financial Parameters of Co-financed Companies per Year and the Project Segment.

Source: fitr.mk

As can be seen from the figure, the percentage of profit has increased from the first year of implementation of this program to the fourth year, where profit has increased by 35%. The average increase also has an increase of 35%, while in contrast, the average expenses have a significant increase of 45%. From 45% we also have the increase in the profit level, but on the other hand we have a very high increase in the loss level by 19%, although compared to the profit, the loss level is significantly lower, and consequently the entire segment has a positive financial result.

Average employee costs have increased by 31% and the average number of employees has also increased by 12%. This shows that employee salaries are increasing at a high rate and it can be seen that the average net salary has increased by 18%.

The participation of employee costs in expenses is continuous and stable between 38% - 41%, which highlights the fact that employees significantly affect the entire expense structure and between 34%-36% of income. The net profit margin is between 11% - 13% and is stable, in addition to the absolute increase in profit by 45%.

References

1. A. S. R. Y. Flora Kulembayeva, "Economic Efficiency of Outsourcing Business Models: A Comparative Assessment," *Global Journal of Flexible Systems Management,* vol. 23, no. 1, pp. 75-88, 2022.
2. Q. ". B. O. A. Ezekiel Leo, "Outsourcing for Sustainable Performance: Insights from Two Studies on Achieving Innovation through Information Technology and Business Process Outsourcing," *Sustainability,* 2022.
3. A. A. J. K. Marfri-Jay Gambal, "Strategic Innovation Through Outsourcing in the ITO and BPO context – A Theoretical Review," *Journal of Strategic Information Systems,* 2022.
4. L. F. M. H. E. P. Q. E. S. L. B. I. V. H. M. O. Morelia Magaly Barra Solano, "OUTSOURCING AS A PRODUCTION MODEL AND CUSTOMER LOYALTY OF A FINANCIAL COMPANY," *Visión de Futuro,* vol. 27, no. 2, pp. 153-169, 2023.
5. S. Paudel, "Validation of Information Technology Outsourcing Success Model using Structural Equation Modelling," *Asia Pacific Journal of Information Systems,* vol. 33, no. 1, pp. 206-226, 2023.
6. J. Sloniec, "Risk Factors Determining the Benefits of It Outsourcing - a Structural Model," *Preprints,* 2023.
7. S. Xhemo, "WHY CHOOSE NORTH MACEDONIA FOR IT OUTSOURCING IN 2021?," ITWorks, 2020.
8. R. Pavlovska, "Why IT Outsourcing in North Macedonia Thrives," IWConnect, 2024.
9. D. Gallimore, "Top 20 BPO companies in North Macedonia," Outsourceaccelerator, 2024.
10. Technoperia, "The Benefits of Outsourcing Tech Projects to North Macedonia," Technoperia, Skopje, 2024.
11. T. Ameti, "YOUTH AND RISE OF OUTSOURCING OPPORTUNITIES IN BALKAN COUNTRIES - CASE OF BULGARIA, ROMANIA AND NORTH MACEDONIA," *Trends in Economics, Finance and Management Journal,* vol. 1, no. 1, pp. 74-90, 2019.
12. Axeltra, "Why outsource to Macedonia? Top reasons for nearshore outsourcing to Macedonia," Axeltra.
13. B. Manev, "ICT segment condition in North Macedonia and Western Balkan Region," International Growth Agency, 2023.
14. E. I. Solutions, "Why North Macedonia is a Top Destination for IT Outsourcing?," Eagle IT Solutions, Struga, 2023.
15. A. B. D. S. B. Ceneta Telak, "Process of Digitalization as Outsourcing: Challenge for the International Companies in the Republic of North Macedonia," in 8th FEB International Scientific Conference: CHALLENGES IN THE TURBULENTECONOMIC ENVIRONMENT AND ORGANIZATIONS' SUSTAINABLE DEVELOPMENT, 2024.
16. EMAPTA, "Outsourcing to Macedonia FAQs," EMAPTA, 2024.
17. FITD, "PERFORMANCE ANALYSIS OF COMPANIES CO-FUNDED BY FITD," Fund for Innovation and Technological Development, Skopje, 2021

Section 3: Cloud Computing and Distributed Systems

21. Cloud-Based AI Solutions for Personalized Learning: An Exploration of Algorithms, Scalability, and Challenges in Implementation

Elida Hadro[1], Jaumin Ajdari[2] , Merve Paçarizi Kabashi[3] and Nida Santuri Fishekqiu[4]

[1,2,3,4]South East European University, Tetovo, North Macedonia
[1]eh33256@seeu.edu.mk
[2]j.ajdari@seeu.edu.mk
[3]mp33255@seeu.edu.mk
[4]ns33254@seeu.edu.mk

Abstract. In the rapidly evolving landscape of education, personalized learning has emerged as a key approach of modern pedagogy, leveraging adaptive instructional strategies to meet the individual needs of learners. The integration of cloud computing and artificial intelligence (AI) has improved personalized learning by enabling scalable, data-driven, and real-time adaptive learning experiences. AI-powered cloud solutions enable more effective and targeted learning by analyzing student interactions, dynamically tailoring content, and predictive learning results. Nevertheless, while prior studies have investigated AI in education, there is a paucity of study about its large-scale implementation inside cloud-based infrastructure and its direct impact on student performance. This research explores the essential AI algorithms driving personalized learning, the scalability of cloud-based AI systems, and challenges including data privacy, computational costs, and accessibility issues. The study offers a conceptual analysis of the potential future impact, limitations, and current advancements of AI-powered cloud solutions in education. The PACE Model – Personalization, Architecture, Challenges, and Evaluation – is introduced as a guiding framework. The findings enhance understanding of the intersection between AI, cloud computing, and personalized learning, highlighting critical points for educators, policymakers, and researchers. The purpose of this study is to synthesize existing knowledge in order to help future developments in AI-driven learning systems and their use across diverse educational contexts. Significant challenges remain to be solved in the areas of ethics, data protection and privacy, and equal access.

Keywords: Cloud Computing, Artificial Intelligence, Personalized Learning, Adaptive Learning Systems, Educational Technology.

1 Introduction

Technology has permeated every aspect of our life in the digital age, changed whole sectors, and reinterpreted our interactions with the environment and each other. Education is one area that has experienced tremendous transformation as a result of technology developments [1]. Human development depends on education since it helps to acquire the knowledge, abilities, and moral principles required for success in life. It helps people to make informed decisions, solve problems, and actively participate in their communities [2], [3]. Conventional educational systems usually follow a uniform approach whereby every student receives same instruction at an equivalent rate. This approach neglects to consider the several learning styles, capacities, and interests of students [4].

Information technologies, especially artificial intelligence (AI), are transforming the educational landscape. AI algorithms and instructional robots have become integral components of learning management and training systems, facilitating a diverse range of teaching and learning activities [5]. AI delivers personalized and adaptable experiences that address individual learner needs based on their capabilities and preferences, hence improving learning efficacy and productivity [6]. AI algorithms are essential for facilitating personalization in educational platforms. Artificial intelligence encompasses a range of sophisticated technologies, including machine learning (ML), natural language processing (NLP), and data analytics. The incorporation of these technology components facilitates the realization of individualized learning experiences customized to the distinct needs and preferences of individual learners [7]. Moreover, AI facilitates the provision of immediate feedback, providing learners with timely and precise guidance while simultaneously improving and streamlining administrative processes to ensure smooth and efficient operations [8].

As educational institutions globally strive to offer personalized, accessible, and affordable learning experiences, cloud-based technologies have surfaced as significant facilitators of these goals [9]. The application of cloud computing in education spans a broad spectrum, encompassing adaptive learning platforms, virtual classrooms, and sophisticated analytics systems that offer real-time insights into student performance and engagement [10]. Cloud computing, as an emerging intelligent technology, can efficiently process large-scale data within seconds, significantly enhancing computational speed and efficiency [11]. In this context, cloud computing and AI stand out as two of the most promising technologies. The cloud provides the ability to scale and access resources seamlessly, removing limitations imposed by physical locations and geography. Simultaneously, AI offers the potential for personalization and efficiency, tailoring content to meet the unique needs of each student and providing educators with important insights [1], [12]. One of the most impactful applications of cloud computing in this sector is the scalable infrastructure it provides [13]. This enables educational institutions to adaptively distribute computational resources according to fluctuating instructional needs, thus improving both cost-effectiveness and the operational efficiency of cloud-based educational applications [14].

While cloud-based AI solutions in education hold significant promise, their implementation faces numerous obstacles. Important issues encompass data privacy and security [15], the interoperability of systems, algorithmic bias present in AI models [16], and technical constraints like infrastructure deficiencies in low-resource regions [17]. Addressing these issues requires not only technical solutions but also ethical guidelines and policy frameworks.

In response to the need for an integrated approach that combines personalization techniques, scalable cloud infrastructures, and the resolution of implementation challenges this paper provides a conceptual analysis and introduces the PACE framework – an integrated conceptual model that focuses on four interrelated layers: Personalization, Architecture, Challenges, and Evaluation. The PACE model synthesizes insights from current literature into a coherent structure that might direct researchers, developers, and educational institutions in their attempts to establish more effective and sustainable AI-driven learning environments. Though the framework has not yet been empirically applied, it provides a theoretical basis for next studies and pragmatic uses in cloud-based AI solutions for personalized learning.

2 Literature Review

The origins of personalized education can be found in ancient teaching methods, where the learning experience was inherently customized to suit the individual learner. Recent studies indicate that the revival of personalized teaching reflects a return to its foundational principles, now enhanced by cutting-edge technologies [18], [19]. The evolution of AI in education originates from Skinner's research on adaptive learning machines in the 1950s, emphasizing the adaptation of instruction to the learner's attributes. The emergence of Intelligent Tutoring Systems and Adaptive Hypermedia Systems in the 1990s significantly accelerated this progression [20]. The advancement of AI, especially deep neural networks from the 1990s to the 2020s, has significantly improved the capacity to model learner behavior and provide real-time adaptive feedback [21].

Adaptive learning systems currently employ algorithms to replicate human cognitive behavior and customize content accordingly [20], [22]. The recent developments have enabled adaptive learning systems to refine content delivery, pinpoint learning gaps, and modify instructional strategies in real-time through data analytics, providing more detailed and personalized learning experiences [20].

The integration of AI and cloud computing has significantly improved the scalability and efficiency of these systems. Cloud platforms facilitate demanding computational tasks and provide real-time updates, all while supporting a significant number of users simultaneously. The teacher-student dynamic has been redefined, with teachers empowered to serve as facilitators while AI systems manage personalized instruction [23]. The growth of AI in education is anticipated to lead to a rapid expansion of the AIED (Artificial Intelligence in Education) market, indicating a significant uptake by learners and educators alike [5], [24].

Numerous frameworks have been developed to combine AI and cloud computing for personalized educational experiences. The frameworks differ in their methodologies, yet they predominantly emphasize the application of AI to provide real-time personalized learning recommendations and feedback. For example, the LearnMate framework utilizes large language models (LLMs) to create customized learning plans and deliver explanations suited to the specific needs of each learner [25]. PlanGlow highlights the importance of user control and transparency in the generation of learning paths, allowing students to remain informed about their academic progress and to adapt their study plans as needed [26]. The Artificial Intelligence-Enabled Intelligent Assistant (AIIA) framework combines NLP with cognitive modeling to alleviate cognitive load and offer immediate learning assistance [27]. In the meantime, the framework developed by Murtaza et al. [28] introduces a modular system that includes data, adaptive learning, and recommender modules, facilitating the scalable integration of AI-driven personalization and learner feedback. Zhang [19] presents an innovative teaching platform enhanced by AI, employing ML techniques such as Pearson correlation and regression to assess student behavior and provide timely adaptive interventions. The system is constructed on a cloud-based architecture, guaranteeing scalability and low latency, which facilitates personalized learning on a large scale. In a similar vein, Ghallabi et al. [29] put forward a cloud-based learner modeling system that incorporates personalization elements like language preference and prior knowledge. Employing Support Vector Machines (SVM), it examines learning behaviors to provide customized content effectively. The model exhibited enhanced scalability, cost-effectiveness, and performance when juxtaposed with conventional systems [29].

Practical applications of AI and cloud-based systems underscore their capabilities. The Croatian e-Schools Programme has effectively incorporated AI tools, including virtual assistants and recommendation systems, into the educational framework. The tools offer tailored study plans and immediate feedback, greatly enhancing learning results and supporting educators [30]. In the same way, Yang [31] created a platform that integrates digital empowerment technologies with AI to provide tailored learning recommendations, thereby facilitating students' independent learning. The capabilities of AI in education are exemplified by models that forecast personal learning styles through cognitive and emotional indicators. The study conducted by Lokare and Jadhav [32] utilized electroencephalogram (EEG) data to forecast learning preferences, thereby aiding in the customization of learning experiences to accommodate the diverse cognitive loads and engagement levels of students.

Despite the advancements, there is still a gap in frameworks that integrate cloud infrastructure, AI algorithms, and practical limitations like scalability, privacy, and equality. The majority of current models either ignore ethical issues or concentrate only on particular technological elements. In order to integrate these elements into a thorough model that tackles the ethical and technological aspects of AI-powered customized learning in cloud settings, this study suggests the PACE framework (Personalization, Architecture, Challenges, and Evaluation).

3 Research Methodology

This paper develops the PACE framework using a conceptual analysis approach. Particularly when empirical data collecting is not feasible, conceptual analysis is appropriate for research aiming at organizing and structuring current knowledge into a cohesive theoretical model. In order to build an integrated framework addressing important features crucial for future research and application, this paper systematically reviews and synthesizes existing literature on cloud-based AI systems, personalized learning technologies, and implementation challenges.

The decision to apply a conceptual analysis approach results from the complex and changing landscape of including artificial intelligence and cloud technologies inside the educational sector. Establishing a theoretical basis by extensive literature synthesis is a major starting point for future empirical validation given the present constraints of empirical studies in this interdisciplinary domain.

The methodology consisted of an organized review of journal articles, conference proceedings, systematic reviews, technical reports, and case studies. Sources were chosen for their relevance to cloud computing, the role of AI in education, personalized learning, and the challenges associated with system implementation. Essential concepts and frameworks were identified, analyzed, and synthesized to develop the PACE framework. Careful consideration was devoted to pinpointing gaps, inconsistencies, and emerging trends to guarantee that the proposed model remains both forward-thinking and practically applicable.

4 Conceptual Analysis: Key Dimensions of Cloud-Based AI for Personalized Learning

4.1 Defining Personalized Learning in the Educational Context

The learning environment is being completely redesigned with AI technologies to ensure that learners acquire the necessary skills for future employment and to enhance educational outcomes [33]. Personalization means customizing learning trajectories, support systems, and instructional materials based on the unique needs, preferences, and attributes of individual learners [28]. AI-based systems evaluate students' skills and provide relevant materials dynamically, therefore guaranteeing more customized and effective learning environments [34]. Four main elements underlie cloud-based AI personalization and help to enable the dynamic adaptation of the learning process.

Learner Profiling. Creating accurate learner profiles is a key part of personalization. These profiles bring together information from multiple sources, such as interactions with the student, assessments, and demographic data. These profiles help to figure out what the learner can do, what they like, and what their goals are [33]. Learner profiling gathers cognitive, affective, behavioral, and environmental traits to customize education at an individual level [35]. Recent techniques move beyond conventional static methods such self-report surveys by using AI to continuously update learner profiles in real time [20]. Using academic and behavioral data, predictive analytics tools help to early identify learning trends and potential threats, so supporting proactive and scalable personalizing [36]. Support vector machines, decision trees, neural networks, and naive Bayes classifiers among other ML models are extensively used to project learner outcomes and guide tailored interventions [37], [5].

Adaptive Content Delivery. Adaptive content delivery systems, powered by cloud infrastructure's computational scalability, dynamically modify course materials based on student data. These systems use advanced AI algorithms to dynamically modify the difficulty, speed, and format of instruction [38]. Methods including NLP, reinforcement learning, and collaborative filtering guarantee that students get materials fit for their present knowledge level and preferences [39], [40]. Overcoming the restrictions of conventional on-site education systems, cloud computing helps to enable the simultaneous personalizing of learning experiences at scale [9].

Feedback Mechanisms. Real-time feedback is essential for fostering a personalized learning environment. Cloud-based AI systems can provide immediate and actionable feedback that guides learners toward improvement. They offer instantaneous, customized advice including cognitive as well as emotional aspects of learning [4]. Affective computing and sentiment analysis use emotional cues to give responsive and helpful feedback. Reinforcement learning, on the other hand, lets AI bots change their teaching methods based on how students interact with them [6]. Cloud architectures enable the swift implementation of feedback systems across extensive learner populations. Meanwhile, Bayesian knowledge tracing models support the monitoring of learners’ knowledge development over time, ensuring that feedback remains aligned with each individual's progress [3], [41].

AI Algorithms for Personalized Learning. A range of AI algorithms supports the personalizing features of cloud-based learning systems, each of which serves different purposes in adjusting to student needs [42]. ML algorithms are fundamental to the creation of adaptive learning paths, which dynamically adjust the learning experience based on a student's performance and progress. These algorithms analyze student interactions with learning materials, including quiz scores, completion times, and engagement levels, to build a model of each learner's knowledge and skills [32]. Various machine learning techniques are employed in adaptive learning environments. Supervised learning is used to predict student performance. Unsupervised learning helps in grouping students with similar learning patterns. Reinforcement learning is applied to optimize learning pathways based on student interactions [6]. Using historical data, supervised learning models include deep neural networks, random forests, and decision trees predict learner performance [3], [36]. Unsupervised learning techniques, such as k-means and hierarchical clustering, reveal hidden learner groupings to guide focused personalization tactics [43]. Techniques in reinforcement learning, including Deep Q-learning and multi-armed bandit models, enhance content sequencing and adaptation via iterative learning processes [44]. Models in NLP enable the creation of personalized content, streamline evaluation processes, and improve interactive tutoring functionalities [6]. NLP techniques enable the analysis of educational content, allowing systems to understand the meaning of text, extract key concepts, and categorize learning materials. This capability is essential for tailoring content to a student's current knowledge level and learning style [40], [45].

Collectively, these AI technologies create a smart, flexible, and scalable personalization framework that can address the varied and changing requirements of contemporary learners. The selection of an algorithm is contingent upon the objectives of personalization and the accessibility of data.

4.2 Architectural Considerations

The effectiveness, scalability, and security of AI-driven personalizing in education depend on a well-designed cloud infrastructure [1]. Different service models presented by cloud computing have different levels of management responsibility [46], [47]. The particular requirements and technical know-how of the educational institution determine which model is best.

- Infrastructure as a Service (IaaS) provides basic computing resources including servers and storage, so giving maximum control and flexibility for the deployment of specialized AI hardware [48].
- Platform as a Service (PaaS) removes the complexity of infrastructure by providing a platform for creating, executing, and maintaining applications. Specialized tools for creating and implementing AI models are offered by AI PaaS [48].

- Software as a Service (SaaS) offers pre-configured, cloud-based apps that give users instant access to AI-powered teaching resources with minimal setup [48].

Cloud computing provides a secure and accessible environment for educational data by offering a variety of storage solutions that can accommodate the volume of data. For AI algorithms to efficiently retrieve and analyze data, cloud data management and organizing strategies are essential [13].

One major benefit of cloud environments is their significant computing capability, which is necessary to meet the processing needs of AI algorithms used for personalization. High-performance computing resources, like as GPUs, are made available through cloud platforms, greatly speeding up the development and implementation of intricate AI models [21].

Scalability and Adaptability. For personalized learning, cloud-based AI systems must be scalable—that is, able of a system to manage an increase in demand or workload. This holds particular significance in educational settings where there can be heightened demand, necessitating a system capable of rapid adaptation while maintaining service quality [1]. The architectural design of the personalized learning platform is crucial for its scalability. Effective cloud data management solutions, encompassing improved data storage and retrieval systems, are crucial for managing extensive datasets without performance decline [49]. The selection and implementation of AI algorithms influence scalability as well. Certain algorithms may exhibit greater computational intensity than others, necessitating increased processing power as the user count and data points escalate [49]. Therefore, selecting efficient and scalable AI models is crucial for maintaining performance as the system grows. Cloud computing is a solid foundation for scalability. However, careful architectural design, useful AI algorithms, and smart use of AI for resource management are necessary to make sure that cloud-based AI solutions for personalized learning can grow to meet the needs of a wide range of students and students who are growing [48], [49].

Workload management is made more difficult by the wide variety of AI frameworks and technologies available. TensorFlow, PyTorch, and MXNet are just a few of the frameworks that may be used to construct different AI jobs; each has its own set of resource needs and optimization strategies. In order to support a broad range of AI frameworks and optimize resource allocation for each unique scenario, cloud infrastructures must be flexible enough to accommodate this diversity. To adapt to the quickly shifting landscape of AI development, compatibility and smooth interaction with developing AI tools are crucial [50].

4.3 Challenges in Implementation

There are a number of important challenges that need to be properly considered when implementing cloud-based AI systems for personalized learning in the educational environments.

Data Privacy and Security. The importance of privacy and data security cannot be overstated, given the extensive and sensitive nature of the data collected from learners, which encompasses personal information, learning progress, and engagement metrics. Concerns over data collecting transparency, storage security, and informed permission from learners have been raised by AI gathering significant volumes of data to customize learning experiences [51]. Privacy and data governance structures will have to evolve together to guarantee that data collecting, permission, and security policies stay strong to fit the complexity of these systems. The data includes a wide range of information, from personally identifiable characteristics like names, email addresses, and contact information to academic achievement measures, learning patterns, behavioral data, and sensitive biometric data [8]. Given the sensitivity of this information, privacy issues are crucial. Institutions must manage such data in a safe and responsible manner, following existing data protection rules and industry best practices [8]. Zhang [19] stated that colleges should prioritize the establishment of strong security mechanisms to protect student data. This requires implementing encryption techniques, establishing secure authentication procedures, conducting frequent vulnerability assessments, and monitoring alert systems for possible threats [8].

Institutions must ensure compliance with pertinent data protection regulations and privacy laws. For instance, in the European Union, the General Data Protection Regulation (GDPR) imposes stringent requirements for processing personal data [52]. Similarly, jurisdictions like California have introduced privacy regulations such as the California Consumer Privacy Act (CCPA) [53]. Compliance means fully knowing the legal duties related to data privacy, putting in place the right safeguards, doing data protection effect studies, and giving students the right information about their rights and how their data is used [8].

Addressing privacy problems in the context of AI-driven cloud settings allows institutions to create an environment that protects learners' privacy while using AI's revolutionary potential.

Ethical Considerations. Recent advances in AI ethics highlight the importance of transparent and explainable models, particularly in education, where trust and understanding are crucial. Explainable AI frameworks enable students and educators to understand the underlying logic of AI-generated decisions, boosting confidence and responsibility in adaptive learning systems [54].

To support the ethical management of user data, educational institutions can adopt the Fairness, Accountability, and Transparency (FAT) framework—an increasingly influential model in AI ethics. By integrating FAT principles, institutions can proactively safeguard learner privacy while cultivating confidence in AI-driven systems. Fairness addresses the possibility of algorithmic bias by underscoring the necessity for AI models to be trained on varied datasets that encompass a wide array of learner experiences [8]. Accountability emphasizes the importance of regular audits and continuous oversight of AI systems to maintain ethical integrity and responsible data management. Effective accountability mechanisms include strong data governance structures that support ongoing performance evaluations and algorithmic adjustments [55]. Transparency is critical for retaining user trust, particularly in AI-powered learning settings. By explicitly disclosing data procedures, such as how personal information is collected, processed, and utilized, learners gain greater control over their data [8].

Integration and Equity Challenges. It might be challenging to combine cloud-based AI solutions with current learning environment. A significant concern can be the digital divide, denoting the disparity in access to technology and the Internet among various communities. Although AI-driven platforms provide novel educational opportunities, it is essential to guarantee that all learners, irrespective of their socioeconomic status, have access to the requisite infrastructure and resources [56].

Many educational institutions might lack the required technological infrastructure, knowledge, or financial means to smoothly use these cutting-edge technologies or have legacy systems. Ensuring compatibility between new AI-powered platforms and existing Learning Management Systems (LMS) and other educational tools is crucial for effective integration. Furthermore, educators need adequate training and professional development to effectively utilize these AI tools in their teaching practices and to adapt their pedagogical approaches to leverage the benefits of personalized learning.

5 Proposed Conceptual Framework: The PACE Model

Building on the knowledge acquired from existing literature, this paper presents the PACE framework – an integrated conceptual model meant to advance the evolution and application of cloud-based artificial intelligence systems for personalized learning. Personalization, architecture, challenges, and evaluation – the PACE model – stands for the essential elements needed to produce scalable, flexible, and successful learning solutions.

Personalization as the initial layer provides tailored educational content and learning pathways as well as pacing that match learner-specific needs and preferences through

AI technologies. The Architecture layer requires the development of a cloud-based system that remains scalable and secure while supporting personalized education experiences. The Challenges layer identifies major technological, educational, ethical, and organizational obstacles which need anticipation and resolution throughout the implementation process. The Evaluation layer focuses on continuous assessment techniques needed to evaluate system performance, learning achievements, user contentment, ethical compliance, and operational efficiency across time periods.

Figure 1 visually presents the PACE model, emphasizing the interplay between its four interconnected layers in a continuous improvement cycle. This diagram highlights how cloud infrastructure enables personalized delivery, while also accounting for systemic challenges and embedding feedback mechanisms for iterative refinement. It serves as a practical guide for stakeholders by offering a cohesive and adaptable strategy for developing AI-enhanced learning environments.

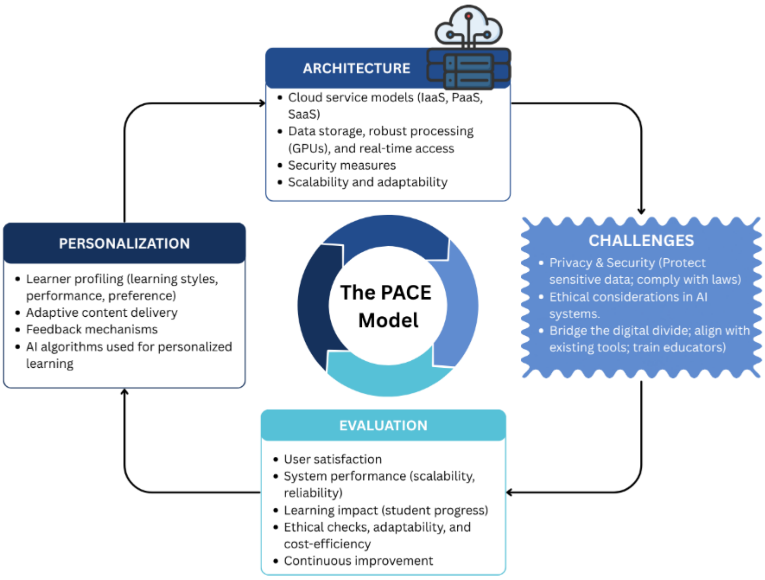

ARCHITECTURE
• Cloud service models (IaaS, PaaS, SaaS)
• Data storage, robust processing (GPUs), and real-time access
• Security measures
• Scalability and adaptability
PERSONALIZATION
• Learner profiling (learning styles, performance, preference)
• Adaptive content delivery
• Feedback mechanisms
• AI algorithms used for personalized learning
The PACE Model
CHALLENGES
• Privacy & Security (Protect sensitive data; comply with laws)
• Ethical considerations in AI systems.
• Bridge the digital divide; align with existing tools; train educators)
EVALUATION
• User satisfaction
• System performance (scalability, reliability)
• Learning impact (student progress)
• Ethical checks, adaptability, and cost-efficiency
• Continuous improvement

Fig. 10. The PACE Model. A cyclical framework integrating Personalization, Architecture, Challenges, and Evaluation for AI-driven personalized learning.

The PACE framework offers institutions, academics, and developers hoping to change educational environments via cloud-based AI solutions a complete and forward-looking strategy by organizing these four interconnected layers.

5.1 Personalization

The personalization component of the PACE model seeks to improve cloud-based AI solutions' effectiveness and student involvement in personalized learning settings. The personalization process examines AI algorithms and techniques that modify the learning process content delivery, pacing, feedback timing and nature as well as the student's learning trajectory. Which algorithm you select will affect how the remaining elements of the framework operate. Important factors in this component consist of evaluating different AI approaches for effectiveness alongside the degree of individualization achieved and how well student preferences and learning styles are integrated.

5.2 Architecture

The Architecture component demonstrates how cloud infrastructure functions as an essential element in delivering personalized learning experiences powered by AI technology. The analysis covers how data storage solutions scale to meet the demands of large datasets from personalized learning platforms, evaluates the processing power required for real-time complex AI algorithms and assesses cloud-based data management strategies for efficiency. Cloud computing models, especially Platform as a Service (PaaS) and Infrastructure as a Service (IaaS), provide the necessary elasticity, flexibility, and cost-effectiveness for managing large-scale deployments [48]. The system architecture needs to provide continuous accessibility and reliability for personalized services while maintaining security for varied student groups.

5.3 Challenges

The third layer of the model, Challenges, addresses the several obstacles and factors that surface during the process of adding cloud-based AI solutions into learning settings. Ensuring compliance with regulations like GDPR and CCPA, using encryption, and creating transparent consent procedures helps one to give data privacy and security top importance. Adopting Fairness, Accountability, and Transparency (FAT) ideas helps one to solve ethical issues such algorithmic bias by guaranteeing objective and explainable AI unbiased. Integration with current Learning Management Systems (LMS) calls both staff training investments and addressing compatibility concerns. Last but not least, by giving every student required technology and digital literacy help, therefore closing the digital divide and ensuring equity and access.

5.4 Evaluation

Even though the proposed framework is still conceptual and its operation in a real-world environment has not been achieved, clear evaluation requirements are crucial to guarantee its effectiveness in future real-world deployments. The Evaluation Layer concentrates on delineating the criteria for assessing the effectiveness and impact of cloud-based AI personalized learning systems. User satisfaction (e.g., feedback from teachers and students via surveys and engagement metrics), system performance (e.g., scalability, latency, and reliability of cloud services), and learning effectiveness (e.g., gains in student performance and skill acquisition) are important evaluation dimensions [57]. In addition, this layer includes constant monitoring for ethical compliance, such as data privacy protection and algorithmic fairness. The capacity to adapt to changing learner needs over time, as well as the cost-effective utilization of cloud resources, are also stressed. By specifying these evaluation methodologies, the framework assures that future implementations are carefully evaluated, refined, and aligned with educational aims and ethical norms.

6 Conclusion

This study has examined the revolutionary potential of cloud computing and AI in education, with a focus on how they can facilitate scalable and customized learning environments. These technologies raise the quality and efficiency of instruction and learning in addition to improving access and adaptability. Their inclusion into education is not optional in a world going more and more digital; it is rather necessary. Proactive adoption of these tools will help institutions to fulfill future educational needs and provide high-quality, learner-centered instruction, so enabling their position.

Building on the existing literature, this paper provides a conceptual framework that maps the complex interplay between AI algorithms, cloud scalability, and their implementation challenges. The conceptual analysis highlights that the choice of AI algorithms will directly impact cloud scalability and shapes the nature of technical, ethical, and pedagogical challenges. On the other hand, cloud scalability presents its own concerns regarding cost, security, and reliability. These dimensions are interrelated, meaning that decisions in one area inevitably impact the others.

The proposed conceptual PACE framework (Personalization, Architecture, Challenges, Assessment) offers a structured approach for designing effective, responsive, and sustainable cloud-based AI systems. It helps stakeholders to identify key tradeoffs and synergies, promoting a more integrated and strategic deployment of AI in learning environments. While grounded in extensive literature, the framework has yet to be tested in a real-world contexts, limiting the capacity to evaluate its practical effectiveness. PACE presents a novel theoretical structure that can be used as a base for future investigations and practical advancements in the education field. This is especially important in the post-pandemic era characterized by swift digital transformation.

Future research should focus on implementing and empirically validating the PACE model in real-world educational settings. Pilot studies that involve both students and instructors could provide valuable insights into how the framework supports personalized learning experiences, improves academic outcomes, and enhances engagement. The model's relevance to real-world educational challenges will be confirmed and refined through empirical validation in a variety of institutional contexts.

References

1. Govea, J., Ocampo Edye, E., Revelo-Tapia, S., Villegas-Ch, W.: Optimization and Scalability of Educational Platforms: Integration of Artificial Intelligence and Cloud Computing. Computers. 12, 223 (2023). https://doi.org/10.3390/computers12110223.
2. Maguvhe, M.O.: Supporting Students Experiencing Barriers to Learning in Inclusive Education Settings: A Critical Requirement for Educational Success. In: Maguvhe, M.O. and Masuku, M.M. (eds.) Using African Epistemologies in Shaping Inclusive Education Knowledge. pp. 375–393. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-31115-4_20.
3. Shoaib, M., Sayed, N., Singh, J., Shafi, J., Khan, S., Ali, F.: AI student success predictor: Enhancing personalized learning in campus management systems. Computers in Human Behavior. 158, 108301 (2024). https://doi.org/10.1016/j.chb.2024.108301.
4. Akavova, A., Temirkhanova, Z., Lorsanova, Z.: Adaptive learning and artificial intelligence in the educational space. E3S Web of Conf. 451, 06011 (2023). https://oi.org/10.1051/e3sconf/202345106011.
5. Wang, S., Wang, F., Zhu, Z., Wang, J., Tran, T., Du, Z.: Artificial intelligence in educatin: A systematic literature review. Expert Systems with Applications. 252, 124167 (2024). https://doi.org/10.1016/j.eswa.2024.124167.
6. Abbes, F., Bennani, S., Maalel, A.: Generative AI and Gamification for Personalized Learning: Literature Review and Future Challenges. SN COMPUT. SCI. 5, 1154 (2024). https://doi.org/10.1007/s42979-024-03491-z.
7. Perifanis, N.-A., Kitsios, F.: Investigating the Influence of Artificial Intelligence on Busines Value in the Digital Era of Strategy: A Literature Review. Information. 14, 85 (2023). https://doi.org/10.3390/info14020085.
8. Mohd Amin, M.R., Ismail, I., Sivakumaran, V.M.: Revolutionizing Education with Artificial Intelligence (AI)? Challenges, and Implications for Open and Distance Learning (ODL). Social Sciences & Humanities Open. 11, 101308 (2025). https://doi.org/10.1016/j.ssaho.2025.101308.
9. Peram, P.: Cloud-Enabled Personalization: Transforming Educational Paradigms through Adaptive Learning Technologies. IJRASET. 12, 1338–1346 (2024). https://doi.org/10.22214/ijraset.2024.64346.

10. Anshari, M., Alas, Y., Guan, L.S.: Developing online learning resources: Big data, social networks, and cloud computing to support pervasive knowledge. Educ Inf Technol. 21, 1663–1677 (2016). https://doi.org/10.1007/s10639-015-9407-3.
11. Ali, M., Khan, S.U., Vasilakos, A.V.: Security in cloud computing: Opportunities and challenges. Information Sciences. 305, 357–383 (2015). https://doi.org/10.1016/j.ins.2015.01.025.
12. Lee, B.-H., Park, S.-H.: A study on the NCS based curriculum for educating Technical Director for VFX industry with Artificial Intelligence. KOSCAS. 63, 417–450 (2021). https://doi.org/10.7230/KOSCAS.2021.63.417.
13. Youssef, H.A.H., Hossam, A.T.A.: Privacy Issues in AI and Cloud Computing in E-commerce Setting: A Review. IJRAI. 13, 37–46 (2023).
14. Alharthi, S., Alshamsi, A., Alseiari, A., Alwarafy, A.: Auto-Scaling Techniques in Cloud Computing: Issues and Research Directions. Sensors. 24, 5551 (2024). https://doi.org/10.3390/s24175551.
15. Sultan, N.: Cloud computing for education: A new dawn? International Journal of Information Management. 30, 109–116 (2010). https://doi.org/10.1016/j.ijinfomgt.2009.09.004.
16. Boateng, O., Boateng, B.: Algorithmic bias in educational systems: Examining the impact of AI-driven decision making in modern education. World J. Adv. Res. Rev. 25, 2012–2017 (2025). https://doi.org/10.30574/wjarr.2025.25.1.0253.
17. Hongli, Z., Wai Yie, L.: AI Solutions for Accessible Education in Underserved Communities. joit. 2024, (2024). https://doi.org/10.61453/joit.v2024no11.
18. Tarnopolsky, O.: PRINCIPLED PRAGMATISM, OR WELL-GROUNDED ECLECTICISM: A NEW PARADIGM IN TEACHING ENGLISH AS A FOREIGN LANGUAGE AT UKRAINIAN TERTIARY SCHOOLS? AE. 5, 5–11 (2018). https://doi.org/10.20535/2410-8286.133270.
19. Zhang, P.: Cloud computing English teaching application platform based on machine learning algorithm. Soft Comput. (2023). https://doi.org/10.1007/s00500-023-08589-9.
20. Ezzaim, A., Dahbi, A., Aqqal, A., Haidine, A.: AI-based learning style detection in adaptive learning systems: a systematic literature review. J. Comput. Educ. (2024). https://doi.org/10.1007/s40692-024-00328-9.
21. Wang, Y.-C., Xue, J., Wei, C., Kuo, C.-C.J.: An Overview on Generative AI at Scale With Edge–Cloud Computing. IEEE Open J. Commun. Soc. 4, 2952–2971 (2023). https://doi.org/10.1109/OJCOMS.2023.3320646.
22. Chetradevee, S.L., Anushka Xavier, K., Jayapandian, N.: Artificial Intelligence Technological Revolution in Education and Space for Next Generation. In: Lecture Notes in Networks and Systems. pp. 371–382. Springer Nature Singapore, Singapore (2022). https://doi.org/10.1007/978-981-19-2130-8_30.
23. Yu, Z., Fang, Y.: The Reformation of College English Teaching under the Background of Smart Education. IJNDE. 4, (2022). https://doi.org/10.25236/IJNDE.2022.040103.
24. Popenici, S.A.D., Kerr, S.: Exploring the impact of artificial intelligence on teaching and learning in higher education. RPTEL. 12, 22 (2017). https://doi.org/10.1186/s41039-017-0062-8.

25. Wang, X.J., Lee, C., Mutlu, B.: LearnMate: Enhancing Online Education with LLM-Powered Personalized Learning Plans and Support. (2025). https://doi.org/10.48550/ARXIV.2503.13340.
26. Chun, J., Zhao, Y., Chen, H., Xia, M.: PlanGlow: Personalized Study Planning with an Explainable and Controllable LLM-Driven System, https://arxiv.org/abs/2504.12452, (2025). https://doi.org/10.48550/ARXIV.2504.12452.
27. Sajja, R., Sermet, Y., Cikmaz, M., Cwiertny, D., Demir, I.: Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education, https://arxiv.org/abs/2309.10892, (2023). https://doi.org/10.48550/ARXIV.2309.10892.
28. Murtaza, M., Ahmed, Y., Shamsi, J.A., Sherwani, F., Usman, M.: AI-Based Personalized E-Learning Systems: Issues, Challenges, and Solutions. IEEE Access. 10, 81323–81342 (2022). https://doi.org/10.1109/ACCESS.2022.3193938.
29. Ghallabi, S., Essalmi, F., Jemni, M., Kinshuk: Learner modeling in cloud computing. Educ Inf Technol. 25, 5581–5599 (2020). https://doi.org/10.1007/s10639-020-10185-5.
30. Sabic, I., Puljiz, H., Smoljo, A.: Personalized Learning in the Croatian National Education System: A Study of AI Implementation in the e-Class Register. SN COMPUT. SCI. 5, 1145 (2024). https://doi.org/10.1007/s42979-024-03515-8.
31. Yang, S.: Construction of Personalized Network Autonomous Learning Platform Based on Digital Empowerment Technology. In: 2024 Second International Conference on Data Science and Information System (ICDSIS). pp. 1–5. IEEE, Hassan, India (2024). https://doi.org/10.1109/ICDSIS61070.2024.10594091.
32. Lokare, V.T., Jadhav, P.M.: An AI-based learning style prediction model for personalized and effective learning. Thinking Skills and Creativity. 51, 101421 (2024). https://doi.org/10.1016/j.tsc.2023.101421.
33. Chen, L., Chen, P., Lin, Z.: Artificial Intelligence in Education: A Review. IEEE Access. 8, 75264–75278 (2020). https://doi.org/10.1109/ACCESS.2020.2988510.
34. Chatti, M.A., Muslim, A.: The PERLA Framework: Blending Personalization and Learning Analytics. IRRODL. 20, (2019). https://doi.org/10.19173/irrodl.v20i1.3936.
35. Waladi, C., Khaldi, M., Lamarti Sefian, M.: Machine Learning Approach for an Adaptive E-Learning System Based on Kolb Learning Styles. Int. J. Emerg. Technol. Learn. 18, 4–15 (2023). https://doi.org/10.3991/ijet.v18i12.39327.
36. Hoti, A.H., Zenuni, X., Hamiti, M., Ajdari, J.: Student Performance Prediction Using AI and ML: State of the Art. In: 2023 12th Mediterranean Conference on Embedded Computing (MECO). pp. 1–6. IEEE, Budva, Montenegro (2023). https://doi.org/10.1109/MECO58584.2023.10154933.
37. Winkler, R., Söllner, M., Leimeister, J.M.: Enhancing problem-solving skills with smart personal assistant technology. Computers & Education. 165, 104148 (2021). https://doi.org/10.1016/j.compedu.2021.104148.
38. Li, S., Li, D.: Research on Personalized Learning Recommendation System based on Machine Learning Algorithm. SCPE. 26, 432–440 (2025). https://doi.org/10.12694/scpe.v26i1.3844.
39. Chen, X.: Design of Personalized Recommendation System for Teaching Resources Based on Cloud Edge Computing. Procedia Computer Science. 243, 826–833 (2024). https://doi.org/10.1016/j.procs.2024.09.099.

40. Khaled, Dr.: Natural Language Processing and its Use in Education. ijacsa. 5, (2014). https://doi.org/10.14569/ijacsa.2014.051210.
41. Lipman, E., Moser, S., Rodriguez, A.: Explaining Differences in Voting Patterns Across Voting Domains Using Hierarchical Bayesian Models, https://arxiv.org/abs/2312.15049, (2023). https://doi.org/10.48550/ARXIV.2312.15049.
42. Bennani, S., Maalel, A., Ben Ghezala, H.: Adaptive gamification in E-learning: A literature review and future challenges. Comp Applic In Engineering. 30, 628–642 (2022). https://doi.org/10.1002/cae.22477.
43. Maier, M.-I., Czibula, G., Oneţ-Marian, Z.-E.: Towards Using Unsupervised Learning for Comparing Traditional and Synchronous Online Learning in Assessing Students' Academic Performance. Mathematics. 9, 2870 (2021). https://doi.org/10.3390/math9222870.
44. Cai, Q., Cui, C., Xiong, Y., Wang, W., Xie, Z., Zhang, M.: A Survey on Deep Reinforcement Learning for Data Processing and Analytics. IEEE Trans. Knowl. Data Eng. 1–1 (2022). https://doi.org/10.1109/TKDE.2022.3155196.
45. Li, C., Ishak, I., Ibrahim, H., Zolkepli, M., Sidi, F., Li, C.: Deep Learning-Based Recommendation System: Systematic Review and Classification. IEEE Access. 11, 113790–113835 (2023). https://doi.org/10.1109/ACCESS.2023.3323353.
46. Riza, L.S., Ajdari, J., Hamiti, M.: Challenges of Adoption of Cloud Computing Solutions in Higher Education: Case Study Republic of Kosovo. In: 2023 46th MIPRO ICT and Electronics Convention (MIPRO). pp. 613–618 (2023). https://doi.org/10.23919/MIPRO57284.2023.10159852.
47. Kumar, P., Rawat, S., Tanwar, J., Gupta, R.: An Analytical Evaluation of Cloud Computing Service model IaaS & PaaS using Market Prospective. In: 2021 International Conference on Technological Advancements and Innovations (ICTAI). pp. 537–540. IEEE, Tashkent, Uzbekistan (2021). https://doi.org/10.1109/ICTAI53825.2021.9673240.
48. Walia, K.: Scalable AI Models through Cloud Infrastructure. ESP-IJACT. 2, 1–7 (2024). https://doi.org/10.56472/25838628/IJACT-V2I2P101.
49. Nama, P.: Integrating AI with cloud computing: A framework for scalable and intelligent data processing in distributed environments. Int. J. Sci. Res. Arch. 6, 280–291 (2022). https://doi.org/10.30574/ijsra.2022.6.2.0119.
50. Fawad, A., Zahoor, M.S., Ellahi, E., Yerasuri, S., Muniandi, B., Balasubramanian, Mr.S.: Efficient Workload Allocation and Scheduling Strategies for AI-Intensive Tasks in Cloud Infrastructures. pst. 47, 82–102 (2023). https://doi.org/10.52783/pst.160.
51. Adams, C., Pente, P., Lemermeyer, G., Rockwell, G.: Ethical principles for artificial intelligence in K-12 education. Computers and Education: Artificial Intelligence. 4, 100131 (2023). https://doi.org/10.1016/j.caeai.2023.100131.
52. Peloquin, D., DiMaio, M., Bierer, B., Barnes, M.: Disruptive and avoidable: GDPR challenges to secondary research uses of data. Eur J Hum Genet. 28, 697–705 (2020). https://doi.org/10.1038/s41431-020-0596-x.
53. Rothstein, M.A., Tovino, S.A.: California Takes the Lead on Data Privacy Law. Hastings Center Report. 49, 4–5 (2019). https://doi.org/10.1002/hast.1042.
54. Janssen, M., Hartog, M., Matheus, R., Yi Ding, A., Kuk, G.: Will Algorithms Blind People? The Effect of Explainable AI and Decision-Makers' Experience on AI-supported Decision-Making in Government. Social Science Computer Review. 40, 478–493 (2022). https://doi.org/10.1177/0894439320980118.

55. Zhai, X., Chu, X., Chai, C.S., Jong, M.S.Y., Istenic, A., Spector, M., Liu, J.-B., Yuan, J., Li, Y.: A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. Complexity. 2021, (2021). https://doi.org/10.1155/2021/8812542.
56. Thakkar, D., Kumar, N., Sambasivan, N.: Towards an AI-powered Future that Works for Vocational Workers. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–13. ACM, Honolulu HI USA (2020). https://doi.org/10.1145/3313831.3376674.
57. Leon, M.: Generative AI as a New Paradigm for Personalized Tutoring in Modern Education. IJITE. 15, 49–63 (2024). https://doi.org/10.5121/ijite.2024.13304.

22. An Effective Architecture for e-Healthcare: A Fog-Based Approach Using iFogSim

Edmira Xhaferra[1], Luciana Toti[2], Amarildo Rista[3],
Esmeralda Hoxha[4] and Oltiana Toshkollari[5]
[12345]AleksanderMoisiu University of Durres, Facuty of Information Technology
edmiraxhaferra@uamd.edu.al[1], lucianaloti@uamd.edu[2],
amarildorista@uamd.edu.al[3], esmeraldahoxha@uamd.edu.al[4],
oltianatoshkollari@uamd.edu.al[5]

Abstract. The combination of advanced sensors and medical signal processing is crucial in making IoT devices more convenient and valuable for human use. It has become more challenging to manage healthcare in terms of workload, time wastage, and proper accommodation as the number of patients has increased, especially in remote regions. To compensate for these concerns, designers are working to create better healthcare solutions using cloud paradigm, fog computing, and wireless sensor networks (WSNs); cloud-based e-health systems impose high latencies to process Big Data, thereby preventing their deployment on a broad scale. The fog computing framework appears more suitable to meet the low latency criteria since it has near-network transmission and storage services. A patient who is in the intensive care unit (ICU) cannot self-report. Pain management systems deploy IoT devices that use bio-sensors to measure surface electromyogram (sEMG), and electrocardiogram (ECG) signals to monitor the pain condition of patients. This paper proposed fog computing architecture to deploy a remote pain management (RPM) system. The proposed architecture is intended to shrink latency and network usage. The efficacy of the presented architecture was evaluated by carrying out simulations on iFogSim and by comparing it with cloud-based architecture. Evidence depicted that the proposed approach significantly reduced latency and network consumption problems convincing its large-scale implementation.

Keywords: Fog Computing, Cloud Computing, iFogSim, e-Healthcare, IoT.

1 Introduction

Nowadays, there are approximately 237.1 million wearable products projected to be in the market, with a revenue of approximately USD 117 billion [1]. This growth has enabled advanced healthcare architectures like cloud-based diabetes monitoring systems using machine learning, demonstrating how IoT data can be leveraged for chronic disease management [2]. Typically, healthcare applications use cloud storage to link the Internet of Things (IoT). As the most feasible approach, the cloud provides computational and storage services to work on broad and diverse volumes of health-related data generated by IoT devices [3]. Recent advancements in machine learning further enhance these systems by enabling predictive analytics and real-time decision-making, as highlighted by Xhaferra and Ismaili (2022) in their exploration of ML's role in healthcare [4]. However, the centralization in the cloud architecture confines the applicability of healthcare systems on a larger scales as the increased volume of data sent and processed remotely can cause high network consumption and latency issues. Conversely, the real-time monitoring of patient's data is crucial in healthcare scenarios. Quality of service (QoS) has a more significant impact on the processing of electrocardiogram (ECG) [5] and electroencephalogram (EEG) data that is entirely thwarted by cloud computing [6]. Fog nodes collect the data from the sensors and provide patient-specific diagnoses. This approach mirrors successful implementations in cardiac care, where fog layers process ECG data via deep neural networks for real-time heart attack alerts [7].

A survey conducted on the various patient groups indicated the rise in the demand for autonomous and remote pain detection systems [8]. Patients have to bear the pain for longer timespans due to delayed diagnosis, lack of expression, and manual measuring systems. Therefore, recent studies explored various automatic pain detection techniques that are facial expressions from the face video [9], facial surface electromyography (sEMG), and physiological signal fusion [10]. Nonetheless, several remote pain detection attempts merge cloud computing with IoT sensors [11,12]. Still, the fulfillment of healthcare QoS requirements is an excellent hurdle for the designers before implementing autonomous pain detection systems.

1.1 Study Objectives

Long latencies and inadequate bandwidth limit large-scale deployment of cloud-oriented RPM applications. In [13], a web-based RPM is presented in which wireless bio-sensing nodes to the web platform through the cloud. The authors developed a wearable face-mask that measures sEMG and ECG activity to identify the intensity of pain. The server collects bio-potential inputs from sensors and then shows the detected information in a real-time web application. However, the direct connection of the sensor node to the server caused long delays that are unbearable in such a time-sensitive environment. Hence, to tackle the inadequacies of the cloud paradigm, a fog computing technique for remote pain monitoring systems is proposed in this paper. The main contributions of this work are outlined as follows:

i. A robust fog-based RPM is presented, which consists of a three-layer hierarchy. Fog layer processes the bioelectric signals and dispatches the pain report to the web servers via the gateways. Fog nodes are located in the middle layer, following the fog computing principle.

ii. The proposed approach minimizes the factors in concern involving price, latency, and network utilization by ensuring real-time monitoring and assistance of patients by quickly detecting and reporting the pain as web alerts. Not only it saves time, but it also discards the unnecessary information at the fog nodes.

iii. Simulations of the proposed model are performed on different scales and compared its results with the cloud computing paradigm to justify the supremacy of the presented paradigm in terms of implementation, cost, and network usage.

Background knowledge of cloud computing, fog computing, and their applications are discussed in section 2. In section 3, the intended framework of the RPM system is elucidated. Section 4 describes the simulations and discusses the findings obtained from the comparison done in this study, while the section 5 summarizes the whole paper

2 Literature review

In the current emerging world of IoT, cloud computing is the most practical approach for developing IoT and large data systems. Challenges like insufficient bandwidth and latency arise because of fast traffic making it difficult for the cloud to attain targeted QoS in the time-sensitive environment [14]. Fog computing utilizes fewer network capacity to have an increased “quality of experience” (QoE). Cloud-fog applications reduced power use by 41% in the theoretical model [15]. Computational resources are allocated near the edge to constrain the resources and fulfill sensitive and complex applications [16,17].

Delays in cloud-centric systems increase when applied globally is a real challenge for healthcare systems [18,19]. Table 1 describes the operational complexity of providing E-health services [20]. In January of 2014, Cisco presented the concept of fog computing to reduce network strain in cloud-based applications [21].

Table 1. Quality of service (QoS) requirements for real-time e-healthcare services.

Real-Time Healthcare Services	**Audio communications**	**Video communications**	**Robotic applications**	**Monitoring services**
Healthcare Applications	Audio	Video	Tele-ultrasonography	Remote monitoring
Type of Media	Audio	Video	Robot-related signals	Bio-signals
Max Delay	<150 ms one-way	<250 ms one-way	<300 ms roundtrip	<300 ms for real-time ECG

Fog computing allocates the cloud services around the network by installing nodes at the edge having limited storage and computational capacities [22,23]. Fog computing aims to offer services with more minor delays among cloud and end devices [24]. Recent work by Xhaferra et al. (2024) demonstrates the synergy of cloud-fog systems with machine learning for diabetes prediction, proposing a framework that optimizes latency and accuracy in healthcare analytics [25]. Some applications of cloud and fog computing in the healthcare monitoring domain are explained below.

In [26], the authors discussed various techniques to interconnect the healthcare applications to build a healthcare system based on mobile cloud computing. Tejaswini et al. [27] established a cloud-based monitoring system for reducing child mortality rate in which tearing is used as a pain indicator. SVM-based pattern classifications and ThingSpeak IoT system along with communication devices, such as mobile phones, are used to interlink physicians and nurses. In [13], the authors suggested a cloud-based RPM in which the cloud acts as a bridge between bio-sensors and the web framework. ECG and sEMG data are collected from wearable masks, which are processed further in a real-time manner to evaluate the patient's condition. Authors in [28] described that mobile cloud computing could be used throughout smart cities to provide healthcare services. They presented UbeHealth, a four-layer cloud-based architecture to provide required QoS without delaying responses and consuming much bandwidth.

Farhani et al. [29] deliberated the issues involved in implementing the cloud-based approaches in healthcare applications and promoted the use of fog architecture to resolve the latency and bandwidth problems. Negash [20] presented a fog-dependent healthcare solution of three levels. Signal detection is one of the cores of the proposed design. The sensor layer compresses obtained data and transmits it to the cloud for further analysis. The authors presented [31] a fog-based identification method for chikungunya virus and a diagnosis system to quickly identify and neutralize the outbreak. Classification is done on user data utilizing a decision tree to identify infection, and the final report is sent to patients via mobiles. Temporal network processing is conducted on the users' data in the neighbourhood to detect the outbreak.

Researchers use "iFogSim" as a simulation toolkit for IoT frameworks in fog and cloud computing architectures. The authors in [32] briefly mentioned the design and simulation measures used in fog-based applications. Dar et al. [33], using the iFogSim, proposed an IoT-based disaster control system and evaluated its cloud and fog-based execution. For reliable and stable car parking [34], fog-based architecture was proposed to use fewer network resources than the cloud as per the iFogSim simulation results. Fang and Ma [35] suggested a module positioning and task scheduling approach using dynamic task processing algorithms evaluated in iFogSim, showing improved performance in terms of power usage.

Fog computing can resolve storage and processing issues and is capable of reducing the load from the cloud. The fog nodes are used to process the pain-concentric details via bio-potential signals from the patients in the proposed scheme. iFogSim is the platform used to evaluate the performance of the proposed framework.

3 Architecture Design

In Fig.1, the proposed three-layer architecture has been demonstrated. In the first layer, the bio-potential sensors are used to track and relay the sEMG and ECG signals preprocessed at the hospitals. The central layer is consisting of fog nodes linked to all the sensors over a Wi-Fi module. All the fog nodes are connected to the third layer based on the cloud to attain extra storage and computation power through a proxy server. A web application is connected to the system to gain access to the pain statistics of the patient for instant monitoring and minimization of the treatment delay. The outline of the suggested architecture is given in this section.

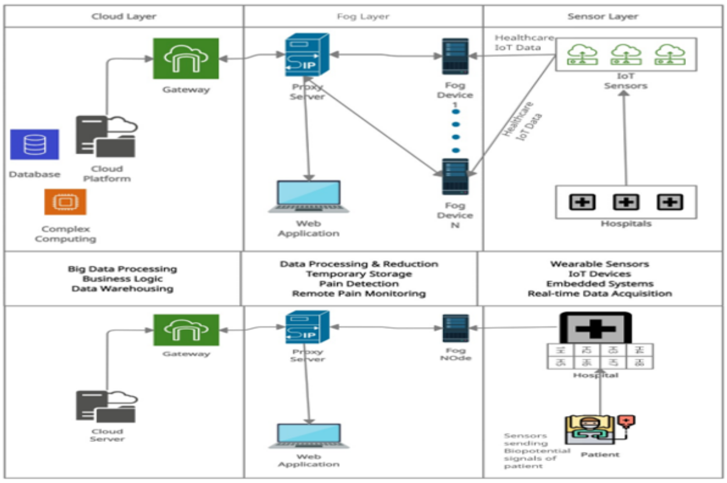


Fig. 1. Three-layer paradigm of the fog-based healthcare system.

I. Sensor layer

The first layer consists of the wearable sensors designed using electrode batteries that acquire ECG and EMG signals from the patients and transmit them to fog devices through Wi-Fi. As constant signal transmission is required, sensors are designed to consume less power to enhance battery life. The device has a sampling rate of 1,000 samples per second in the proposed case to meet the Nyquist criterion.

II. Fog layer

The fog layer resides between the sensor layer and the cloud layer. Fog nodes collect the data from the sensors and provide the patient-specific pain diagnosis. Interoperability is the unique feature of fog computing in which multiple fog nodes perform their local functionalities and share the resources for several IoT devices in connection, as shown in Fig. 2. Latency is not considered in the presence of the interoperability feature of the fog architecture in the proposed approach. Specific indexing is used to differentiate among the patients in a hospital, as shown in fig. 1. After processing the incoming data, the fog node sends the patient's pain report on the associated web application by storing the records inside the fog temporarily and in the cloud for long-term use according to the priority.

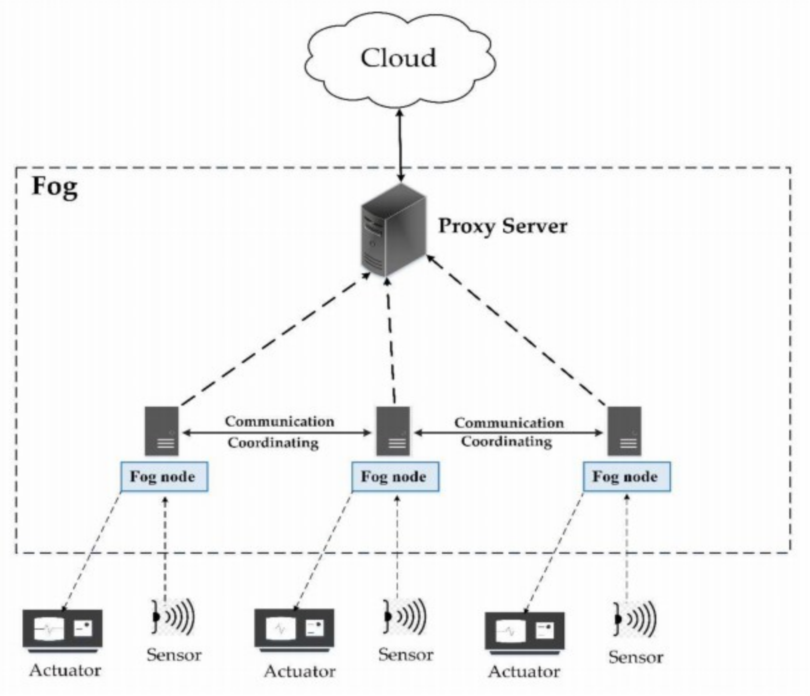


Fig. 2. Interlinkage of cloud server, fog nodes and sensors.

III. Cloud layer

The third layer of the framework consists of the cloud, the prime feature of providing extra storage for record keeping of pain-related information for future use. After using the data of bio-potential signals, fog nodes periodically submits it to the cloud. In this way, the cloud is bypassed through fog nodes that provide computational and temporary storage features at the edge to avoid additional delay.

IV. Analysis

To detect the pain statistics from the incoming ECG and EMG signals, a face action encoding system [36] is employed that characterizes the pain based on the movement of different facial muscles. Afterward, the fog devices enact filtering techniques and digital signal processing techniques to determine the pain [13] remotely. In the case of multi-hospital architecture, multiple sensors are used to detect all the patients, and a single fog node is used to manipulate the incoming signal in the proposed approach. Fog nodes transmit and regularly update the pain information on the linked web application as well as in the cloud. This direct transmission of pain statistics from the fog node to the webserver minimizes the network usage and latency problems. Fig. 3 illustrated the network design for operating the proposed framework in several hospitals.

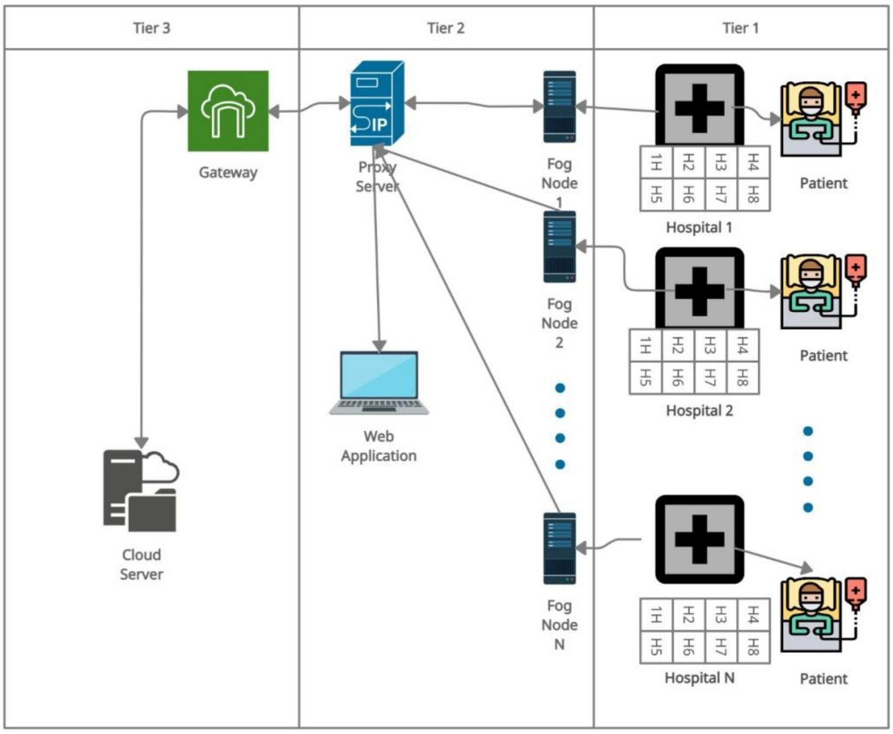


Fig. 3. The healthcare paradigm of RPM for several hospitals.

In multiple hospitals, fog nodes of all the hospitals simultaneously update the cloud at once, increasing the chances of latency and more network consumption. In the beginning, all the sensors are set to acquire EMG and ECG readings. After bio-potential readings are obtained, the facial pattern extraction is executed utilizing root mean square (RMS) attribute visualization and extraction approaches. Eventually, the signal is divided into segments and the dimensionality is reduced [37,38]. Afterward, the processed signals are sent to the application by a web server in order to monitor the pain remotely. Finally, the data are sent to the cloud. Fig. 4 illustrates the critical processes involved in the remote pain detection system.

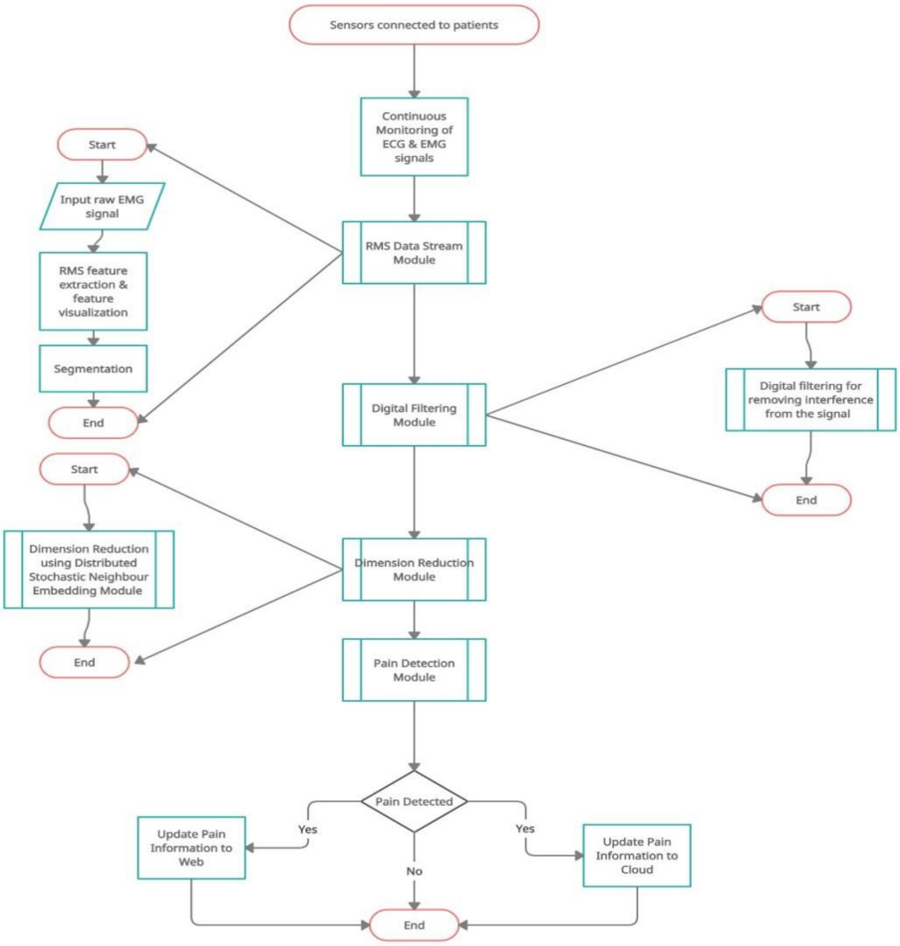

Sensors connected to patients
Continuous Monitoring of ECG & EMG signals
RMS Data Stream Module
Start
Input raw EMG signal
RMS feature extraction & feature visualization
Segmentation
End
Digital Filtering Module
Start
Digital filtering for removing interference from the signal
End
Start
Dimension Reduction using Distributed Stochastic Neighbour Embedding Module
End
Dimension Reduction Module
Pain Detection Module
Pain Detected
Yes
Yes
No
Update Pain Information to Web
Update Pain Information to Cloud
End

Fig. 4. Flowchart of the suggested fog-based pain monitoring system.

4 Simulations and Results

iFogSim method is used to simulate and evaluate our scenarios. New variables are generated during simulations. Four hospitals are involved, and a single fog node is allocated to each of these. Patients with four bio-potential sensors are initially connected to each fog node, further linked with cloud and web applications. The simulation environment is employed with the gradual increase in sensor nodes to evaluate latency and network usage. Table 2 shows the configurations of our simulations.

Table 2. Bio-potential sensor configuration in iFogSim simulator.

CPU Length	Network Length	Sensor Detecting Interval
1200 million instructions	22000 bytes	25 milliseconds

The topology generated to evaluate fog computing is based on FCFS scheduling and visualized in Fig. 5. Four sensor nodes are generated and are all connected to four fog nodes focusing on observing latency and network efficiency. We created an RMS module to collect the bio-potential signal of the patients. The electronic filtering and elimination module is integrated into the pain detection fog nodes. Tasks are usually represented as tuples, generated by sensors, and managed by application components on fog nodes, in iFogSim. Each Virtual Machine (VM) has a particular tuple. These components use fog resources to execute their computations. FCFS (first come, first serve) scheduling system used in our simulations.

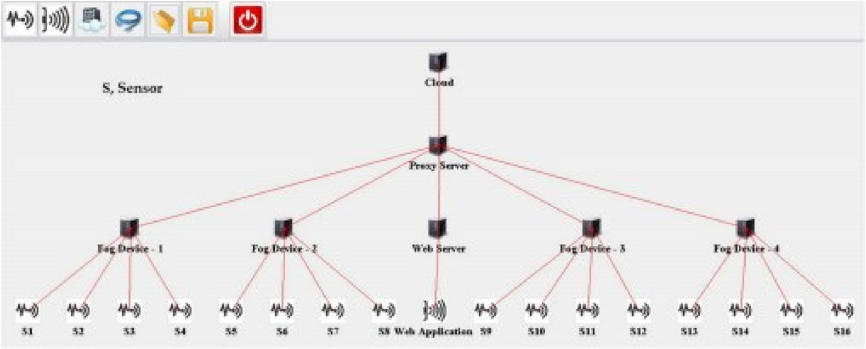


Fig. 5. iFogSim architecture for the proposed fog-centric RPM.

To evaluate the latency and network consumption, the number of sensors is gradually increased in fog-based computing. Performance of the systems starts decreasing due to the increase in the load at a particular node, thus enhancing the latency and network consumption. In a cloud-based scenario, all the sensor nodes and web applications are directly connected to the cloud, as shown in Fig. 6, which resulted in longer delays and power usage compared to fog computing. Table 3 portrays the values of parameters used for the implementations of both cloud and fog computing.

Table 3. Description of parameters employed for cloud- and fog-based implementations.

Parameters	Cloud	Proxy Server	Web Server	Fog Node	Sensor Node
Level	0	1	2	2	3
Rate per MIPS	0.01	0.0	0.0	0.0	0.0
RAM	40000	4000	4000	4000	1000
Idle Power	16 x 83.25	83.43	83.43	83.43	83.44
Download Bandwidth (MB)	10000	10000	10000	10000	-
CPU Length	44800	2800	2800	2800	500
Uplink Bandwidth (MB)	100	10000	10000	10000	10000
Busy Power	16 x 103	107.339	107.339	107.339	87.53

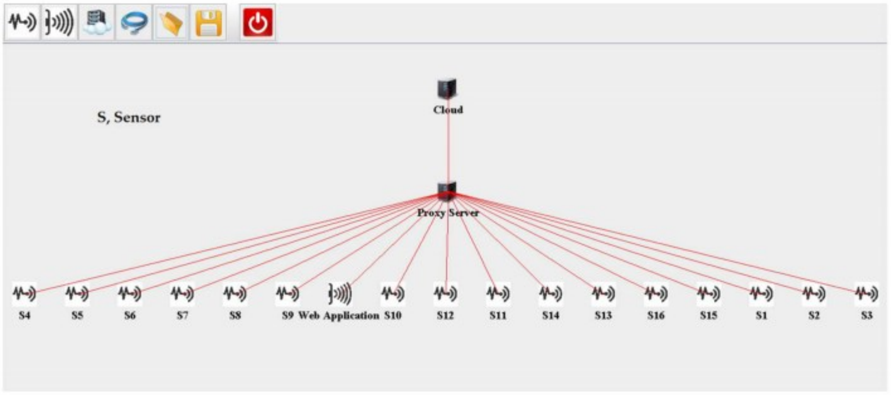


Fig. 6. iFogSim setting for cloud-based remote pain detection system.

4.1 Latency

To operate appropriately in real-time environments on a large scale, the latency of the healthcare systems must be lowered. The primary advantage of fog architecture is that it encourages local execution of tasks to minimize cloud access, reducing latency. Fig. 7 assesses the latencies of all fog and cloud-based simulations. It can be observed that latency of the cloud increased with an increment in the number of sensors from 10 to 60, while on the other hand, in fog computing, the response delays remained constant with an increase in sensors from 16 to 48. After that, the latency of the fog module abruptly increased due to network overloading that can be reduced using an extra fog node.

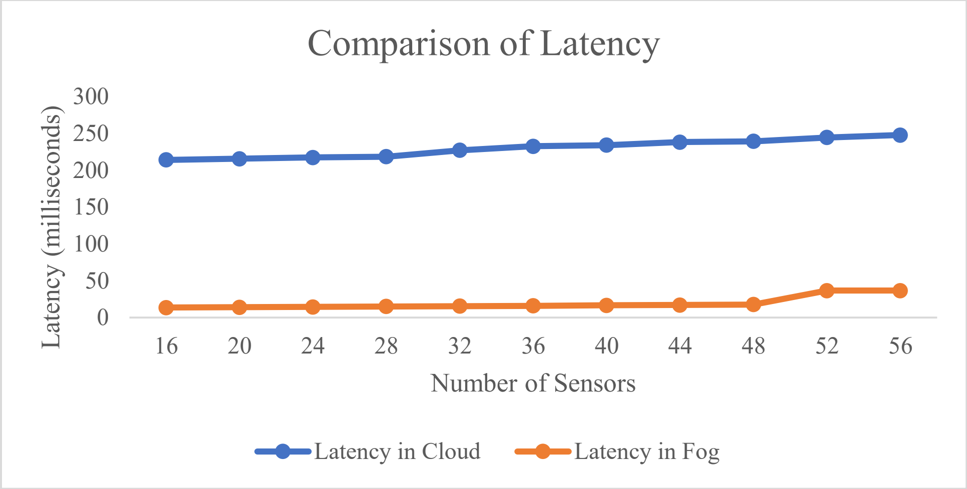


Fig.7. Comparison of Fog and Cloud implementations of proposed system in terms of latency.

4.2 Network Consumption

Consumption of the network depends upon the number of patients. An increment in the quantity of patients increases the incoming data that affects the network's overall bandwidth. Fig. 8 compares the network consumption of both fog and cloud-based implementations. It can be seen that the network consumption in the cloud increased rapidly with an increment in the number of sensors because all the sensors are connected to the same server to fulfill their requests. On the contrary, in the fog-based demonstration, each fog node only has to respond to those sensors to whom it is linked. Fog computing architecture in remote pain monitoring will enable quick extraction of the required report anytime, also reducing the treatment delays.

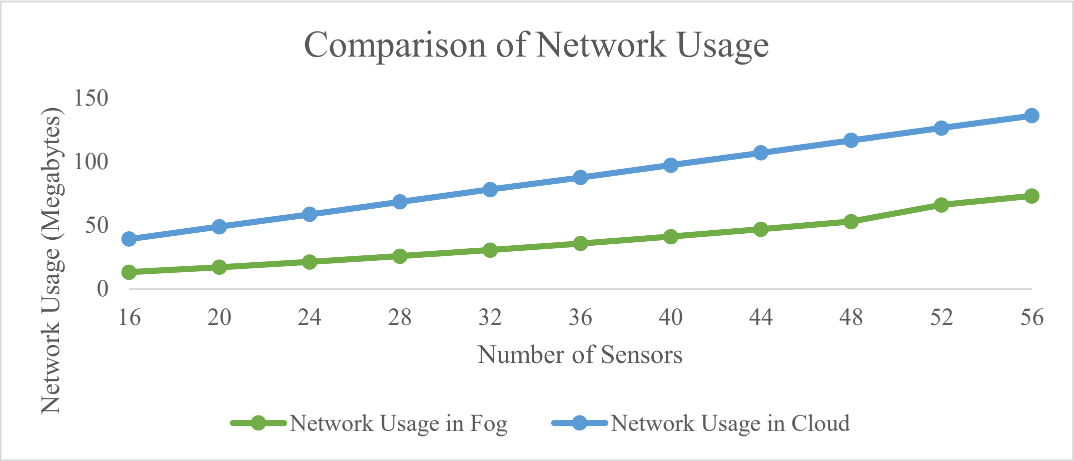


Fig. 8. Comparison of Cloud and Fog implementations of proposed model in terms of Network Consumption.

4.3 Execution Cost

Fog enables data transmission at the network edge. "Fog devices" collect and manage the bio-potential data. The data that need higher resources than the available at the fog node is transported to the cloud. The actual cost needed for the execution of application modules involves the execution cost [39,40]. Fig. 9a depicts that execution cost at cloud is much greater than that of fog-based employment with an increase in sensors. The reason is that the data is reduced for the cloud to store and compute with the involvement of fog nodes. Fig. 9b shows that by increasing the number of sensors in fog-based applications, the graph of reduction cost increases.

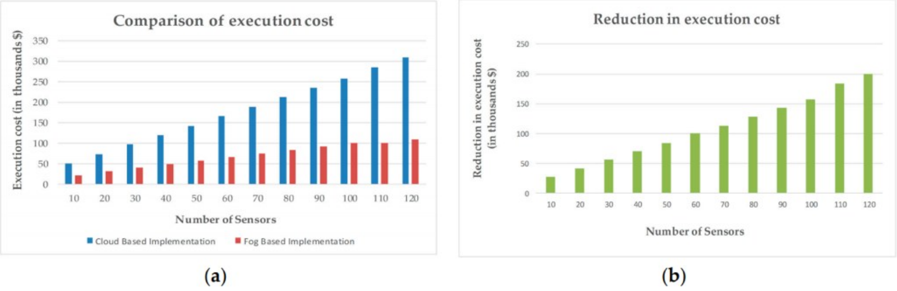


Fig. 9. (a) Comparative Analysis of Execution Cost of cloud and fog based implementation. (b) The proposed approach and reduction in cost execution.

5 Findings

We presented an improvement model that enables autonomous pain detection based on fog computing architecture. No other application has been presented yet who assimilates fog paradigm and remote pain monitoring to the best of our experience. All of the available resources on the internet deal with the cloud implementation for remotely monitoring patient's pain statistics. Different simulations are executed to make comparison between the cloud and fog-based architectures and evaluate the usefulness of fog based approach in RPM. By evaluating fog and cloud architectures in terms of cost, latency, and network consumption, it is clear that the fog model can be a suitable option to implement in the healthcare domain for the remote pain detection system. It can be observed that fog implementation outperforms cloud execution in all performance metrics meeting the QoS demands.

In [13,41,42], all the proposed frameworks adopted cloud servers to store and process data originating from patients. Furthermore, clouds are linked to the web and mobile applications to provide monitoring services remotely. Cloud computing provides the services in a centralized manner that primarily causes latency and bandwidth issues in the deployment of RPM. Fog computing presents a new layer between sensors and cloud to provide cloud-like services near the edge that is more suitable than the cloud for time-sensitive domains such as healthcare. Table 4 represents the comparison of existing healthcare systems with our proposed fog-based RPM. It can be seen that our proposed scheme outperformed all other existing models in terms of response time, cost of execution, and network consumption fulfilling most of the QoS requirements.

Table 4. Comparison of existing techniques with our proposed model.

Reference	[13]	[27]	[41]	[43]	[44]	**Our study**
Paradigm	Cloud	Cloud	Cloud	Cloud	Cloud	Fog
Monitoring	Pain	Pain	Patient	Pain	Health	Pain
Response Time	Moderate	Moderate	Moderate	Moderate	Moderate	Minimum
Cost	High	High	High	High	High	Low
Network Consumption	High	High	High	High	High	Low

6 Conclusion

To provide healthcare for each patient, the medical industry is implementing online RPMs. Numerous cloud-based healthcare systems are available in the market. Such systems cannot be implemented due to delayed responses in the critical times fog-based computing technology provides the services at the edge to make it quick. Thus, a fog computing-based approach is proposed in this study that gathers and processes sEMG signals for pain detection. In this model, pain-related facts would be made accessible to patients across the internet, helping to expedite treatment. Simulations on different scales resulted that the suggested fog-based solution is beneficial in latency reduction and in minimizing networking costs as compared to the cloud.

The proposed model restricts the usage of a single fog device because the greater the number of patients, the more computing capacity needed. Since load balancing is needed to keep the device efficient, our upcoming work involves exploring load balancing problems in fog computing and creating an appropriate solution. Furthermore, the proposed framework is limited to pain monitoring only, and we are eager to incorporate a real-time fog computing system capable of monitoring the patient's overall health from different biostatics.

References

1. S. Shukla, M. F. Hassan, M. K. Khan, L. T. Jung, and A. Awang, "An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment," PLoS One, vol. 14, no. 11, p. e0224934, Nov. 2019, doi: 10.1371/journal.pone.0224934.
2. Xhaferra, E., Ismaili, F., & Chaushi, A. (2023, May). Cloud-Based Healthcare Architecture for Diabetes Patients Using Machine Learning. In International Scientific Conference on Business and Economics (pp. 793-800). Cham: Springer Nature Switzerland.
3. C. S. Nandyala and H.-K. Kim, "From Cloud to Fog and IoT-Based Real-Time U-Healthcare Monitoring for Smart Homes and Hospitals," Int. J. Smart Home, vol. 10, no. 2, pp. 187–196, 2016, doi: 10.14257/ijsh.2016.10.2.18.
4. Xhaferra, E., & Ismaili, F. (2022, June). The Role of Machine Learning in the Healthcare Sector: A Roadmap to the Potential Prospects. In 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-8). IEEE.
5. T. N. Gia, M. J. A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog Computing in Healthcare Internet of Things : A Case Study on ECG Feature Extraction," 2015, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.51.
6. Y. Shi, G. Ding, H. Wang, H. Eduardo Roman, and S. Lu, "The fog computing service for healthcare," in 2015 2nd International Symposium on Future Information and

Communication Technologies for Ubiquitous HealthCare, Ubi-HealthTech 2015, Aug. 2015, pp. 70–74, doi: 10.1109/Ubi-HealthTech.2015.7203325.

7. Xhaferra, E., & Cina, E. (2022, November). A fog-health architecture for early alarming system of heart attack: A deep neural network-based approach. In 2022 International Interdisciplinary Conference on Mathematics, Engineering and Science (MESIICON) (pp. 1-6). IEEE
8. C. R. Jonassaint, N. Shah, J. Jonassaint, and L. De Castro, “Usability and Feasibility of an mHealth Intervention for Monitoring and Managing Pain Symptoms in Sickle Cell Disease: The Sickle Cell Disease Mobile Application to Record Symptoms via Technology (SMART),” Hemoglobin, vol. 39, no. 3, pp. 162–168, Jun. 2015, doi: 10.3109/03630269.2015.1025141.
9. P. Lucey et al., “Automatically detecting pain in video through facial action units,” IEEE Trans. Syst. Man, Cybern. Part B Cybern., vol. 41, no. 3, pp. 664–674, Jun. 2011, doi: 10.1109/TSMCB.2010.2082525.
10. M. Kächele, P. Werner, A. Al-Hamadi, G. Palm, S. Walter, and F. Schwenker, “Bio-visual fusion for person-independent recognition of pain intensity,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 9132, pp. 220–230, doi: 10.1007/978-3-319-20248-8_19.
11. M. S. Hossain and G. Muhammad, “Cloud-Assisted Speech and Face Recognition Framework for Health Monitoring,” Mob. Networks Appl., vol. 20, no. 3, pp. 391–399, Jun. 2015, doi: 10.1007/s11036-015-0586-3.
12. Y. Zhong and L. Liu, “Remote neonatal pain assessment system based on internet of things,” in Proceedings - 2011 IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, iThings/CPSCom 2011, 2011, pp. 629–633, doi: 10.1109/iThings/CPSCom.2011.116.
13. G. Yang et al., “IoT-Based Remote Pain Monitoring System: From Device to Cloud Platform,” IEEE J. Biomed. Heal. Informatics, vol. 22, no. 6, pp. 1711–1719, Nov. 2018, doi: 10.1109/JBHI.2017.2776351.
14. N. L. S. da Fonseca and R. Boutaba, Cloud Services, Networking, and Management. Wiley-IEEE Press, 2015.
15. S. Sarkar and S. Misra, “Theoretical modelling of fog computing: A green computing paradigm to support IoT applications,” IET Networks, vol. 5, no. 2, pp. 23–29, Mar. 2016, doi: 10.1049/iet-net.2015.0034.
16. A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, “Fog Computing for the Internet of Things: Security and Privacy Issues,” IEEE Internet Comput., vol. 21, no. 2, pp. 34–42, Mar. 2017, doi: 10.1109/MIC.2017.37.
17. Y. C. P. Chang, S. Chen, T. J. Wang, and Y. Lee, “Fog Computing Node System Software Architecture and Potential Applications for NB-IoT Industry,” in Proceedings - 2016 International Computer Symposium, ICS 2016, Feb. 2017, pp. 727–730, doi: 10.1109/ICS.2016.0150.

18. A. M. Rahmani et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," Futur. Gener. Comput. Syst., vol. 78, pp. 641–658, Jan. 2018, doi: 10.1016/j.future.2017.02.014.
19. G. Lee, W. Saad, and M. Bennis, "An online optimization framework for distributed fog network formation with minimal latency," IEEE Trans. Wirel. Commun., vol. 18, no. 4, pp. 2244–2258, Apr. 2019, doi: 10.1109/TWC.2019.2901850.
20. L. Skorin-Kapov and M. Matijasevic, "Analysis of QoS requirements for e-Health services and mapping to evolved packet system QoS classes," Int. J. Telemed. Appl., 2010, doi: 10.1155/2010/628086.
21. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop, 2012, pp. 13–15, doi: 10.1145/2342509.2342513.
22. Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," IEEE Access, vol. 5, pp. 25445–25454, Oct. 2017, doi: 10.1109/ACCESS.2017.2766923.
23. R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption," in IEEE Internet of Things Journal, Dec. 2016, vol. 3, no. 6, pp. 1171–1181, doi: 10.1109/JIOT.2016.2565516.
24. R. Nishtala, P. Carpenter, V. Petrucci, and X. Martorell, "Hipster: Hybrid Task Manager for Latency-Critical Cloud Workloads," in Proceedings - International Symposium on High-Performance Computer Architecture, May 2017, pp. 409–420, doi: 10.1109/HPCA.2017.13.
25. Xhaferra, E. D. M. I. R. A., Ismaili, F. L. O. R. I. J. E., Cina, E. L. D. A., & Mitre, A. N. I. L. A. (2024). A conceptual framework for leveraging cloud and fog computing in diabetes prediction via machine learning algorithms: A proposed implementation. J. Theor. Appl. Inf. Technol, 102, 6004-6026
26. F. Muheidat, L. Tawalbeh, and H. Tyrer, "Context-Aware, Accurate, and Real Time Fall Detection System for Elderly People," in Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018, Apr. 2018, vol. 2018-January, pp. 329–333, doi: 10.1109/ICSC.2018.00068.
27. S. Tejaswini, N. Sriraam, and G. C. M. Pradeep, "Cloud-Based Framework for Pain Scale Assessment in NICU- A Primitive Study with Infant Cries," in 2018 IEEE 3rd International Conference on Circuits, Control, Communication and Computing, I4C 2018, Oct. 2018, doi: 10.1109/CIMCA.2018.8739712.
28. T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, "UbeHealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities," IEEE Access, vol. 6. Institute of Electrical and Electronics Engineers Inc., pp. 32258–32285, Jun. 11, 2018, doi: 10.1109/ACCESS.2018.2846609.
29. B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," Futur. Gener. Comput. Syst., vol. 78, pp. 659–676, Jan. 2018, doi: 10.1016/j.future.2017.04.036.

30. B. Negash et al., "Leveraging fog computing for healthcare IoT," in Fog Computing in the Internet of Things: Intelligence at the Edge, Springer International Publishing, 2017, pp. 145–169.
31. S. K. Sood and I. Mahajan, "A Fog-Based Healthcare Framework for Chikungunya," IEEE Internet Things J., vol. 5, no. 2, pp. 794–801, Apr. 2018, doi: 10.1109/JIOT.2017.2768407.
32. H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," Softw. Pract. Exp., vol. 47, no. 9, pp. 1275–1296, Sep. 2017, doi: 10.1002/spe.2509.
33. B. K. Dar, M. A. Shah, H. Shahid, and A. Naseem, "Fog computing based automated accident detection and emergency response system using Android smartphone," in 2018 14th International Conference on Emerging Technologies, ICET 2018, Jan. 2019, doi: 10.1109/ICET.2018.8603557.
34. K. S. Awaisi et al., "Towards a Fog Enabled Efficient Car Parking Architecture," IEEE Access, vol. 7, pp. 159100–159111, 2019, doi: 10.1109/ACCESS.2019.2950950.
35. J. Fang and A. Ma, "IoT Application Modules Placement and Dynamic Task Processing in Edge-Cloud Computing," IEEE Internet Things J., pp. 1–1, Jul. 2020, doi: 10.1109/jiot.2020.3007751.
36. P. Friesen, E.; Ekman, "Facial action coding system: A technique for the... - Google Scholar," 1978. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Facial+action+coding+system%3A+A+technique+for+the+measurement+of+facial+movement&btnG= (accessed Apr. 23, 2021).
37. C. R. García-Alonso, L. M. Pérez-Naranjo, and J. C. Fernández-Caballero, "Multiobjective evolutionary algorithms to identify highly autocorrelated areas: The case of spatial distribution in financially compromised farms," Ann. Oper. Res., vol. 219, no. 1, pp. 187–202, 2014, doi: 10.1007/s10479-011-0841-3.
38. M. T. Sadiq et al., "Motor imagery EEG signals classification based on mode amplitude and frequency components using empirical wavelet transform," IEEE Access, vol. 7, pp. 127678–127692, 2019, doi: 10.1109/ACCESS.2019.2939623.
39. D. Rahbari and M. Nickray, "Scheduling of fog networks with optimized knapsack by symbiotic organisms search," in Conference of Open Innovation Association, FRUCT, Jan. 2018, pp. 278–283, doi: 10.23919/FRUCT.2017.8250193.
40. D. Rahbari and M. Nickray, "Low-latency and energy-efficient scheduling in fog-based IoT applications," TURKISH J. Electr. Eng. Comput. Sci., vol. 27, no. 2, pp. 1406–1427, Mar. 2019, doi: 10.3906/elk-1810-47.
41. G. J. Bharat Kumar, "Internet of Things (IoT) and Cloud Computing based Persistent Vegetative State Patient Monitoring System: A remote Assessment and Management," in Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems, CTEMS 2018, Dec. 2018, pp. 301–305, doi: 10.1109/CTEMS.2018.8769175.
42. A. I. Siam, A. A. Elazm, N. A. El-Bahnasawy, G. El Banby, and F. E. Abd El-Samie, "Smart Health Monitoring System based on IoT and Cloud Computing." Accessed: Apr. 23, 2021. [Online]. Available: http://www.iceem-mu.org.

43. P. Casti et al., “A Personalized Assessment Platform for Non-invasive Monitoring of Pain,” in IEEE Medical Measurements and Applications, MeMeA 2020 - Conference Proceedings, Jun. 2020, doi: 10.1109/MeMeA49120.2020.9137138.
44. M. Al-khafajiy et al., “Remote health monitoring of elderly through wearable sensors,” Multimed. Tools Appl., vol. 78, no. 17, pp. 24681–24706, Sep. 2019, doi: 10.1007/s11042-018-7134-7.

Section 4: Software Engineering and Development

23. A Framework for AI-Assisted Software Development Life Cycle for Generative AI Applications

Stanislav Ustymenko[1]and Abhishek Phadke[2]

[1] Saint Leo University, Saint Leo, FL 33574, USA
[2] Christopher Newport University, 1 Avenue of the Arts, Newport News, VA 23606

stanislav.ustymenk@saintleo.edu

Abstract. The proliferation of Large Language Models led to advances in tools and technologies leveraging Generative AI. Researchers and practitioners in the software engineering domain have demonstrated profound use cases that promise to boost productivity at multiple stages of the software development lifecycle (SDLC), and the resulting advances have an accelerating impact on the industry. At the same time, there is a growing body of knowledge and best practices on developing specific applications of Generative AI models aimed at satisfying user demands while mitigating known challenges to usability and quality. Creating AI artifacts for production use under time and budget constraints gave rise to the AI Development Lifecycle parallelling SDLC. Developing reliable, explainable AI-enhanced information systems that efficiently combine traditional software and LLM technology involves enhancing SDLC with AI support while following the AI Development Lifecycle, with many activities overlapping. In this study, we analyze the available literature and outline the framework for an AI-aware development model suitable for developing Generative AI-enabled software systems. We illustrate the framework with a use case of planning Generative AI-enabled instructional tools in a Computer Science discipline. We identify gaps in existing literature on this topic and propose areas for further development.

Keywords: Software Development Lifecycle, Generative AI, Large Language Models, Software Engineering.

1 Introduction

Software development was a major driver of economic growth and societal change for several decades. Correspondingly, the importance of software engineering can not be overstated [1]. The industry has developed approaches and models aimed at conceiving, planning, designing, coding, validating, and deploying computer software that satisfies the needs of various stakeholders, efficiently uses time, human talent, and financial resources, effectively manages quality, and adheres to ethical standards of practice. While imperfect, these practices and models are key to the success of technological projects and businesses. Over time, software engineering approaches evolved to accommodate demands for software and progress in available supporting technology.

The Software Development Lifecycle is among the key software engineering concepts [2]. SDLC provides a structured framework that divides the development process into iterative phases: planning, requirements gathering, design, implementation, testing, deployment, and maintenance. The goal is to systematically address stakeholder needs while mitigating risks like scope creep, budget overruns, and security vulnerabilities. The SDLC ensures efficient resource allocation and high-quality outcomes by emphasizing early defect detection, continuous validation, and alignment between technical and business goals. Modern SDLC methodologies integrate security practices and iterative feedback loops to adapt to evolving technological and ethical demands, reinforcing its role as a cornerstone of reliable, scalable software engineering.

The ongoing boom in AI adoption is primarily fueled by Generative AI tools that use Large Language Models (LLMs). LLMs, like Generative Pretrained Transformer 3 (GPT-3) and Generative Pre-trained Transformer 4 (GPT-4), use unsupervised pre-training on large text corpora to overcome data labeling difficulties. LLMs predict word probabilities given context, utilizing transformer architecture and achieving near-human quality text generation. They excel in classic Natural Language Processing (NLP) tasks and can be fine-tuned for specific applications. GPT-3, with 175 billion parameters trained on 500 billion tokens, demonstrates the scale of LLM capabilities. Other LLMs like Google's BERT and Meta's Llama offer similar capabilities, contributing to the advancement of AI in modern information systems.

Ever since the dawn of computing technology, researchers envisioned computers performing tasks associated with intelligence [3]. Recent advances in generative transformer deep learning technology [4] enabled the creation of pre-trained, large language models (LLMs) that excel in language understanding tasks and can be easily instructed to imitate humans in a wide array of intelligent work. Despite stubborn challenges with trustworthiness and bias and the high computational cost of training and inference, leading LLMs quickly gained adoption. Their future impacts are speculated to be even more profound [5]. Accordingly, there is a vigorous debate on AI's impact on society [6] and specific industries.

In this article, we argue that the impact of unleashing Generative AI on the software development lifecycle is twofold. First, applying Generative AI tools (GAITs) to various tasks within the SDLC will change how we engineer our software. Second, more software will integrate machine learning and Generative AI systems, making AI engineering activities part of the overall development process. We identify these influences and outline the contours of the emerging framework of the AI-assisted SDLC for these applications.

2 Generative AI in SDLC

2.1 Industry Adoption of GAITs

Software engineers were on the cutting edge of LLM tool adoption. As access to LLM-powered chatbots became widespread, developers started experimenting with their ability to generate code from natural language prompts [7]. While the quality of generated code was inconsistent, the speed and ease of these tools proved attractive to users. As early as 2021, OpenAI produced its Codex model optimized for code by fine-tuning its GPT-3 LLM with code available on GitHub. Code assistants like GitHub Copilot [8], Cursor AI, and Replit's Ghostwriter AI seek to integrate code generation and AI-assisted coding into everyday developer workflow.

There are emerging Generative AI tools that are aimed at other phases of SDLC as well. [9] provides comprehensive (at the time of writing) mapping of these tools to phases of Waterfall SDLC. He finds production tools and prototypes emerging related to all the classic SDLC phases: Requirements gathering and analysis, Design, Implementation and Coding, Testing, and Maintenance. These phases correspond to development activities in all modern SDLC models, including iterative and agile development.

Requirement gathering and analysis lends itself to text summarization and understanding capabilities of general LLM-based tools such as ChatGPT and CodeLlama [10] to generate software requirements specification documents. Other tools showed promise as well. For example, in [11], users can analyze meeting notes and generate high-level requirements using provided templates. [9] gives one use case study of this tool. Google's NotebookLM [12] provides a notes repository that enables natural language queries, question-answering, and summarization grounded in uploaded documents. Such capabilities lend themselves well to the requirements engineering phase of the Software Development Lifecycle.

In the design phase, software engineers develop macro and micro architecture for the system, dividing the task into manageable parts. LLMs can be helpful here as well. Similar to requirements analysis, general AI tools like ChatGPT, pre-trained on large corpora that include technical documents, have proven capable of creating initial drafts of the documents [13]. In addition, specialized tools have been created to create and maintain common software design diagrams (class diagrams, user stories, sequence diagrams, and others from the UML toolkit). Tools like DiagramGPT [14] and ChartGPT [15] can create diagrams from requirements descriptions, design notes, and code. They can be a valuable addition to software engineering workflows in software developer teams and improve design clarity and documentation quality.

Several tools leverage AI, including Large Language Models (LLMs), to aid in the software testing phase. GitHub Copilot, Tabnine, and Codeium are identified as

Generative AI tools (GAITs) capable of generating unit test code and test cases. These tools have been demonstrated to create test scenarios covering normal and edge case conditions for varying levels of code complexity [16]. Beyond code generation, Generative AI-based tools can also assist by generating test case ideas and helping developers ensure more comprehensive testing. In the future, using such AI-driven testing, sometimes called autonomous testing, promises benefits such as accelerated and uninterrupted testing, full automation, and enhanced testing capabilities.

During the Maintenance phase, GenAI enables predictive analytics by analyzing historical data to foresee system failures, while adaptive scaling dynamically optimizes cloud resources [17], [18]. These advancements enhance operational efficiency, with studies highlighting AI's role in minimizing errors and improving resource management across the software lifecycle [19].

Generative AI solutions have the potential to affect every element of the SDLC. In practice, however, the developers gravitate to a few specific use cases. In one survey [20], 72% of the 500 software engineers surveyed at a large company reported using Generative AI. Of these engineers, 81% reported using the tools for code generation, 67% for documentation, 45% for testing, and 32% for debugging. Despite reporting issues with accuracy, the majority reported productivity gains from using GAITs. This study agrees with the prevailing themes in the literature and reveals gaps in the use of GAITs in other SDLC phases.

2.2 Effect of GAITs on Agile SDLC

The use of GAITs promises to streamline software engineering processes and facilitate adherence to the best practices. Application of the SDLC practices takes different forms according to the characteristics of the project, and different process models arose to accommodate typical. For many practitioners, Agile principles [21] and process models like Extreme Programming [22] and SCRUM [23]. Using GAITs has the potential to enable and transform these processes. Generative AI (GenAI) holds transformative potential for realizing the Agile Manifesto's core values by automating bottlenecks, enhancing collaboration, and fostering adaptability. Below, we explore its impact on each principle and speculate on emerging practices in software engineering.

Individuals and Interactions Over Processes and Tools. GenAI reduces reliance on rigid workflows by streamlining communication and decision-making. For instance, AI-powered tools like sentiment analysis systems can monitor team interactions to identify friction points, enabling Scrum Masters to address morale or misalignments proactively [24]. Developers can leverage prompt-driven simulations to rapidly prototype logic (e.g., "Generate a REST API mockup for a payment gateway"), bypassing boilerplate coding and focusing on collaborative refinement. This shifts team energy from procedural adherence to creative problem-solving.

Working Software Over Comprehensive Documentation. GenAI accelerates the delivery of functional software by automating repetitive tasks. Beyond Test-Driven Development (TDD), AI can refactor legacy code by analyzing dependencies and generating optimized alternatives. For example, a prompt like "Modify this Java class to use reactive streams" could yield refactored code with minimal manual intervention. Similarly, AI agents can integrate new frameworks (e.g., migrating from Angular to React) by generating compatibility layers or identifying deprecated methods, reducing integration risks.

Customer Collaboration Over Contract Negotiation A fine-tuned chatbot trained on historical user feedback and product data could act as an "embedded customer" within Agile teams. Such a bot would simulate real-user interactions during sprint reviews, challenging assumptions (e.g., "As a frequent traveler, I need faster checkout") and prioritizing feature requests. This aligns with Agile's emphasis on continuous feedback, bridging the gap between static requirements and evolving user needs.

Responding to Change Over Following a Plan. GenAI enhances adaptability through predictive analytics and dynamic resource allocation. For instance, AI-driven tools could forecast sprint delays by analyzing commit histories or simulate the impact of adopting a new cloud provider. Teams can then pivot strategies mid-sprint without derailing timelines. Additionally, AI Scrum Masters could automate retrospective summaries, backlog grooming, and conflict resolution, ensuring processes remain fluid and responsive.

GAITs can potentially transform specific practices, e.g., the ones that form a part of Extreme Programming (Beck 2004), and make them more feasible. One transformative approach involves zero-shot simulations, where developers use natural language prompts (e.g., "Simulate a microservice architecture for real-time inventory management") to generate functional mockups for requirement validation before implementation. These simulations reduce ambiguity in sprint planning by providing tangible prototypes for stakeholder feedback. Additionally, AI-driven tools like the Scrum Alliance's AI co-pilot exemplify how automation can reinforce self-organizing teams. Such systems facilitate daily stand-ups, track impediments, and recommend process optimizations, ensuring continuous alignment with Agile's emphasis on adaptability and collaboration. By integrating these technologies, teams can shift focus from administrative overhead to value-driven innovation, bridging the gap between Agile theory and practical execution.

Vibe coding with AI denotes an intuitive, rapid development style where programmers collaborate with AI tools to generate and refine code interactively. This approach accelerates prototyping, fosters experimentation, and shifts developers' roles from manual coding to guiding AI through natural language prompts. While it enhances creativity and productivity, it may introduce inconsistencies without proper oversight. AI tools like ChatGPT can improve code quality and efficiency but also raise concerns about trust and job security among developers.

By augmenting Extreme Programming practices and redefining roles, GenAI enables Agile teams to transcend procedural constraints, fostering a culture of innovation and customer-centricity.

3 Development Lifecycle Considerations for AI-Enabled Products

Given the promise of AI-based technologies, they will find their way into many software projects. Thus, it is important to understand AI development process models and how they integrate into traditional and Agile SDLC. In addition, tools, libraries, and frameworks for Generative AI-based solutions need to be classified and assessed to determine their impact on SDLC activities.

3.1 Artificial Intelligence Lifecycle

Specialized lifecycle models are being developed to address the unique challenges of AI projects that existing adapted methodologies fail to meet. These include limited coverage, lack of detail, and insufficient consideration of ethics, governance, risk, and workforce needs. The CDAC AI life cycle [25] is a comprehensive framework that spans from the initial conception of an AI solution to its final production. It is structured around three key phases: design, development, and deployment, further broken into 19 constituent stages. The process begins with a preliminary risk assessment at the systems level, focusing on aspects like privacy, cybersecurity, trust, explainability, robustness, usability, and social implications. The design phase involves problem formulation, literature review, data identification, ethics review, and exploration. The development phase focuses on data pre-processing, building and evaluating multiple AI models and assessing secondary metrics. The deployment phase includes technical risk classification, AI pipeline operationalization, hyper-automation integration, and continuous monitoring and evaluation.

While distinct in their focus, the CDAC AI lifecycle and the SDLC would interact in several ways. The preliminary risk assessment of the CDAC AI lifecycle could inform the SDLC's planning and requirements-gathering phases by highlighting potential AI-specific risks and ethical considerations early on. During the design phase of the SDLC, the detailed stages of the CDAC AI lifecycle concerning data identification, literature review of AI algorithms, and ethical considerations would provide specific guidance for AI components within the broader software architecture. The development phase of both lifecycles would align with the actual building and initial testing of the AI models and their integration into the software. The testing phase of the SDLC would likely incorporate the model evaluation and risk assessment stages of the CDAC AI lifecycle to ensure the AI components function correctly, ethically, and securely. Finally, the deployment and maintenance phases of the SDLC would directly benefit from the operationalization and monitoring stages of the CDAC AI lifecycle, providing a structured approach to deploying and continuously evaluating the performance and reliability of the AI elements within the software system.

3.2 Classification of AI-Powered Projects

Systems using artificial intelligence capabilities will differ in complexity, implementation strategies, and levels of autonomy. Classifying these systems can help understand development processes and process models required for their creation. It can also help select tools, technologies, and design patterns appropriate for a given project.

One way to classify these systems is by the scope of customization to the underlying model, ranging from using stock chatbots to highly customized solutions requiring partial or complete training of the LLM. One way to classify these systems is by the scope of customization to the underlying model, ranging from using stock chatbots to highly customized solutions requiring partial or complete training of the LLM. At the foundational level, generic chatbots leverage off-the-shelf large language models (LLMs) with minimal adaptation, suitable for standardized tasks like basic Q&A.

Intermediate approaches, such as retrieval-augmented generation (RAG) systems, integrate domain-specific knowledge bases to enhance accuracy without modifying the core model [26]. Advanced implementations of language models often employ fine-tuning or full custom training to meet specialized needs. Fine-tuning involves adapting a pre-trained model to a specific task or domain by continuing training on a smaller, domain-specific dataset [27]. This is useful for applications like clinical document classification or legal contract review.

In contrast, full custom training builds a language model from the ground up, using a sizeable proprietary corpus and custom architecture to meet high-security or performance requirements, for instance, in defense or pharmaceutical R&D. These approaches are essential in environments with proprietary data and strict regulatory standards, or when a smaller model is required because of limited available resources. These models are often trained using data generated using larger models, bootstrapping, or distillation [28]. **Table 6** outlines a classification of AI-enhanced systems, and **Figure 1** visualizes areas of high adoption.

Table 6. Classification of AI-Enhanced Systems.

System Type	Characteristics	Example Use Cases

Generic Chatbots	Task-specific prompts, zero-shot learning	Basic Q&A systems
Low-Code Platforms	Visual interfaces, prebuilt components	Internal tools development
RAG Systems	Domain-specific knowledge bases	Technical support bots
Fine-Tuned Models	Domain-adapted LLMs	Medical diagnosis assistants
Full-Custom AI	Proprietary model training	Defense systems

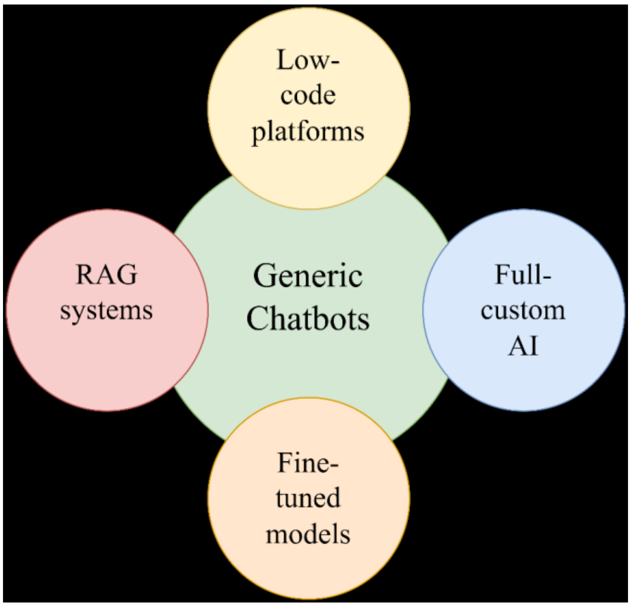


Fig. 11. Mapping of representative Generative AI tools to respective SDLC phases. Demonstrates areas of high adoption and emerging gaps.

A variety of tools were created to support the development of AI-enhanced systems. Generic chatbots can be configured with carefully engineered prompts; this approach is feasible for internally used tools and prototypes. A somewhat higher level of customization can be achieved using chatbot features like OpenAI's Custom GPT or platforms like CustomGPT.ai, which offer advanced fine-tuning and data privacy controlshttps://customgpt.ai/create-custom-gpt-openai/ (Murgia, 2023). Low-code tools such as Appsmith AI [29] enable rapid deployment of AI applications with prebuilt LLM integrations and drag-and-drop interfaces7. Google AI Studio (Google 2025) provides a streamlined environment for prototyping generative AI models, while RAG libraries like LangChain [30] facilitate context-aware responses through document indexing and retrieval. Vendor APIs such as OpenAI's GPT-4 and Anthropic's Claude 3.7 Sonnet deliver plug-and-play AI capabilities3, complemented by Python libraries like TensorFlow, PyTorch, and scikit-learn [31] for custom model development and classical ML tasks.

4 Framework for AI SDLC

In defining an updated SDLC framework for AI projects, we must address the incorporation of Generative AI tools (GAITs) into software development and specific activities aimed at AI-enabled components of a software project. The new model should be able to harness AI for development tasks and construct AI-enabled systems by optimizing each phase of the SDLC with AI-specific workflows and validation steps. The adjustments span from leveraging Large Language Models (LLMs) for precise requirements gathering to deploying AI for autonomous test-case generation.

An important aspect of the framework is attention to AI safety and alignment. This includes allocating sprints for data curation and model retraining, implementing guardrails to prevent hallucinated outputs in production systems, and monitoring model drift and performance degradation. The framework must also prioritize ethical and security reviews to mitigate biases, prevent privacy leaks, and address prompt injection vulnerabilities. **Figure 2** shows an integrated SDLC framework.

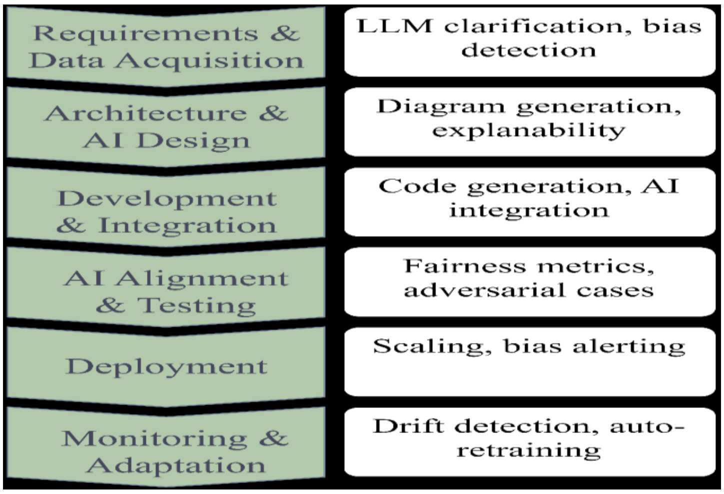

Requirements & Data Acquisition
LLM clarification, bias detection
Architecture & AI Design
Diagram generation, explanability
Development & Integration
Code generation, AI integration
AI Alignment & Testing
Fairness metrics, adversarial cases
Deployment
Scaling, bias alerting
Monitoring & Adaptation
Drift detection, auto-retraining

Fig. 2. An integrated Software Development Life Cycle (SDLC) framework enhanced with Generative AI tools (GAITs) and alignment-specific activities.

One key aspect of the AI-aware lifecycle is tools and artifacts specific to GAITs. In addition to the usual software and documentation deliverables, the process will yield AI models, prompts, and training sets used in the software development process and involved in creating AI components of the target system. These artifacts must be preserved and version-controlled, along with code, tests, and documentation. All this contributes to an evolving knowledge base that will make GAITs more capable and adapted to the project. Starting with simple prompts to clarify requirements, the team will iteratively gather documents, test data, code, and reference material to build AI-based, RAG-enhanced, and potentially fine-tuned expert team members to accompany the product in its lifecycle.

Table 2. Augmented SDLC Phases with Integrated GenAI, Alignment, and Bias Mitigation.

Stage	Activities	GenAI Tools and Project-Specific AI Assistant
Requirements & Data Acquisition	Stakeholder analysis, requirements gathering, data source identification, data acquisition planning, initial alignment and bias considerations, data profiling and bias assessment	LLM-based requirements clarification, initial knowledge base creation, AI-driven bias detection in initial data sets, requirements traceability matrix, AI-enabled zero-shot mockups, and prototype development.
Architecture & AI Design	System architecture design, UI/UX design, API design, AI component design (model selection, data flow), alignment strategy definition, explainability framework definition	AI-driven diagram generation, design artifact creation, continued knowledge base refinement, AI-assisted architecture optimization for AI components, and AI alignment and security design patterns.

Development & AI Integration	Coding, unit testing, integration testing, AI component integration (model deployment, API integration), alignment implementation, bias mitigation strategies implementation	AI-assisted coding, automated test generation, integration of AI components, enhanced knowledge base, AI-driven code review for alignment and bias issues, monitoring system logging of input and output data from the AI system.
AI Alignment & Bias Testing	Focused testing on AI alignment and bias, stress testing of alignment mechanisms, adversarial testing to expose biases, fairness metric evaluation	Automated alignment testing, AI-driven adversarial example generation, tools for fairness metric calculation and visualization.
Functional & Performance Testing	System testing, user acceptance testing, performance testing, security testing, standard functionality verification	AI-driven test case generation, automated testing, integration of test results into the knowledge base for AI-driven troubleshooting, performance optimization, and security vulnerability detection.
Deployment	Release management, infrastructure setup, deployment automation, alignment monitoring setup, bias detection in production data streams	AI is used for predictive scaling, anomaly detection, integration of deployment data into the AI assistant for proactive support, real-time bias monitoring, and alerting system.

Monitoring & Adaptation	Performance monitoring, security monitoring, bug fixing, feature updates, model retraining, adaptation to new data, continuous alignment monitoring and refinement, bias drift detection and mitigation	AI-driven log analysis, automated patch suggestions, adaptive scaling, AI assistant providing lifecycle support, automated retraining with bias correction, and real-time alignment assessment and recalibration tools.

5 Case Study: Vulnerability Scanning Tutor

To bring the AI-aware SDLC framework to life, we highlight the build phase cycle for building an intelligent tutor that teaches vulnerability scanning in an undergraduate penetration-testing course. The first step is to gather the course's learning objectives, such as "Understand Vulnerability Scanning," "Identify Scanning Techniques," and "Analyze Scan Results." Subject-matter experts share slides, lab manuals, and tool guides (for tools like Nmap and OpenVAS). An AI assistant drafts a preliminary requirements document, which needs to be reviewed, ensuring every objective ties back to specific prompts and data sources. Simple bias checks are required on source materials, ensuring that specific network setups or operating systems are not overlooked.

The next step is to sketch out a microservice-based architecture: a React front end for interactive quizzes and labs, a Python FastAPI back end to handle tutoring requests and an AI module that uses a retrieval-augmented pipeline with a fine-tuned language model. The AI helps generate UML class and sequence diagrams early on; manual adjustment is needed later to match the decided design conventions and ensure they accurately represent data flows. Logging mechanisms are required to capture prompt–response interactions so instructors can trace how the tutor arrived at each recommendation.

Coding assistants can provide boilerplate and unit-test stubs during development. Prompt templates such as "Explain how to use Nmap for port enumeration" are stored alongside the source code and are versioned in Git. Given sample network snapshots, integration tests verify that the AI module suggests valid commands and interprets results correctly. Separate sprints tackle AI-specific concerns: craft adversarial prompts to probe for hallucinations, run fairness tests to compare performance across different OS images, and simulate prompt-injection attacks to confirm our input sanitization.

Before launch, response times under load should be measured to keep them under 200 ms, and AI-generated code snippets should be scanned for potential security issues.

Deployment can be done using Docker containers on Kubernetes, with monitoring agents that feed real-time performance and fairness metrics into a dashboard. Finally, weekly retraining jobs should be set up in production to ingest new student interactions, correcting for any drift in accuracy or bias while alerting anomalies that merit a closer look.

6 Conclusion and Research Gaps

By weaving AI-powered tools into each SDLC phase, from requirements gathering and architecture design to testing, deployment, and continuous monitoring, we have outlined a practical model for creating Generative AI-enabled applications. The vulnerability-scanning tutor example shows how language models can accelerate documentation, diagramming, test creation, and even bias detection, all while remaining under human guidance. Careful sprint planning around adversarial testing and fairness metrics ensures the AI component stays reliable and trustworthy, while traditional QA processes keep performance and security on track.

However, significant questions remain. We lack hard data on how AI-assisted workflows improve developer productivity or reduce defects in real projects, especially in educational settings. There is no standard library for real-time bias or alignment checks, so teams must stitch their scripts and dashboards together. Governance is another blind spot: how do we balance intellectual property, data privacy, and regulatory compliance when retraining models on production data? Finally, the human side of human-AI collaboration deserves more study: what interface patterns help developers trust and steer AI assistants most effectively? Exploring these areas will be crucial to moving from promising prototypes to production-ready processes that blend software engineering and AI development into a cohesive discipline.

References

1. Akinepalli, K.: The Societal Nexus of Software Engineering: Balancing Innovation and Ethical Responsibility. Int. J. Res. Comput. Appl. Inf. Technol. 7(2), 634–650 (2024).
2. Rastogi, V.: Software development life cycle models-comparison, consequences. Int. J. Comput. Sci. Inf. Technol. 6(1), 168–172 (2015).
3. Turing, A.M.: Computing Machinery and Intelligence. Mind 49, 433–460 (1950).
4. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving Language Understanding by Generative Pre-Training. arXiv preprint arXiv:1801.06146 (2018).
5. Accenture: Generative AI: Understanding generative AI and how it will fundamentally transform our world. https://www.accenture.com/us-en/insights/generative-ai. Accessed 8 Apr 2025.
6. Sabherwal, R., Grover, V.: The societal impacts of generative artificial intelligence: A balanced perspective. J. Assoc. Inf. Syst. 25(1), 13–22 (2024).

7. Buscemi, A.: A comparative study of code generation using chatgpt 3.5 across 10 programming languages. arXiv preprint arXiv:2308.04477 (2023).
8. Friedman, N.: Introducing github copilot: Your ai pair programmer. https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/. Accessed 8 Apr 2025.
9. Prakash, M.: Role of Generative AI Tools (GAITs) in Software Development Life Cycle (SDLC)-Waterfall Model. Master's Thesis, Massachusetts Institute of Technology (2024).
10. Krishna, M., Gaur, B., Verma, A., Jalote, P.: Using LLMs in software requirements specifications: an empirical evaluation. In: 2024 IEEE 32nd Int. Requirements Eng. Conf. (RE), pp. 475–483. IEEE (2024).
11. Notion: Notion AI tool. https://www.notion.so/product/ai. Accessed 8 Apr 2025.
12. Google: Introducing NotebookLM. https://blog.google/technology/ai/notebooklm-google-ai/. Accessed 2 Apr 2025.
13. Jahić, J., Sami, A.: State of Practice: LLMs in Software Engineering and Software Architecture. In: 2024 IEEE 21st Int. Conf. Softw. Archit. Companion (ICSA-C), pp. 311–318. IEEE (2024).
14. DiagramGPT: DiagramGPT. https://www.eraser.io/diagramgpt. Accessed 8 Apr 2025.
15. Tian, Y., Cui, W., Deng, D., Yi, X., Yang, Y., Zhang, H., Wu, Y.: Chartgpt: Leveraging llms to generate charts from abstract natural language. IEEE Trans. Vis. Comput. Graph. (2024).
16. Joshi, V., Band, I.: Disrupting Test Development with AI Assistants. arXiv preprint arXiv:2411.02328 (2024).
17. Nandi, A.: Generative AI in Software Development; Revolutionizing Deployment and Maintenance. AIM Research (2024). https://aimresearch.co/council-posts/generative-ai-in-software-development-revolutionizing-deployment-and-maintenance. Accessed 8 Apr 2025.
18. Crespo Márquez, A., Pérez Oliver, D.: Leveraging Generative AI for Modelling and Optimization of Maintenance Policies in Industrial Systems. Information 16(3), 217 (2025). https://doi.org/10.3390/info16030217.
19. Nag, S.: Generative AI and its Impact in Software Development Lifecycle. Calsoft Inc. (2025). https://www.calsoftinc.com/blogs/generative-ai-and-the-changing-face-of-software-development-lifecycle.html. Accessed 8 Apr 2025.
20. Erolin, J.: 72% of Software Engineers Are Now Using GenAI, Boosting Productivity. BairesDev Blog (2024), https://www.bairesdev.com/blog/72-software-engineers-genai-productivity. Accessed 9 Apr 2025
21. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development. Agile Alliance (2001), https://agilemanifesto.org, last accessed 2025/04/16.
22. Beck, K.: Extreme Programming Explained: Embrace Change, 2nd edn. Addison-Wesley Professional, Boston (2004)
23. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall, Upper Saddle River (2002)
24. Naiburg, E.: AI as a Scrum Team Member. Scrum.org. (July 10, 2024) https://www.scrum.org/resources/blog/ai-scrum-team-member, accessed 16 Apr 2025.

25. De Silva, D. and Alahakoon, D.: An artificial intelligence life cycle: From conception to production. *Patterns*, *3*(6). (2025) https://doi.org/10.1016/j.patter.2022.100489
26. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) NeurIPS 2020, Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474. Curran Associates, Inc. (2020)
27. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) NeurIPS 2020, Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474. Curran Associates, Inc. (2020)
28. Taori, R., et al.: Alpaca: A Strong, Replicable Instruction-Following Model. Stanford CRFM (2023). https://github.com/tatsu-lab/stanford_alpaca
29. Appsmith: appsmithorg/appsmith. https://github.com/appsmithorg/appsmith (2023) Accessed 20 Apr 2025
30. Topsakal, O., Akinci, T.C.: Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In: International Conference on Applied Engineering and Natural Sciences, vol. 1, no. 1, pp. 1050-1056 (2023)
31. Kanagachidambaresan, G.R., Bharathi, N.: Python Packages for Learning Algorithms. In: Learning Algorithms for Internet of Things: Applying Python Tools to Improve Data Collection Use for System Performance, pp. 21-75. Apress, Berkeley, CA (2024)

24. Low Complexity Encoder Hardware for a Novel Family of Linear Block Codes

Peter Farkaš[1], Katarína Farkašová[2] and Frederik Pavelka[3]

[1,2,3] Slovak University of Technology, Ilkovičova 3, 841 04 Bratislava, Slovakia
[1]Pan-European University, Tematínska 10, 851 05 Bratislava, Slovakia
peter.farkas@stuba.sk

Abstract. A novel family of three-erasure-correcting linear block codes over $GF(q)$, where the characteristic of the field is two, has been recently proposed, along with efficient syndrome decoding procedure. The code construction relies on a specially designed parity-check matrix composed of multiple Vandermonde matrices and matrices with interleaved columns derived from them. These codes feature long codewords and high code rates, while maintaining low computational complexity for encoding and syndrome decoding, making them suitable for some erasure correction applications. One possible application is in emerging nanopore-based DNA data storage systems, where robust and efficient error correction is critical. Recently, low-complexity hardware architecture for syndrome computation, essential for syndrome decoding, has been developed for this code family. However, since the codes are defined through their parity-check matrices in nonsystematic form, an open question was whether similarly efficient hardware implementations for encoding could be achieved. This paper demonstrates that such encoding circuits are indeed possible thanks to the encoding method proposed in this paper, which is based on the structure of the codes. An example of the corresponding hardware design is also presented.

Keywords: Error Control Codes, Hardware, Encoding.

1 Introduction

The construction of practical erasure correcting codes is mainly motivated by their applications in storage systems including DNA based storages [1-8] and packet switched networks [9,10] including 6G NTN [11,12]. In 6G the erasure coding is expected to be used some massive machine type communication (mMTC) and distributed computing applications [13-19]. For example in IoT application for maritime transport systems erasure codes can also improving efficiency, security and loss of revenue [19].

Practical family of codes for erasure correction with simple and efficient realizations of encoders and decoders was discovered by McAuley in [20]. In [21] it was noted that these codes are in essence extended Reed Solomon codes and their shortened versions.

This inspired some further work and results in this direction [22]. Not long ago this efforts brought a discovery of five times extended Reed Solomon codes over finite fields $GF(2^{\xi})$ where ξ is an odd integer, were discovered in [23]. In [24] it was shown that these codes can correct up to two errors and they could be decoded using syndrome decoding. In [25] new family of three erasure correcting codes was published which is distinguished by simple implementation of decoding procedures [26].

However software based realization of the codec especially in low cost IoT or WSN devices shall have very low computational complexity. Hardware can alleviate or overcome this problem. In numerous applications the collection of date from wireless sensors allows for asymmetry in this requirement. It means that these applications can tolerate a system with low complexity encoding and more complex decoding procedures.

In this paper it is shown that the encoder for the new family of 3-erasure new Error Control Codes (3E-ECC) can be supported by using a relatively simple hardware.

The paper is organized as follows. In Section II. the new family of 3E-ECC from [25] is described for a convenience of the reader. In Section III. the structure of 3E-ECC is analyzed in more details. In Section IV. encoding procedure is proposed for these codes. In the following Section V. a encoder hardware is designed and explained using also an example of a code from the 3E-ECC family. The hardware system can simplify or speed up the encoding process. In Conclusions the contents of the paper is given in concise form.

2 Three erasures correcting block codes

It is well known [6] that any linear block code can be defined by its parity check matrix $\mathbf{H}$ with dimensions: $(n-k)\times k$. Its rows describe parity check equations valid for all codewords $\mathbf{c}$ from a corresponding code C . Therefore:

$$\mathbf{cH}^{T} = \mathbf{0} \tag{1}$$

In [5] it was proven that the following control matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \cdots & \mathbf{A} & \mathbf{A} & \mathbf{A} \\ \mathbf{B}_{q-2} & \cdots & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 \end{bmatrix}, \tag{2}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 \\ \alpha^{(q-2)} & \cdots & \alpha^{2} & \alpha^{1} & 1 \\ \alpha^{2(q-2)} & \cdots & \alpha^{4} & \alpha^{2} & 1 \end{bmatrix} \tag{3}$$

$$\mathbf{B}_0 = \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{B}_1 = \begin{bmatrix} \alpha & \cdots & \alpha & \alpha & \alpha \\ \alpha^2 & \cdots & \alpha^2 & \alpha^2 & \alpha^2 \end{bmatrix} \tag{5}$$

$$\mathbf{B}_2 = \begin{bmatrix} \alpha^2 & \cdots & \alpha^2 & \alpha^2 & \alpha^2 \\ \alpha^4 & \cdots & \alpha^4 & \alpha^4 & \alpha^4 \end{bmatrix} \tag{6}$$

$$\mathbf{B}_3 = \begin{bmatrix} \alpha^3 & \cdots & \alpha^3 & \alpha^3 & \alpha^3 \\ \alpha^6 & \cdots & \alpha^6 & \alpha^6 & \alpha^6 \end{bmatrix} \tag{7}$$

$$\vdots$$

$$\mathbf{B}_{q-2} = \begin{bmatrix} \alpha^{(q-2)} & \cdots & \alpha^{(q-2)} & \alpha^{(q-2)} & \alpha^{(q-2)} \\ \alpha^{2(q-2)} & \cdots & \alpha^{2(q-2)} & \alpha^{2(q-2)} & \alpha^{2(q-2)} \end{bmatrix} \tag{8}$$

, and α is a primitive element from $GF(q)$, defines a family of linear block codes over $GF(q)$, which could correct up to 3 erasures in each codeword.

3 On structure of the family of 3E-ECC

The goal of 3E ECC encoding is to calculate the values of 5 parity symbols corresponding to $n-5$ payload symbols in such a way that (1) will be valid. The standard approach to encoding any linear block code is based on generating matrix $\mathbf{G}$ for which

$$\mathbf{GH}^T = \mathbf{0} \tag{9}$$

Usually $\mathbf{G}$ is obtained from $\mathbf{H}$ brought into systematic form

$$\mathbf{H} = \left[\mathbf{P}^T_{(n-k)\times k} \middle| \mathbf{I}_{(n-k)\times(n-k)}\right] \tag{10}$$

as follows

$$\mathbf{G} = \left[\mathbf{I}_{k\times k} \middle| \mathbf{P}_{k\times(n-k)}\right] \tag{11}$$

The encoding - obtaining a codeword $\mathbf{c}$ with length n from a message vector $\mathbf{m}$ containing k so called information symbols can be than expressed as

$$\mathbf{c} = \mathbf{mG} \tag{12}$$

However application of such approach to the 3E-ECC codes will destroy the structure of $\mathbf{H}$, which is essential for simple hardware implementation of the encoding. To explain this in more detail let us analyze (1) and (2)-(8). It is obvious that (1) for the 3E-ECC codes defined by $\mathbf{H}$ given by (2)-(8) can be interpreted as 5 equations, which could be expressed explicitly after denoting the codeword from 3E-ECC as a vector

$$\begin{gathered}\mathbf{c}=(c_{q-2,q-2},\ldots,c_{1,q-2},c_{0,q-2}\mid\ldots\\ \ldots\mid c_{q-2,1},\ldots,c_{1,1},c_{0,1}\mid\ldots\mid c_{q-2,0},\ldots,c_{1,0},c_{0,0})\end{gathered}\tag{13}$$

1-st equation corresponding to the 1-st row from top in $\mathbf{H}$

$$\begin{gathered}(c_{0,0}+c_{0,1}+c_{0,2}+\ldots+c_{0,q-2})+\\ +(c_{1,0}+c_{1,1}+c_{1,2}+\ldots+c_{1,q-2})+\\ \vdots\\ +(c_{q-2,0}+c_{q-2,1}+c_{q-2,2}+\ldots+c_{q-2,q-2})=0\end{gathered}\tag{14}$$

2-nd equation corresponding to the 2-nd row from top in $\mathbf{H}$

$$\begin{gathered}(c_{0,0}+c_{0,1}+c_{0,2}+\ldots+c_{0,q-2})+\\ +(c_{1,0}+c_{1,1}+c_{1,2}+\ldots+c_{1,q-2})\alpha+\\ \vdots\\ +(c_{q-2,0}+c_{q-2,1}+c_{q-2,2}+\ldots+c_{q-2,q-2})\alpha^{q-2}=0\end{gathered}\tag{15}$$

3-rd equation corresponding to the 3-rd row from top in $\mathbf{H}$

$$\begin{gathered}(c_{0,0}+c_{0,1}+c_{0,2}+\ldots+c_{0,q-2})+\\ +(c_{1,0}+c_{1,1}+c_{1,2}+\ldots+c_{1,q-2})\alpha^{2}+\\ \vdots\\ +(c_{q-2,0}+c_{q-2,1}+c_{q-2,2}+\ldots+c_{q-2,q-2})\alpha^{2(q-2)}=0\end{gathered}\tag{16}$$

4-th equation corresponding to the 4-th row from top in $\mathbf{H}$

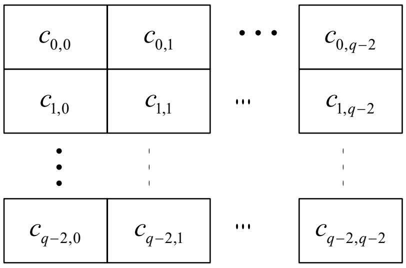
$$\begin{gathered}(c_{0,0}+c_{1,0}+c_{2,0}+\ldots+c_{q-2,0})+\\ +(c_{0,1}+c_{1,1}+c_{2,1}+\ldots+c_{q-2,1})\alpha+\\ \vdots\\ +(c_{0,q-2}+c_{1,q-2}+c_{2,q-2}+\ldots+c_{q-2,q-2})\alpha^{(q-2)}\end{gathered}\tag{17}$$

5-th equation corresponding to the 5-th row from top in $\mathbf{H}$

$$\begin{aligned}&(c_{0,0}+c_{1,0}+c_{2,0}+\ldots+c_{q-2,0})+\\&+(c_{0,1}+c_{1,1}+c_{2,1}+\ldots+c_{q-2,1})\alpha^{2}+\\&\qquad\vdots\\&+(c_{0,q-2}+c_{1,q-2}+c_{2,q-2}+\ldots+c_{q-2,q-2})\alpha^{2(q-2)}\end{aligned}\tag{18}$$

One can see that the codeword symbols in brackets in equations (11), (12) and (13) are ordered identically. After reordering by interleaving the same is true for equations (14) and (15). This interleaving is illustrated in Fig. 1

After filling the codeword in a two-dimensional memory (a block interleaver), the order of the symbols in brackets in (11), (12) and (13) corresponds to the order of symbols in columns of the interleaver in Fig. 1. On the other hand the order of the symbols in brackets in (14) and (15) corresponds to the order of symbols in rows of the interleaver in Fig. 1.



$c_{0,0}$	$c_{0,1}$	$\cdots$	$c_{0,q-2}$
$c_{1,0}$	$c_{1,1}$	$\cdots$	$c_{1,q-2}$
$\vdots$	$\vdots$		$\vdots$
$c_{q-2,0}$	$c_{q-2,1}$	$\cdots$	$c_{q-2,q-2}$

Fig. 12. Codeword stored in a 2-dimensional memory (interleaver).

Later in this paper this structure will be exploited to design a simple hardware supporting encoding of the 3E-ECC. However before in next section a method will be described, which will allow to calculate the parity check symbols without using a **G** matrix and obtain codewords in systematic form.

4 Encoding method based on the structure of the family of 3E-ECC

In case that the symbols containing information (the message symbols) are present in a codeword in explicit form it is said that the codeword was obtained by systematic encoding. Common approach is to use (12) with **G** in systematic form (11). However as was already mentioned transforming $c_{q-2,1}$ will destroy the structure, which allows us to use the interleaver described in previous section in our hardware system supporting encoding.

Another option is to use a method based on (1). It represents in compact form 5 linear equations. If the codeword $\mathbf{c}$ has to be in systematic form and the $n-k$ information symbols and their positions in $\mathbf{c}$ are known, than they can be used for obtaining a linear combination $\mathbf{r}=(r_1,r_2,r_3,r_4,r_5)$ of the corresponding rows from $\mathbf{H}^T$. It could be done simply by inserting zeros into the remaining five coordinates of $\mathbf{c}$ and use resulting vector $\mathbf{c}'$ as follows

$$\mathbf{r}=\mathbf{c}'\mathbf{H}^T \tag{19}$$

The five parity check symbols, $p_I, p_{II}, p_{III}, p_{IV}, p_V$, the columns $\mathbf{h}_I, \mathbf{h}_{II}, \mathbf{h}_{III}, \mathbf{h}_{IV}, \mathbf{h}_V$ from $c_{q-2,1}$ corresponding to their positions and $\mathbf{r}^T$ must fulfill the following equation

$$p_I\mathbf{h}_I + p_{II}\mathbf{h}_{II} + p_{III}\mathbf{h}_{III} + p_{IV}\mathbf{h}_{IV} + p_V\mathbf{h}_V = \mathbf{r}^T \tag{20}$$

It is obvious that the positions of the parity check symbols in a codeword must be chosen carefully in order that the system (20) will have solution. Fortunately the matrix $c_{q-2,1}$ allows fulfilling this requirement in numerous different selections.

To finish the encoding it is necessary to solve (20). The obtained parity check symbols $p_I, p_{II}, p_{III}, p_{IV}, p_V$ can be than inserted into appropriate positions of $\mathbf{c}'$ and so obtain the codeword $\mathbf{c}$ in systematic form.

It has to be noted that there has to be an agreement between encoder and decoder about the positions of parity check symbols. This can be done once in a protocol and then used forever.

The just presented method for encoding is based on trivial algebra and some readers may ask why it was described in such detail. Hopefully the fact that it will make the explanation of the hardware for encoding in next Section easier will which will be excuse for it.

5 Encoder hardware for 3E-ECC

In this section hardware will be presented which can be used in hybrid or hardware encoders. Usually the hardware solutions can be faster or more energy efficient than software solutions. This can be interesting in IoT or mMTC communication.

Before we start to present the encoder hardware, let us mentioning a known trick [27, 28], allowing evaluating polynomial

$$p(x)=p_m x^m + p_{m-1}x^{m-1} + \dots \qquad p_0 \tag{21}$$

, which is defined over $GF(2^\xi)$ in any point $\alpha^w \in GF(2^\xi)$. The polynomial can be expressed by the Horn scheme

$$p_m x^m + p_{m-1} x^{m-1} + \dots \qquad p_0 = \\ = (\dots \qquad \dots \tag{22}$$

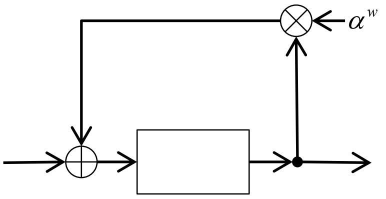


Fig. 2. Hardware evaluating polynomila in point $x = \alpha^w$. (The lines in represent a bus composed from ξ parallel wires.)

The form in (22) explains the basic idea behind the hardware in Fig. 2. It contains a storage in which the result is accumulated and one multiplier and one summation element. The lines in Fig. 2 are buses with ξ parallel lines.

Based on the analysis made in previous section the encoding could be divided into two tasks in order to make the explanation simpler. The first task is to compute $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5)$. The second task is to obtain $p_I, p_{II}, p_{III}, p_{IV}, p_V$ based on the input $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5)$.

In Fig. 3 a schematics of a hardware system is presented which can fulfill the first task, namely to calculate $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5)$.

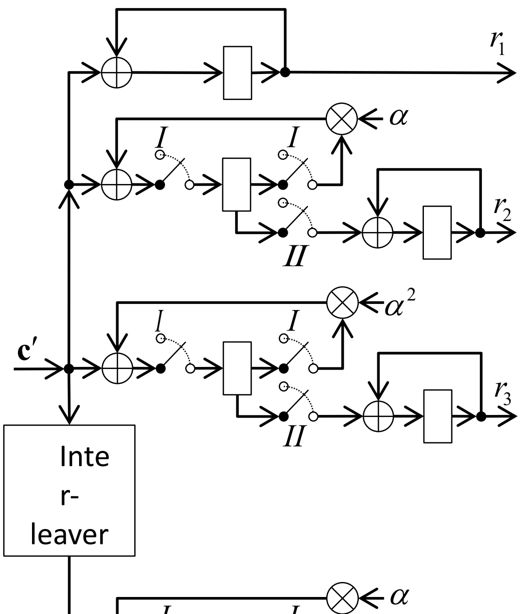


Fig. 3. Hardware evaluating the coordinates of $\mathbf{r}$. (The lines represent a bus composed from ξ parallel wires.) The switches denoted I are closed for $q-1$ clock period than they are open for 1 clock period in which the switches II are closed and this cycle is repeated $q-1$ times.

The circuits computing r_1 , r_2 and r_3 (in upper part of Fig. 3) could be better understood after observing rows of the two-dimensional memory (Interleaver) in Fig. 1 and introducing the following sums:

$$\kappa_i = \sum_{j=0}^{q-2} c'_{i,j}; \qquad i = 0, \quad \ldots \qquad\qquad ! \tag{23}$$

The first circuit from top calculates:

$$r_1 = \sum_{j=0}^{q-2} \kappa_i \tag{24}$$

The second circuit from top calculates:

$$r_2 = \sum_{j=0}^{q-2} \kappa_i \alpha^i \tag{25}$$

And the third one:

$$r_3 = \sum_{j=0}^{q-2} \kappa_i \alpha^{2i} \tag{26}$$

Similarly, the computation of r_4 and r_5 can be better understood after observing the columns in interleaver in Fig. 1 and introducing following sums

$$\rho_j = \sum_{i=0}^{q-2} v_{i,j}; \qquad j = 0, \quad \ldots \tag{27}$$

Then it is more clear that

$$r_4 = \sum_{j=0}^{q-2} \rho_j \alpha^j \tag{28}$$

$$r_5 = \sum_{j=0}^{q-2} \rho_j \alpha^{2j} \tag{29}$$

This also explains the function of the interleaver in Fig. 3.

The following alternative polynomial expression of 3E-ECC codewords can also help to understand (23) – (29).

$$\begin{aligned} & c(x) = \\ & = c_{(q-2),(q-2)} x^{(q-2)} + \ldots \qquad)x^2 + c_{1,(q-2)} x + c_{0,(q-2)} + \\ & + \ldots \\ & + c_{(q-2),1} x^{(q-2)} + \ldots \qquad + c_{1,1} x + c_{0,1} + \\ & + c_{(q-2),0} x^{(q-2)} + \ldots \qquad + c_{1,0} x + c_{0,0} \end{aligned} \tag{30}$$

This polynomial can be also expressed as follows

$$\begin{aligned} & c_{(q-2),j} x^{q-2} + \ldots \qquad + c_{1,j} x + c_{0,j} = \\ & = (((\ldots \qquad \ldots \qquad c_{0,j} \\ & ; \quad j = 0, \quad \ldots \end{aligned} \tag{31}$$

To explain the second task, namely calculation of the parity check symbols, we will use a $[49,44,4]_{GF(8)}$ code from the family proposed in [25]. In this example the codeword from this code will be denoted as a following polynomial:

$$c(x) = c_{48}x^{48} + c_{47}x^{47} + \ldots \quad c_0 \tag{32}$$

Its submatrices of $c_{q-2,1}$ are:

$$\mathbf{B}_0 = \begin{bmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \end{bmatrix} \tag{33}$$

$$\mathbf{B}_1 = \begin{bmatrix} \alpha & \cdots & \alpha & \alpha \\ \alpha^2 & \cdots & \alpha^2 & \alpha^2 \end{bmatrix} \tag{34}$$

$$\mathbf{B}_2 = \begin{bmatrix} \alpha^2 & \cdots & \alpha^2 & \alpha^2 \\ \alpha^4 & \cdots & \alpha^4 & \alpha^4 \end{bmatrix} \tag{35}$$

$$c_{0,0} \tag{36}$$

…

$$\mathbf{B}_6 = \begin{bmatrix} \alpha^6 & \cdots & \alpha^6 & \alpha^6 \\ \alpha^5 & \cdots & \alpha^5 & \alpha^5 \end{bmatrix} \tag{37}$$

For evaluating the values of the parity check symbols it is necessary to select their positions so that the system of linear equations given in (20) will be solvable. In our example the following symbols were selected as parity symbols: $c_{42},\ c_{35},\ c_2,\ c_1,\ c_0$.

$$\begin{array}{c} \begin{bmatrix} 1 & \cdots & & \cdots & & \cdots & \\ \alpha^6 & \cdots & & \cdots & & \cdots & \\ \alpha^5 & \cdots & & \cdots & & \cdots & \\ \alpha^6 & \cdots & & \cdots & & \cdots & \\ \alpha^5 & \cdots & & \cdots & & \cdots & \end{bmatrix} \\ \mathbf{h}_{48} \quad \cdots \qquad\qquad \cdots \qquad\qquad \cdots \end{array}$$

Fig. 4. Fragment of $[49,44,4]_{GF(8)}$ code parity check matrix $\mathbf{H}$ from which their columns $\mathbf{h}_{42},\ \mathbf{h}_{35},\ \mathbf{h}_2,\ \mathbf{h}_1,\ \mathbf{h}_0$ corresponding to positions of parity check symbols $c_{42},\ c_{35},\ c_2,\ c_1,\ c_0$ can be determined. (In this case the codeword is denoted as

This allows us to extract from $\mathbf{H}$ the corresponding columns $\mathbf{h}_{42}$, $\mathbf{h}_{35}$, $\mathbf{h}_{2}$, $\mathbf{h}_{1}$, $\mathbf{h}_{0}$ (please see Fig.4) and form and write (20) for this particular choice of parity check symbols in our example

$$c_{42} + c_{35} + c_2 + c_1 + c_0 = r_1 \tag{38}$$

$$c_{42} + c_{35} + \alpha^2 c_2 + \alpha c_1 + c_0 = r_2 \tag{39}$$

$$c_{42} + c_{35} + \alpha^4 c_2 + \alpha^2 c_1 + c_0 = r_3 \tag{40}$$

$$\alpha^6 c_{42} + \alpha^5 c_{35} + c_2 + c_1 + c_0 = r_4 \tag{41}$$

$$\alpha^5 c_{42} + \alpha^6 c_{35} + c_2 + c_1 + c_0 = r_5 \tag{42}$$

From (38) – (42) it follows that

$$c_{42} = \alpha r_1 + r_4 + \alpha^3 r_5 \tag{43}$$

$$c_{35} = r_1 + \alpha^3 r_4 + \alpha r_5 \tag{44}$$

$$c_2 = \alpha^4 r_1 + \alpha^6 r_2 + \alpha^3 r_3 \tag{45}$$

$$c_1 = \alpha r_1 + \alpha^5 r_2 + \alpha^6 r_3 \tag{46}$$

$$c_0 = \alpha r_1 + \alpha r_2 + \alpha^4 r_3 + \alpha r_4 + r_5 \tag{47}$$

This allows designing a simple hardware for calculation of each parity check symbol. For example in Fig 5 hardware for calculating c_{42} is illustrated. Similar hardware solution as depicted in Fig. 5 can be made for the other parity check symbols based on (44) – (47). It is obvious that the second task, namely the computation of parity check symbols during encoding has to be adapted to the chosen positions of them, but in principle it requires at most 5 circuits with at most 5 elements of the type depicted in Fig. 2.

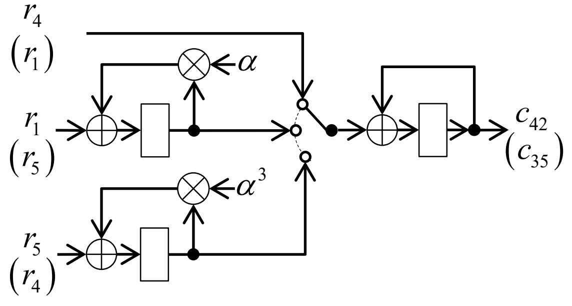


Fig. 5. Hardware for computing c_{42} and (c_{35}) respectively

6 Conclusions

In this paper, we proposed a hardware design that enables systematic encoding of the 3-erasure correcting codes introduced in [25]. These codes are originally defined using non-systematic parity-check matrices constructed from multiple Vandermonde matrices and interleaved columns of Vandermonde matrices. To facilitate systematic encoding, we developed a corresponding mathematical method. This analytical formulation also reveals certain protocol requirements, such as the need to agree on the positions of parity symbols within the encoded codewords.

The proposed encoder hardware is based on this method, incorporating both novel elements and previously established circuit techniques for finite field computations. For the reader's convenience, we also provide a detailed example of the hardware implementation for a selected code from the family.

These codes are suitable for applications requiring erasure correction, particularly those involving long codewords, high code rates, and low-complexity encoding and decoding. Potential use cases include deletion correction in nanopore-based DNA storage systems and massive machine-type communication (mMTC) in 6G networks.

Future research could explore whether similar error-correcting codes that support soft decoding strategies can be developed.

7 Acknowledgement

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project 09I05-03-V02-00051.

References

1. Cheng, K., et al.: Toward Load-Balanced Redundancy Transitioning for Erasure-Coded Storage. IEEE Transactions on Parallel and Distributed Systems, (Early Access).
2. Moav, B., Gabrys, R., Yaakobi, E.: Tail-Erasure-Correcting Codes. IEEE Transactions on Information Theory 70(12), 8595-8613 (2024).
3. Cai, H., et al.: Repairing Schemes for Tamo-Barg Codes. IEEE Transactions on Information Theory, 71(1) 227-243 (2025).
4. Zhou, Z., et al.: High-Performance Error and Erasure Decoding With Low Complexities Using SPC-RS Concatenated Codes. IEEE Transactions on Very Large Scale Integration Systems, 33(1), 293-297 (2025).

5. Wang, W., et al.: Fast Acceleration Strategies for XOR-Based Erasure Codes. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 44(1), 331-344 (2025).
6. Ren, J. Li, J., T. Li, T.: Optimal Codes for Distributed Storage. In: 2023 International Conference on Computing, Networking and Communications, pp. 42-46. IEEE, Honolulu, USA, (2023).
7. Liu, X.: Repairing Triple Data Erasure with Extending Row Diagonal Parity. In: 2022 International Symposium on Advances in Informatics, Electronics and Education, pp. 45-50. IEEE, Frankfurt, Germany, (2022)
8. Yuning, Z., Shu, F.: Research on Single Disk Error Recovery Technology Based on Erasure Code. In: 4th International Conference on Civil Aviation Safety and Information Technology, pp. 1409-1413. IEEE, Dali, China, (2022).
9. Datta, A., Oggier, F.: Concurrency Control and Consistency Over Erasure Coded Data. IEEE Access 10, pp. 118617-118638 (2022).
10. Ono, M., et al.: A Data Transfer Method Combining Erasure-coding with Cumulative Acknowledgment for Lossy Optical Packet Switching Networks. In 20th Consumer Communications & Networking Conference, pp. 670-671. IEEE, Las Vegas, USA, (2023).
11. Li, Y., et al.: Low-Complexity Streaming Forward Erasure Correction for Non-Terrestrial Networks. IEEE Transactions on Communications, 71(12), 6870-6883 (2023).
12. Alessi, N., et al.: Packet Layer Erasure Coding in Interplanetary Links: The LTP Erasure Coding Link Service Adapter. IEEE Transactions on Aerospace and Electronic Systems, 56(1), 403-414 (2020).
13. Lu, X. et al.: Cescpra: A Cloud-Edge-Sensor Collaborative Proactive Reliability Assurance Technology. IEEE Sensors Journal, 25(4), 7508-7518 (2025).
14. Sumi, T., et al.: Firmware Distribution with Erasure Code for IoT applications on IEEE 802.15.4g Mesh Network. In: 29th Asia Pacific Conference on Communications, pp. 519-522. IEEE, BALI, Indonesia, (2024).
15. Dussoye, S. Issack Z., A. Chiniah, A.: Erasure Code and Edge Computing for Providing an Optimal Platform for Storage of IoT Data, In: 2019 Global Conference on Internet of Things, pp. 1-4. IEEE, Dubai, United Arab Emirates, (2019).
16. Köse, A., et al.: Impact of Block Coding on Age of Information in Centralized IoT: Insights From BEC and BSC. IEEE Communications Letters, 28(7), 1494-1498 (2024).
17. Han, D.-J., Sohn J. -Y., Moon, J.: Coded Wireless Distributed Computing With Packet Losses and Retransmissions. IEEE Transactions on Wireless Communications, 20(12), 8204-8217 (2021).
18. Kuldeep, G., Zhang, Q.: A Novel Efficient Secure and Error-Robust Scheme for Internet of Things Using Compressive Sensing. IEEE Access, 9, 40903-40914 (2021).
19. Liu, D., et al.: Flexible Data Integrity Checking With Original Data Recovery in IoT-Enabled Maritime Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, 24(2), 2618-2629 (2023).
20. McAuley, A. J.: Weighted sum codes for error detection and their comparison with existing codes. IEEE/ACM Transactions on Networking, 2(1), 16-22, (1994).
21. Farkas, P.: Comments on "Weighted sum codes for error detection and their comparison with existing codes. IEEE/ACM Transactions on Networking, 3(2), 222-223 (1995).

22. Farkas, P. Baylis, J.: Modified Generalized Weighted Sum Codes for Error Control. In: Farrell, P., Darnell, M., Honary, B. (eds.) Coding, Communications and. Broadcasting, vol. 4, pp. 63–72. Research studies Press LTD Baldock, Hertfordshire, England, (2000).
23. Rakús, M., et al.: Five Times Extended Reed-Solomon Codes Applicable in Memory Storage Systems. IEEE Letters of the Computer Society, 2(2), 9-11 (2019).
24. Farkaš, P., M. Rakús, M.: Decoding five times extended Reed Solomon codes using syndromes. Computing and Informatics, 36(6), 1311-1335 (2021).
25. Farkaš, P., et al.: New family of linear 3-erasures correcting block codes with possible applications in storage systems. Computing and Informatics, 44(2), 429-444 (2025).
26. Farkaš, P., et al.: Low complexity decoder hardware for a novel family of linear lock codes. Accepted to: 60-th International Scientific Conference on Information, Communication and Energy Systems and Technologies, Ohrid, N. Macedonia (2025).
27. Juane, L., et al.: Fundamentals of classical and moder Error-Correcting Codes. Cambringe University Press (2021).
28. Clark, G. C., Cain, J. B.: Error-correcting coding for digital communications. Plenum Press, New York (1981).

Section 5: Emerging Technologies and Application.

25. ECG-Based Heart Condition Classification: A Systematic Algorithmic Approach

Gledis Basha[1], Elma [2], Lorena Balliu[3] and Anita Xhemali[4]

[1234]Polytechnic University of Tirana, Nënë Tereza Square, Nr. 1, Tirana, Albania
gbasha@fti.edu.al, lballiu@ fti.edu.al, axhemali@ fti.edu.al

Abstract. Electrocardiogram (ECG) analysis serves as an essential diagnostic method for identifying and evaluating cardiac disorders. This study introduces a methodical, algorithm-driven approach to categorize cardiac abnormalities using ECG data. The framework employs a structured workflow comprising four stages: data collection, signal preprocessing, extraction of clinically relevant features, and classification through a hybrid system combining rule-based decision systems and Machine Learning algorithms. This approach enhances the accuracy and efficiency of ECG interpretation, enabling early detection of arrhythmias and other cardiac conditions. It is particularly beneficial for elderly individuals living alone, who may be at higher risk for undiagnosed heart problems due to limited access to regular medical checkups. Integrating this system into wearable devices or telemedicine platforms can provide real-time monitoring, early alerts, and remote medical intervention, reducing the likelihood of severe complications. By offering a proactive, AI-assisted healthcare solution, this approach improves quality of life and safety for elderly patients while reducing the burden on healthcare systems.

Keywords: ECG, Heart Condition, Classification, Systematic Algorithmic Approach, Segmentation.

1 Introduction

1.1 The importance of ECG Analysis

Electrocardiogram (ECG), Fig 1., is a vital diagnostic tool for assessing heart health, offering real-time data on the heart's electrical activity [1], [2]. It is essential for detecting cardiac abnormalities like arrhythmias, myocardial ischemia, conduction disorders, and heart failure, which can result in serious complications if not identified early. For older adults, ECG analysis is particularly important due to the increased likelihood of age-related cardiovascular conditions, including Atrial Fibrillation (AF), silent ischemia and heart block, [4]. Continuous ECG monitoring enables early detection of these issues, helping to prevent severe outcomes such as: stroke, sudden cardiac arrest, and other critical events.

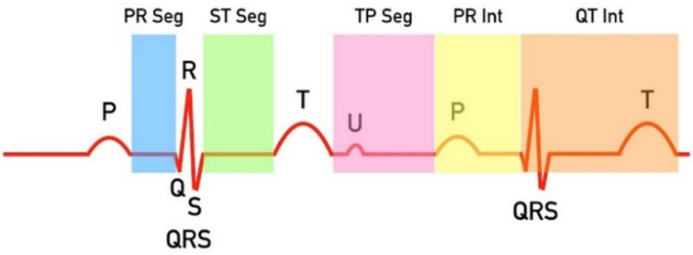


Fig. 1. Segmentation of the different parts of the ECG signal.

This not only enhances life expectancy but also improves overall quality of life for elderly individuals. Critical physiological markers—including Heart Rate Variability (HRV), QRS complex morphology, ST-segment deviations, and rhythm irregularities—are evaluated to diagnose pathologies such as arrhythmias, myocardial ischemia, and conduction system defects, Fig 1, show a normal segmentation of the EGC signal [4]. The methodology advances diagnostic precision while establishing a scalable infrastructure for continuous cardiac surveillance. By incorporating rule-based systems, the system augments its prognostic potential, enabling effective deployment in both clinical environments and remote telemedicine platforms, [5]. This integration supports timely interventions and expands accessibility in cardiovascular care, [6].

Notably, this framework holds significant promise for enhancing care for elderly individuals living alone. By enabling continuous, non-invasive cardiac monitoring through wearable ECG devices, the system can detect early signs of life-threatening conditions (e.g., atrial fibrillation, silent ischemia) and trigger automated alerts to caregivers or emergency services. This capability reduces risks associated with delayed medical intervention, such as stroke or cardiac arrest, while promoting independent living.

1.2 Current Challenges in Cardiac Diagnosis

Delayed Symptom Recognition. Many cardiac conditions, such as silent ischemia and atrial fibrillation, often lack noticeable symptoms, making them difficult to detect without medical intervention [7]. This delay can result in the progression of potentially life-threatening conditions, emphasizing the need for proactive monitoring and early detection systems.

Lack of Continuous Monitoring. Traditional ECG methods, such as hospital-based tests or Holter monitors, only capture short-term or intermittent data, often missing critical abnormalities that occur during specific times, such as at night [7]. Continuous monitoring through wearable ECG devices offers a solution by providing real-time, round-the-clock data.

Limited Access to Immediate Medical Help. For elderly individuals living alone, sudden cardiac events like heart attacks, strokes, or severe arrhythmia can be particularly dangerous[7]. Integrating automated emergency alert systems with ECG monitoring could significantly improve response times, ensuring timely medical assistance and potentially saving lives.

Age-Related Physiological Variations. ECG signals in older adults often differ from those in younger individuals due to factors such as weakened cardiac muscles, slower conduction pathways, and the effects of medications [7]. Age declaration in algorithms that adapt to these physiological changes are essential for more diagnostic accuracy.

Telemedicine & AI Integration Barriers. While AI-driven ECG analysis and telemedicine solutions hold great promise for remote cardiac care, several challenges hinder their widespread adoption among elderly individuals [7]. Addressing these challenges is crucial to ensuring that elderly individuals can fully benefit from advancements in cardiac monitoring and telemedicine.

2 Methodology

2.1 Flow of the Algorithmic Approach

This study introduces an automated ECG-based heart condition classification system, specifically designed for continuous monitoring and early detection of cardiac abnormalities, with a focus on elderly individuals living alone. The system follows a structured, five-stage methodology to ensure accurate and reliable analysis, Fig. 2.

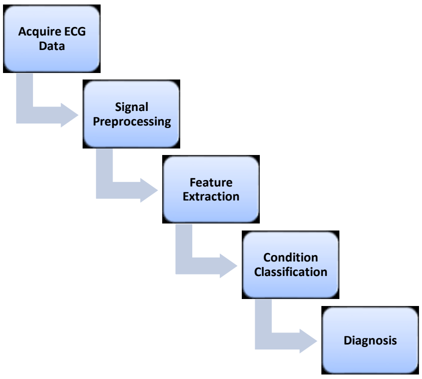


Fig. 2. Flow of the Algorithm from data acquired to the diagnosis.

Data acquisition. The collected data is securely transmitted to a cloud or server for further processing, ensuring accessibility and scalability. In our case the data are collected throw a sensor AD8232 connected to an ESP32 [1], [7].

Signal Preprocessing. To ensure high-quality analysis, the raw ECG signals undergo preprocessing. Artifacts caused by muscle movements, power line interference, and baseline drift are removed using advanced filtering techniques such as wavelet demising or Butterworth filters. R-Peak Detection, the QRS complex, a critical component of the ECG waveform, is identified using algorithms like Pan-Tompkins or machine learning-based peak detection methods. Segmentation of the ECG signal is divided into meaningful segments (e.g., P-wave, QRS complex, T-wave) to facilitate detailed analysis of each cardiac cycle. The data acquisition is made with the AD8232 is an integrated signal conditioning amplifier designed specifically for ECG and other biopotential measurements. This sensor had its own built-in noise reduction filters as it includes a high-pass filter with a cutoff frequency around 0.5 Hz. This helps remove baseline wander (low-frequency noise), which is common in ECG signals due to patient movement or slow fluctuations in the signal. It also has an internal low-pass filter with a cutoff frequency around 40 Hz. This helps filter out high-frequency noise (like electrical interference) that is above the typical range of an ECG signal, which is generally between 0.05 Hz and 100 Hz.

The AD8232 includes a right-leg drive circuit that helps improve the common-mode rejection ratio (CMRR), which reduces interference from external sources, such as power line noise or other electrical equipment. This is achieved by driving a reference signal to the patient’s body, which helps minimize common-mode noise.

Also includes lead-off detection circuitry, which can indicate when the ECG electrodes are not properly attached to the patient, providing an additional layer of signal quality control.

Feature Extraction. The segmentation shows parts of Time-Domain Features, metrics such as heart rate variability (HRV), PR interval, QRS duration, and QT interval are calculated to assess temporal changes in cardiac activity. Frequency-Domain Features, power spectral density (PSD) analysis is performed to evaluate autonomic nervous system function and detect irregularities. Morphological Features, specific waveform characteristics, such as ST-segment deviations, T-wave abnormalities, and irregular rhythm patterns, are analyzed to identify potential cardiac conditions.

Condition Classification. A hybrid classification approach is employed to enhance diagnostic accuracy. Rule-Based System: predefined thresholds and rules are applied to distinguish between normal and abnormal ECG patterns. The data are analyzed in real time and not after. The classification is made seconds after the ECG trace is detected from the sensor. One classification is made from the frequency of the heart rate itself by determining the different sectors we have from the segmentation of PR interval, QRS duration, and QT interval. The algorithm is based only on rules we have predefined, and the data are being compared. The only input is the age of the patient as the thresholds values are different for elder and early patients.

In the future we want to use the Machine Learning Model. Advanced classifiers, such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), or Long Short-Term Memory (LSTM) networks, are trained to improve detection accuracy and adapt to complex ECG patterns. AI Confidence Scoring: by combining multiple diagnostic signals, the system reduces false alarms and ensures reliable results, [5]. One limitation is the lower hardware ability of the ESP 32 to process complex algorithm such AI. From what we have seen so far, the most suitable is the CNNs but this is subject of future work.

Diagnosis. When abnormalities such as atrial fibrillation, ischemia, or other critical conditions are detected. Immediate alerts are sent to caregivers, family members, or medical professionals, enabling prompt intervention.

This comprehensive system not only enhances early detection of cardiac abnormalities but also provides a robust framework for real-time monitoring and timely intervention, particularly benefiting elderly individuals living alone.

Table 7. Table of simple output of the analysis

Heart Rate	ST-Segment	QRS Duration	Rhythm	Condition
55 bpm	Normal	Normal	Regular	**Sinus Bradycardia**
120 bpm	Normal	Normal	Regular	**Sinus Tachycardia**
80 bpm	ST-elevation	Normal	Regular	**STEMI (Heart Attack)**
90 bpm	Normal	Prolonged	Irregular	**Atrial Fibrillation**

Accuracy. After having the data, we were able to identify events in which the diagnoses were made. These events were part of the real-time monitoring we had for 12 hours of the same patient.

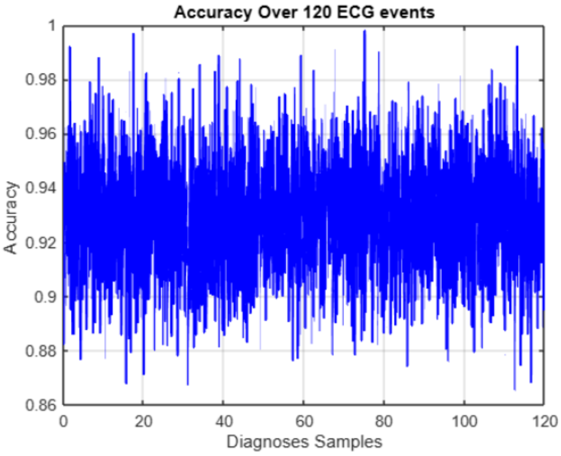


Fig. 3. Accuracy of the classification.

As we can see in Fig.3. in this case we have 120 different events, not in the normal values of the pre-defined as shown in the Table 1. Most of the events are not life threatening but it is very good to understand them and have them classified as abnormalities in a certain class. In Fig.3 we have the ratio of the classification values between our algorithm and a commercial algorithm CardioDay V2.6 [8]. From the graph the average report of the ratio generated from the classification values is 0.93 or 93% accuracy shown in Fig.3. Both algorithms share the same methodology and range of values, and we see this fit as a model for comparison and a way to test our approach. The data that we have collected are submitted in the Research option that CardioDay V2.6 has. This option allows us to run the software in developer mode, making us able to export or import patient vital sign database. All measurements are analyzed and compared to detect the different events, in our case 120 events. Although CardioDay, as a commercial algorithm, covers a much wider range of medical events compared to our simpler model, the comparable results we achieve for the targeted events motivate us to continue enhancing our algorithm by introducing additional thresholds. The results are very promising to keep evolving and addressing the approach off such simple solution to machine learning more effectively.

3 Conclusions

This study presents a structured, algorithm-driven approach for ECG-based heart condition classification, integrating signal processing, feature extraction, and AI-driven classification to enhance diagnostic accuracy. By leveraging a hybrid system of rule-based decision-making and Machine Learning models. The proposed framework ensures real-time, automated detection of cardiac abnormalities, particularly benefiting elderly individuals living alone. The implementation of this system in wearable ECG devices and telehealth platforms enables continuous, non-invasive cardiac monitoring, allowing for early detection of life-threatening conditions such as atrial fibrillation and silent ischemia. Also, the data that we gather from the system output helps us to determinate some diagnoses and help with automated and more effective analysis. All the outputs in the future help for an automated relations to the diagnoses and the patient situation. This system compared to commercial software used in hospitals is 93% accurate. With these results make us more eager to keep the future work and evolve the algorithm to more complex tasks and more diagnoses but always to be compatible for lower hardware platforms to make the approach cheaper and more accessible.

References

1. Pratik Kanani, Mamta C. Padole: Recognizing Real Time EKG Anomalies Using Arduino, AD8232 and Java, University of Baroda, Vadodara (2018)
2. Aleksandra Rashkovska, Matjaž Depolli, Ivan Tomašic,Viktor Avbelj, Roman Trobec: Medical-Grade EKG Sensor for Long-Term Monitoring, Ljubljana, Sloven (2020)
3. Ayaskanta Mishra, Biswarup Chakraborty:AD8232 based Smart Healthcare System using Internet of Things (IoT), International Journal of Engineering Research & Technology (IJERT)(2018)
4. Fakhara Sabir, Mahmood Barani, Mahwash Mukhtar: Nanodiagnosis and Nanotreatment of cardiovascular diseases. Nanostructured Devices for Biochemical Sensing (2021).
5. Lorena Balliu, Blerina Zanaj, Gledis Basha, ElmaZanaj, Elinda Kajo Meçe "Enhancing Heart Disease Prediction Accuracy by Comparing Classification Models Employing Varied Feature Selection Techniques", Serbian Journal of Electrical Engineering, Vol 21 No 3 (2024).
6. Elma Zanaj., Gledis Basha , Aleksander Biberaj and Lorena Balliu, 10th International Conference on Modern Power Systems – MPS 2023, "Introducing an Intelligent Wireless Monitoring System in Telemedicine using IoT Technology", Cluj Napoca, Rumani. (2023)
7. Gledis Basha, Lorena Balliu. and Elma Zanaj, 2nd International Conference on Information Technologies and Educational Engineering (ICITEE 2023), "Wearable sensors for cardiac disease monitoring using IoT", Tiranë, Shqipëri. (2023)
8. https://www.gehealthcare.co.uk/products/diagnostic-cardiology/ambulatory-ecg/cardioday-holter-ecg-software

26. How to teach and evaluate students in computer science in the era of LLMs?

Raphaël Couturier [1], Joseph Azar [2], Stéphane Domas [3]

Shkelqim Fortuzi[4] , Amarildo Rista[5] and Frida Gjermeni[6]

[1,2,3] Université Marie et Louis Pasteur, CNRS, Institut FEMTO-ST, F-90000, Belfort, France
[4,5] Aleksandër Moisiu University of Durrës, Albania

raphael.couturier@univ-fcomte.fr

Abstract. With the progress of LLMs (Large Language Models), it has become easier to program in many languages. Therefore, it is clear that students in computer science or IT (Information Technology) can use LLMs for various programming applications. Using LLMs can provide significant assistance to teachers in proposing new exercises or projects. Similarly, students can program more efficiently with the help of LLMs. What are the benefits and drawbacks of understanding new concepts in programming through LLMs? This question is interesting and perhaps challenging to answer because while there are advantages, there are also significant drawbacks. In this paper, we will explore these different aspects that can positively and negatively impact the way programming is taught and the way students learn to program.

Keywords: Large Language Model, Computer science teaching, Information technology evaluation.

1 Introduction

Large Language Models (LLMs) have challenged conventional teaching strategies and evaluation procedures, radically altering the field of computer science education [1]. Teachers are faced with previously unheard-of challenges regarding how to accurately assess student competency and guarantee authentic learning outcomes as these complex AI tools exhibit ever-more-advanced capabilities in code generation, debugging, and explanation [2]. Current research on the effects of LLMs on computer science education is summarized in this paper, with a focus on learning objectives and assessment techniques.

Computer science is like all other disciplines taught in universities. There are lectures, directed works, and practical labs in which students are expected to learn the concepts and practice of many topics in computer science (CS) or information technology (IT). For convenience, in this paper, IT students and CS will be used interchangeably. It would be impossible to make an exhaustive list of all the possible lessons taught in universities in this domain. For many years, even as new concepts emerge regularly in computer science, teachers and students have always used the same methods of teaching and learning. For example, in [18] authors have students how LLMs can help students to debug some programs.

With the advances in artificial intelligence, many things have changed, especially in programming, since many AI systems are able to understand questions in natural languages and produce code and explanations that can be very accurate. Of course, there are many positive aspects to this because it delegates to the AI the boring and repetitive tasks that many programmers often do regularly. Similarly, it can help teachers with quick corrections for some exercises, and they can even ask the AI to produce exercises or problems on given issues in their lectures. For the students, there are many positive aspects but also many negative ones. The essence of this paper is to discuss some implications of the use of AI in the education of IT students.

In the following, the plan of this paper is described. Section 2 presents the evaluation of computer science students before LLMs. Section 3 describes some capabilities of AI. Section 4 focuses on the evaluation of IT students in the era of LLMs. A discussion is provided in the next section. Finally, Section 6 concludes this paper and provides some perspectives.

2 Evaluation of computer science students before LLM

Evaluation of students in computer science for the era of LLMs was done much like in most other courses: with projects, written exams, and oral presentations.

Projects were designed to allow students to work alone or with a group on one or many tasks, covering different aspects of programming. Of course, by the end of 2020, many students had the opportunity to consult various interesting websites like Stack Overflow to find solutions to common problems or directly ask questions. In general, they needed to understand and adapt the answers to their specific problems. From the teacher's point of view, students learned a lot by doing this because they were able to apply knowledge to their own context. This step is interesting because most of the time, someone remembers the problem they had and how they were able to solve it.

Written exams were important for evaluating the students' ability to solve more or less complex problems in computer science. Additionally, these exams could assess the students' reasoning skills. Even if writing algorithms on paper is more difficult and may contain syntactic errors, teachers are accustomed to understanding the principle of an algorithm and can significantly evaluate the students' capability to devise solutions for various problems.

Finally, oral evaluation is very important to enable students to present their work. IT workers are often asked to present their technical work to non-IT specialists. This requires the ability to use abstraction to present functionalities while not providing too many technical details. However, it also requires the ability to answer some difficult questions.

3 Capacity of LLMs and impact on student learning

3.1 Large Language Models

Large Language Models (LLMs) are based on the Transformer architecture [19]. Many models are available either from big companies or large consortiums. For some of these models, open-source versions are also provided to let developers create new applications and allow users to interact with the LLM. Concerning programming, LLMs are very good at this task. It is not surprising since many algorithms are similar and a lot of code is available on various platforms like GitHub or Stack Overflow.

So, concerning programming, LLMs are able to generate code corresponding to a user's specifications, improve parts of existing code, write comments for existing code, and add new functionalities to an existing project.

From the point of view of students, LLMs could be very instructive for many reasons. First, LLMs can help them to acquire a lot of new knowledges on so many domains, as soon as they are curious enough.

3.2 LLMs impact on student performance

According to recent empirical research, LLM integration has conflicting effects on students' performance in computer science courses. Several studies show that their performance significantly improves when students use LLM support. While Lyu et al. [4] reported notable gains in final scores for students using an LLM-powered tool called CodeTutor, Akçapınar, and Sidan [3] noted notable increases in exam scores among students allowed to use LLMs. After using HypoCompass, an LLM-based training system, students' debugging performance improved by 12%, and they also reported feeling more confident, according to Ma et al. [5].

Nonetheless, several studies point out possible disadvantages of using LLM. Jošt et al. [6] found that final grades in a second-year programming course were negatively correlated with extensive use of LLM for code generation and debugging. Similarly, in an intermediate-level undergraduate course, Padiyath et al. [7] discovered a negative relationship between early LLM usage and midterm performance as well as self-efficacy. These results imply that although LLMs might improve task performance right away, they might jeopardize deeper learning processes and the capacity for autonomous problem-solving.

Similar complexity exists in the effects on self-perception and student engagement. Kelly et al. [9] reported improved student engagement with an LLM-powered gamified quiz, and Kumar et al. [8] noted increased self-confidence among students who participated in LLM-guided reflection. However, Lyu et al. [4] observed that students eventually started to favor human teaching assistants for specific tasks after voicing concerns about LLMs' inability to develop critical thinking abilities.

All these contradictions are increasingly noticeable in our computer science department, particularly among our current first and second-year students, who already started using LLMs in secondary school. These students almost always use LLMs, through chatbots or coding assistants, for every problem they need to implement, sometimes even before reading the problem statement. The core of the learning issue is that if the generated solution seems to work—that is, it compiles, executes, and produces output—it is generally considered sufficient by the students. No further analysis is conducted to evaluate the correctness of the solution, let alone to understand it. This behavior is based on complete confidence in the ability of LLMs to produce the expected results, which fosters the lack of critical thinking mentioned in the studies above.

Unfortunately, this confidence tends to be increasingly justified, which reinforces the reflex to use LLMs, even for cheating on exams or in coding contests. This tendency is perfectly illustrated by Figure 1 and Table 1 below.

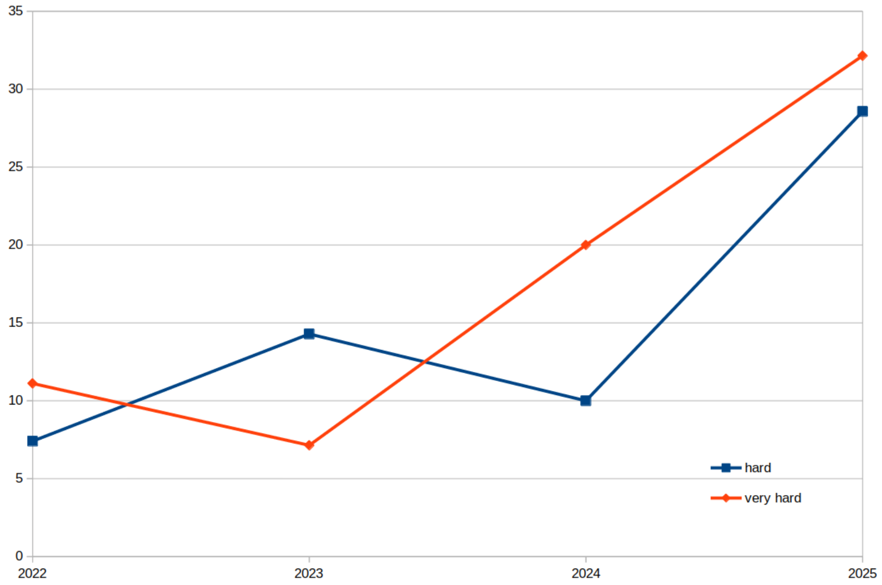


Fig. 1. % of student teams solving hard and very hard riddles in a coding contest

Figure 1 shows statistics for a coding contest that we have organized in our department in Belfort, France for several years. This contest presents algorithmic challenges with varying difficulties, and groups of students submit their solutions to a server (DOMJudge). Upon submission, the server checks the correctness of the solution using secret inputs and compares the results to an associated oracle. The blue line represents the percentage of students who succeed in solving hard challenges, and the red line represents very hard challenges, both ranging from the year 2022 to 2025. Solving these challenges mostly requires the use of complex data structures (trees, graphs) and recursion. Since the contest is open to all our students, including first-year students, it explains the average success rate of around 10% (the red line showing 20% in 2024 is an artifact due to unforeseen help from a teacher), which generally corresponds to one or two groups considering the total number of participating groups (between 10 and 14). Nevertheless, the percentage drastically increases for the current year. The reason is very simple: compared to prior years, LLMs are now able to correctly solve these problems by taking the entire subject as input. This is clearly evidenced by the codes submitted by students, which are 95% similar and clearly generated through an LLM. Moreover, this year, students passed a given challenge with only one or two submissions, whereas they often needed more than 5 or even 10 in the past. Finally, as usual, a few other groups tried to submit a solution to these exercises without succeeding. Nevertheless, this year, the ratio of succeeding groups to failing groups has largely exceeded 1. This is an example of the fact that students are increasingly tempted to use LLMs to solve problems that are beyond their skills, despite being warned that their solution will be rejected if generated through LLMs.

Table 1. Average/standard deviation (out of 20) on a MCQ test on vuejs, using moodle

	2023	2024	2025
Average	13.66	15.09	16.1
Std dev.	2.26	1.85	1.9

Table 1 presents statistics from an exam conducted on the Moodle platform for second-year students, spanning from 2023 to 2025. It is a test on Vue.js, primarily using multiple-choice questions. Each student receives a personalized set of questions randomly selected from a wide pool that has remained unchanged over the years. It is important to note that this exam is taken without active observation of the students. The goal is to evaluate the students' understanding of the basic mechanisms and syntax of Vue.js. Thus, an LLM is perfectly capable of providing very accurate answers to the exam questions. Over the past two years, Table 1 clearly shows a drastic increase in the average results, along with a decrease in the standard deviation. This can only be reasonably explained by an increasing use of LLMs to obtain answers and, thus, to cheat. This could surely be confirmed by active observation during future sessions.

At first glance, these two examples seem to confirm that LLMs particularly encourage students to take the 'easy way out' and diminish their ability to solve complex problems independently or simply to acquire knowledge. Nevertheless, after some discussions about the coding contest, it appears that some participants genuinely learned from the solutions they obtained via LLMs, even if they did not fully understand how they work. In fact, these examples illustrate two contradictory aspects of LLMs. Despite the fact that they can be used blindly as a tool to cheat efficiently (in terms of result accuracy), they also provide a way to discover and test new algorithms or knowledge, provided that students are aware of, or prepared by teachers for, such an approach.

3.3 Challenges to traditional assessment methods

Traditional methods of assessment in computer science education face significant challenges as a result of the integration of LLMs. Although they had trouble with calculation and image-based questions, Quille et al. [10] showed that LLMs performed noticeably better than students when responding to exam questions. When LLM assistance is available, this finding calls into question the validity of traditional assessment methods.

The research reveals specific effects across various assessment types. Akçapınar and Sidan [3] emphasized the possibility of automated solution generation for programming tests, which calls for a change in focus to assess higher-order thinking abilities. Arora et al. [11] discovered that LLMs helped with code generation and debugging for project work, indicating that evaluation should concentrate on decision-making and process rather than just the end result. While noting difficulties in identifying LLM-generated content, Bernabei et al. [12] found that students who used ChatGPT's help produced excellent essays for written assignments.

Because Leinonen et al. [13] found that LLM-generated explanations were often more accurate and easier to understand than student-generated ones, code explanation tasks are particularly disrupted. According to this research, traditional code explanation tasks need to be significantly redesigned in order to continue serving as reliable gauges of students' comprehension.

4 How to evaluate computer science in the era of LLMs

4.1 Emerging assessment strategies

The evaluation of students in computer science in the era of LLMs has necessarily changed since LLMs can help students a lot for many tasks. Usually, IT projects can be partially or totally written by LLMs especially for bachelor students.

Concerning written exams, it seems much more difficult to check that students will not cheat because with only 2 seconds of use of a smartphone is necessary to take a picture of the exam and a LLM can be used to generate a solution of the exam.

Finally, concerning oral presentation, it is different since students are in front of the teachers and even if they can use a LLM to generate some parts of the slides, the students need to be able to answer some questions and to show some parts of the code.

So, clearly, teachers need to take that into account. There are several possible approaches. First, we think that teachers should test their exams or evaluations with LLMs to see their ability to find the solution or not. According to some lecturers, sometimes LLMs are less effective. For example, if students are asked to connect to a remote machine and they have access only to some files for which there are no other clues, it is much more difficult for an LLM to help them. This is typically the case for penetration test exams or debugging code without external indications. One possibility consists of giving an evaluation that is related to a concept that students have studied in class and which is specific enough that LLMs would not be useful.

Researchers have started creating novel assessment strategies that consider LLM capabilities in response to these difficulties. In recognition of prompt engineering as a developing skill deserving of assessment, Denny et al. [14] developed "Prompt Problems" to instruct and evaluate students' capacity to create efficient prompts for LLMs. In order to evaluate both debugging abilities and LLM interaction capabilities, Ma et al. [5] created a system in which students serve as teaching assistants to assist LLM-simulated students in debugging code.

Hybrid evaluation strategies that blend conventional and LLM-integrated techniques are becoming more popular. Grandel et al. [15] tackled the use of ChatGPT-4 as a replacement to human graders, which could increase assessment efficiency and objectivity. In order to evaluate the planning process as well as the capacity to critically assess LLM-generated solutions, Rivera et al. [16] looked at using LLMs to generate code based on student-created plans. An LLM-powered adaptive quiz system that merges engagement and individualized evaluation was created by Kelly et al. [9].

Modern assessments now place a special emphasis on process-focused evaluation. Several studies recommend concentrating on students' decision-making processes, critical analysis, and capacity to enhance LLM-generated work rather than solely assessing final products, which may be significantly impacted by LLM assistance [11, 13, 16].

4.2 Strategies for specific assessment types

Table 2. Evaluation Strategies for Different Assessment Types in the LLM Era

Assessment Type	**Recommended Evaluation Strategies**
Programming Exams and Coding Tasks	• **Problem decomposition focus:** Assess students' ability to break down complex problems into manageable components rather than merely implementing solutions [11]. • **Reasoning documentation:** Require students to document their thought processes and decision-making, explaining why particular approaches were chosen [5]. • **LLM-resistant questions:** Design questions that require contextual understanding, domain-specific knowledge, or reference to course-specific material that LLMs may lack [10]. • **Comparative analysis:** Ask students to evaluate multiple potential solutions (including LLM-generated ones) and justify their selection of the optimal approach [13].
Project Work and Practical Assignments	• **Incremental development assessment:** Evaluate students' work at multiple stages throughout the project,

	requiring explanations of changes and decisions at each milestone [11]. • **Prompt crafting evaluation:** Assess students' ability to construct effective prompts that elicit useful LLM responses, recognizing this as a valuable skill [14]. • **LLM-human collaboration:** Design assignments that explicitly require students to collaborate with LLMs, then evaluate how effectively they leverage these tools while maintaining intellectual ownership [8]. • **Process artifacts:** Require documentation of the development process, including false starts, debugging challenges, and strategic pivots [16].
Written Assignments and Oral Presentations	• **Critical analysis emphasis:** Shift focus from content generation to critical analysis of content, including the evaluation of LLM-generated material [12]. • **Unique perspective requirement:** Design prompts that necessitate incorporation of personal experiences, course-specific discussions, or recent events that LLMs may lack context for [7]. • **Follow-up questioning:** Incorporate adaptive questioning that probes understanding beyond prepared content, particularly in oral presentations [9]. • **Revision and improvement:** Assess students' ability to meaningfully revise and improve upon initial drafts, whether human or LLM-generated [8].

5 Discussion

So, globally, teachers of CS or IT need to take into account that it is important for students to learn how to use LLMs because later, in many companies, they will be asked to do so.

Nevertheless, students also need to be able to reason by themselves, and for that, LLMs are a big problem. In fact, if students do not make huge efforts by themselves, they will not be able to solve problems that LLMs are not able to solve. Even though LLMs have made a lot of progress, there are many situations in which they do not understand the context, the challenge, and the hints needed to find a good solution for a given problem.

Some people will say that professors can be replaced by AI. Okay, that would be possible for some excellent and autonomous students. Many people thought that when MOOCs appeared, for example.

That is why we strongly think that it is important to know the progress that AI is making and what LLMs provide. However, it is also really important to know their limitations and weaknesses.

Based on the findings of this work, educational institutions should create unambiguous policies on LLM use, stating when and how these tools could be properly used [3, 12]. Instructors should second explicitly teach proper and ethical LLM use, including prompt engineering, output assessment, and suitable attribution [14]. Designing assessments should, third, include ongoing review and adjustment as LLM capacity changes [11].

Technically, implementation could mean creating bespoke LLM tools suited to particular educational settings [4], applying organized LLM use guidance [8], and progressively rolling out LLM tools beginning with basic tasks and moving on to more complicated uses [9]. Teachers also need to think about analyzing knowledge of student competency using several assessment techniques [10].

Assessment techniques will need constant improvement as the LLMs' terrain changes. Future studies ought to look at how LLM use affects knowledge retention and skill development over time [8]. Ensuring educational equity will also depend on studies on possible differences in LLM access and competence [7]. Another interesting path is the creation of specialized educational LLMs with suitable limitations and openness characteristics [5, 15]. Evaluating computer science education in the LLM era is difficult, but it finally offers a chance to rethink what knowledge and skills are most useful in a society of more and more powerful artificial intelligence tools [6, 11]. Teachers can guarantee that computer science education keeps producing really competent practitioners even as the technology environment changes by stressing higher-order thinking, creative problem-solving, ethical issues, and efficient tool use in their assessments.

6 Conclusion

LLMs are quite recent, as they became publicly available at the end of 2022. Very few papers have analyzed how their use by CS or IT students can be beneficial and what problems they may cause. This is also an explanation for why only a few papers propose studies in this domain. With this paper, we propose our point of view, which can necessarily be discussed. We hope that teachers will think about these elements for their future lectures.

References

1. Akçapınar, G., & Sidan, E. (2024). AI Chatbots in Programming Education: Guiding Success or Encouraging Plagiarism. Discover Artificial Intelligence.
2. Alves, P., & Cipriano, B. P. (2024). "Give Me the Code" - Log Analysis of First-Year CS Students' Interactions With GPT. ArXiv.org.
3. Akçapınar, G., & Sidan, E. (2024). AI Chatbots in Programming Education: Guiding Success or Encouraging Plagiarism. Discover Artificial Intelligence.
4. Lyu, W., Wang, Y., Chung, T. R., Sun, Y., & Zhang, Y. (2024). Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study. ACM Conference on Learning @ Scale.
5. Ma, Q., Shen, H., Koedinger, K., & Wu, T. (2023). How to Teach Programming in the AI Era? Using LLMs as a Teachable Agent for Debugging. International Conference on Artificial Intelligence in Education.
6. Jošt, G., Taneski, V., & Karakatič, S. (2024). The Impact of Large Language Models on Programming Education and Student Learning Outcomes. Applied Sciences.
7. Padiyath, A., Hou, X., Pang, A., Viramontes Vargas, D., Gu, X., Nelson-Fromm, T., Wu, Z., Guzdial, M., & Ericson, B. (2024). Insights from Social Shaping Theory: The Appropriation of Large Language Models in an Undergraduate Programming Course. International Computing Education Research Workshop.
8. Kumar, H., Xiao, R., Lawson, B., Musabirov, I., Shi, J., Wang, X., Luo, H., et al. (2024). Supporting Self-Reflection at Scale with Large Language Models: Insights from Randomized Field Experiments in Classrooms. ACM Conference on Learning @ Scale.
9. Kelly, K., Wu, B., & Liebe, C. (2024). Gamification Powered by a Large Language Model to Enhance Flipped Classroom Learning in Undergraduate Computer Science. European Conference on Games Based Learning.
10. Quille, K., Becker, B. A., Faherty, R., Gordon, D., Harte, M., Hensman, S., Hofmann, M., Nolan, K. E., & O'Leary, C. (2024). LLMs in Open and Closed Book Examinations in a Final Year Applied Machine Learning Course (Early Findings). Annual Conference on Innovation and Technology in Computer Science Education.
11. Arora, C., Venaik, U., Singh, P., Goyal, S., Tyagi, J., Goel, S., Singhal, U., & Kumar, D. (2024). Analyzing LLM Usage in an Advanced Computing Class in India. ArXiv.org.
12. Bernabei, M., Colabianchi, S., Falegnami, A., & Costantino, F. (2023). Students' Use of Large Language Models in Engineering Education: A Case Study on Technology Acceptance, Perceptions, Efficacy, and Detection Chances. Computers and Education: Artificial Intelligence.
13. Leinonen, J., Denny, P., Macneil, S., Sarsa, S., Bernstein, S., Kim, J., Tran, A., & Hellas, A. (2023). Comparing Code Explanations Created by Students and Large Language Models. Annual Conference on Innovation and Technology in Computer Science Education.
14. Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. (2023). Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators. ArXiv.org.
15. Grandel, S., Schmidt, D. C., & Leach, K. (2024). Applying Large Language Models to Enhance the Assessment of Parallel Functional Programming Assignments. 2024 IEEE/ACM International Workshop on Large Language Models for Code (LLM4Code).

16. Rivera, E., Steinmaurer, A., Fisler, K., & Krishnamurthi, S. (2024). Iterative Student Program Planning Using Transformer-Driven Feedback. Annual Conference on Innovation and Technology in Computer Science Education.
17. Kumar, H., Musabirov, I., Reza, M., Shi, J., Kuzminykh, A., Williams, J., & Liut, M. (2023). Impact of Guidance and Interaction Strategies for LLM Use on Learner Performance and Perception. Proc. ACM Hum. Comput. Interact.
18. Ma, Q., Shen, H., Koedinger, K. and Wu, S.T., 2024, July. How to teach programming in the ai era? using llms as a teachable agent for debugging. In International Conference on Artificial Intelligence in Education (pp. 265-279). Cham: Springer Nature Switzerland.
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

27. Innovative Autonomous Vehicle Designs for Enhancing Mobility of Disabled Individuals

Bekir Karlik[1], Dylber Caushi[2]

[1,2]Department of Computer Engineering, Faculty of Engineering and Architecture, Epoka University, Rruga Tiranë-Rinas, Km 12, 1039, Tirana/Albania Albania,
bkarlik@epoka.edu.al, dcaushi18@epoka.edu.al

Abstract. Intelligent automation systems are specially designed vehicles that provide effective and rapid solutions to complex problems and provide efficiency and progress in various fields. Recent developments in artificial intelligence and smart sensors have provided significant progress in the design of autonomous vehicles. The feature of autonomous vehicles is that they can perform all driving activities such as perceiving their environment, monitoring their path, and controlling the vehicle from departure to destination on their own. This study presents a prototype design of an intelligent autonomous vehicle that follows a color-based line on its path. Autonomous vehicles aim to increase speed, efficiency, and accuracy while monitoring the path and reduce errors. The components used in autonomous vehicle design are an infrared (IR) sensor, Arduino, motor driver, freenove starter kit, and motors. The data of the road information detected by the IR sensors is sent to Arduino. The desired road selection is determined by the k-NN supervised learning algorithm. and then the motor driver connected to it is controlled by Arduino. This study proposes hybrid PID/k-NN based control for two assistive autonomous vehicles prototypes as line tracking and hand tracking. This proposed autonomous vehicle is suitable to use for electrical wheelchair by disabled people for enhancing their mobility. Moreover, it can be used for delivery purposes, self-driving cars, transportation, and industrial purposes. The development of autonomous vehicle technology can also increase the use of such vehicles for the comfort of disabled people.

Keywords: Autonomous Vehicles, Intelligent automation, Machine Learning, k-NN, Arduino uno

1 Introduction

Today, with rapid technological advancement, intelligent adaptive automation systems are quite advanced and are used in all kinds of jobs, from daily house cleaning to medical devices and complex industrial tasks. One of these application areas is autonomous vehicles that follow the line or path on their own. Artificial Intelligence is widely used in the automotive sector, especially in autonomous driving technologies, and its main purpose is to increase safety, fuel efficiency and to help drivers on long journeys on highways or when they fall into micro-sleep. For example, Mercedes Benz has made an incredible machine learning composition in its all-electric latest type "EQS", where a camera and sensor placed on the steering wheel detect movements such as fatigue and inattention, take into account various factors such as driving behavior, steering wheel movements, driving time and many more factors, and then the vehicle warns the driver to take a break audibly and visually [1].

Robot (or autonomous vehicle) designs that examine lines drawn on the ground have been made for the last 2 decades [2-4]. In all these studies, IR sensors or infrared distance sensors were used to determine the black color line on the ground. Then, line follower robots that can select the target line between black and white or different colored lines were developed. In the systems of these vehicles are used IR sensor array, LDR-based color sensor, ultrasonic sensors, and RFID-based identification sensors. Later, artificial intelligence-based smart sensors were developed for color perception and subsequently, designs of smart autonomous line-following vehicles or robots began [5-7]. Previously, a study was conducted using the k-NN technique to track vehicles in real time and estimate the distances between stops using the haversine formula [8]. Here, GPS technology was used to analyze the spatial dynamics of the vehicle's movement. The study addresses the unpredictability of urban traffic patterns, which has a significant impact on bus travel times, by dividing the data into different temporal segments such as rush hours and off-peak hours and weekdays and weekends.

Machine learning (ML) techniques is different from the other traditional learning methods in that no assumptions are made about the types of relationships between variables [9]. The last decade, intelligent car automation system has been using machine learning algorithms [10]. ANN is a subset of machine learning and a very important part in deep learning algorithms. The k-NN approach can increase the prediction accuracy by dividing the vehicle route into smaller segments for more in-depth study with the time segmentation mechanism. Therefore, this method provides more precise predictions that can respond to sudden changes in traffic conditions.

Recently, deep learning methods have been used in autonomous line-following vehicles. In these applications are used hybrid sensors or sensor fusion like LiDAR and Radar sensors, and cameras, so the data size is very large data. Thus, they needed to use deep learning algorithms such as Reinforcement Learning, CNNs, ResNet (or Residual Networks) etc [11-15].

In this study, two prototype vehicles that follow the intelligent line tracking or hand tracking are proposed using the freenove starter kit for Arduino UNO and a machine learning algorithm (K-Nearest Neighbor). Line-following vehicles follow the line using sensors. The vehicle will follow a black line on a white surface and decide which way to go to the roundabout. Many line-following vehicles have been produced, but the main challenge is to make them fast-reacting, precise, and adaptable to various environments. The sensor accuracy used in this study and developed control algorithms are better.

This practical work will increase interest in intelligent control systems not only in industrial applications but also in education. The line-tracking autonomous vehicle, also called a line-following robot, is designed to follow a predefined path or line on the ground. It uses sensors to detect the line (usually a black line on a white surface or vice versa) and moves along it. In the other hand, the hand-tracking autonomous vehicle is designed to track and respond to the movement of a human hand. These robots use sensors like cameras, infrared, or depth sensors to detect and interpret hand gestures or positions [16]. These two prototype autonomous vehicles aim to design and implement a smart line following autonomous vehicle using a freenove starter kit for Arduino. The study uses Arduino to control vehicle movements and make the right decision at the right time. Recent works [15-16] combine sensor fusion with RL for wheelchair navigation, but our PID/k-NN approach offers lower computational cost. Thus, this study proposes a hybrid PID/k-NN control system for two autonomous vehicle prototypes: a line-tracking robot and a hand-tracking assistive cart. Unlike recent sensor-fusion approaches (e.g., Zhang et al., 2025), our method reduces computational costs while maintaining accuracy (>95% line-tracking precision). The prototypes, built with Arduino and Raspberry Pi, demonstrate potential for disabled mobility, delivery services, and industrial applications

2 Materials and Methods

2.1 PID Controller

The efficiency of line-following autonomous vehicles depends on their balance of performance, ability to navigate autonomously on a predefined line, and good response to dynamic conditions. When sensors detect that the robot has lost its path, the robot's control system will try to overcome this error and bring the robot back on track, but if the control system is inefficient, it will oscillate along the line and will not be able to reach balance and gain speed. To avoid these situations, it is necessary to use an intelligent and effective control system in line-following robots. In this work, a proportional integral derivative (PID) control strategy was used to develop a fast, precise, and stable robot using a freenove start-er kit for Arduino.

Feedback control systems are always more stable than the others. The feedback of the output signal finds the error between the set point and the feedback signal. This error indicates the deviation of the system response from the target set point. Based on this error the system always tries to correct itself. The PID control system operates on the principle of minimizing the error between the set point and the system actual value by using a feedback control mechanism, and the difference between these values is called error (see Figure 1). The total output of the PID controller u(x) which is the sum of three terms, Proportional, P(x), Integral, I(x), and Derivative, D(x) is expressed as.

$$u(x) = P(x)+I(x)+D(x)=K_p.e(x) + \boldsymbol{K_i}.\int_0^t e(x)dt + K_d.(de(x))/dx \qquad (1)$$

where Kp is the proportional gain; e(t) is the error at x time. Ki and Kd are the integral and derivative gains respectively; (de(x))/dx is the derivative of the error at time respect.

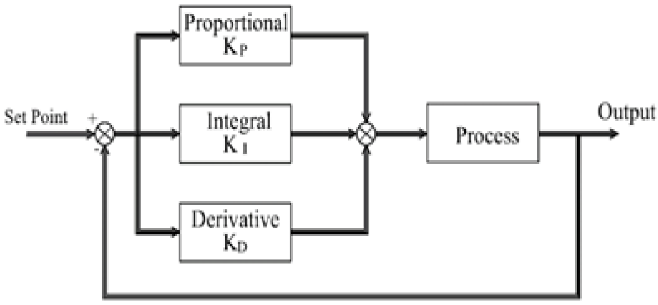


Fig. 13. Closed loop operation of PID controller

The output response of the controller of PID defends the tuning of its three coefficients K_p, K_i, and K_d. Adjusting proper values for these three constants will give the desired response and will minimize the damping oscillations and overshoots. One of the most important features of the PID control system, which is frequently used in industrial applications and robots, is that it has a feedback mechanism that allows dynamic systems to improve their stability and performance.

In this study the PID algorithm is implemented in the programming code of LFR. The algorithm enables the autonomous vehicle to follow the line precisely and with high accuracy even if the autonomous vehicle goes off track it resets itself online.

2.2 Implementation of Hardware and Software Setups

What we should not forget is that in this study, a miniature prototype of the autonomous vehicle was designed. Naturally, these hardware components will be very different for a large vehicle. However, the software we will implement will be almost the same. The following hardware was chosen to make a miniature autonomous vehicle that follows a line:

- An array of 8-channel Infrared sensors for line detection and environmental observation,
- Arduino UNO is used for processing and deciding on the response of IR sensors,
- L293D motor driver to control the speed and power of the DC motors and driver follows the Arduino commands,

Battery for DC motors, power supply, wheels for movement.

As seen in Figure 2, the hardware setup of functional line follower vehicle consists of following components: Chassis and frame, DC motors and wheels, 8 IR sensor array, Arduino. Motor driver. The chassis and frame hold and give support to all the components. The selection of chassis is very important in the case of making a robot because the overall structure of a line follower autonomous vehicle depends on the frame and the structure directly affects the performance of the robot. The frame selected for this robot is strong enough to support the components and the battery weight.

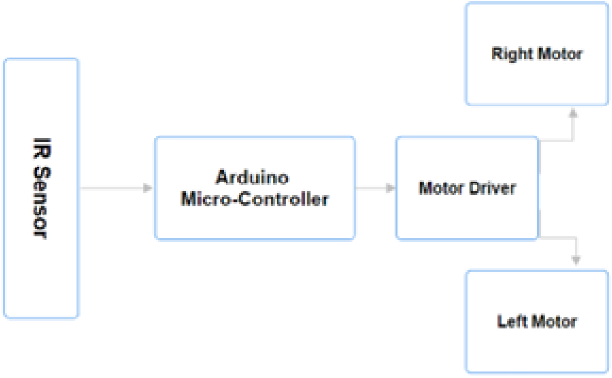


Fig. 2. Line follower autonomous vehicle block diagram

DC motors are mounted on the right and left sides of the chassis. The wheels are attached to the motor shaft and the 3rd wheel is independent and mounted on the front and can rotate 360^0 DC motors connected through wires to the L293D motor driver. IR sensors and motor drivers are directly connected to Arduino. We used Arduino UNO that is powered by AT-Mega328P microcontroller and has fourteen input output pins. It receives data from IR sensors based on that data it gives commands to the motor driver. Arduino is mount-ed on the frame, and IR sensors and motor drivers are directly connected to Arduino. It receives data from IR sensors based on that data it gives commands to the motor driver. The software used to write a program for the Arduino microcontrollers is Arduino IDE. The program defines the algorithm used. The program is written through Arduino IDE and when uploaded to Arduino, it becomes capable of processing data, making decisions, and controlling the motors.

Coding without Obstacle detection (Existing Code)

The code is divided into parts and each part is explained:

Variable Declaration: These lines declare the integer variables to store pin numbers for motor speed control, IR sensor reading and threshold values for sensors.

- Setup Function: The function here is the configuration of Arduino pins for motor control. The function "setup" calls setup motor (1,1) to initialize the motor to move forward.
- Brake Function: It sets both DC motors for the brake when the signals to its pins are high.
- Setup Motor Function: It configures the direction of each motor based on the values of the two variables as forward a and forward b.
- Change Speed Function: It can be changed the duty cycles of PWM signals.
- Read Eye Function: this function reads analog values from the sensor and compares them with threshold values, if the reading is below the threshold, it returns 0, otherwise 1.

Coding with Obstacle detection

Obstacle detection is a good feature of the line follower autonomous vehicle. With this feature, the autonomous vehicle can avoid obstacles coming in its path deciding to turn left or right and managing to come back to the predefined path. To add this feature, we need a servo motor and ultrasonic sensor. A portion of the code is given which uses a servo motor and ultrasonic sensor to avoid obstacles.

The autonomous vehicle will continue its forward movement through the line until an obstacle comes in its path. The line follower robot checks the distance ahead by using an ultrasonic sensor (distance F). If the distance ahead is greater than the predefined threshold (set) the forward function (forward) moves the autonomous vehicle forward. When the distance is less than the threshold or equal to the threshold, the robot stops and calls a function called Check Side.

2.3 Calibration and Rasberry Pi

Calibration is a crucial step while designing an autonomous vehicle. calibration involves the configuration of software parameters to match the specific characteristics of sensors, the environment, and the robot. The calibration of the sensor threshold determines the difference between the lines and the background. It determines the analog voltage levels corresponding to the robot positions on the line, off the line, or at the edge of the line. These threshold values help the robot algorithm to make a correct decision based on the sensor readings. The environment is properly set for the robot like a black line is set as a path on the white surface.

Raspberry Pi is a computing device featuring a processor capable of handling complex algorithms, in this case input from color sensors which will use a photodiode array to detect the color. It is “the brain” of the robot makes real time decisions, guiding the robot to stay along the track with precision. Integrating a Raspberry Pi with a color sensor helps us to differentiate the main color with other colored lines that are on the same track. This process is achieved by taking some steps which will convert raw color sensor data to color temperature. By using RGB (red, green, and blue) values sensed by the sensor we estimate the color temperature of the light source that is illuminated and based on concept of blackbody radiation, a specific temperature can be mapped on the CIE 1931 Planckian Locus (see Fig. 3).

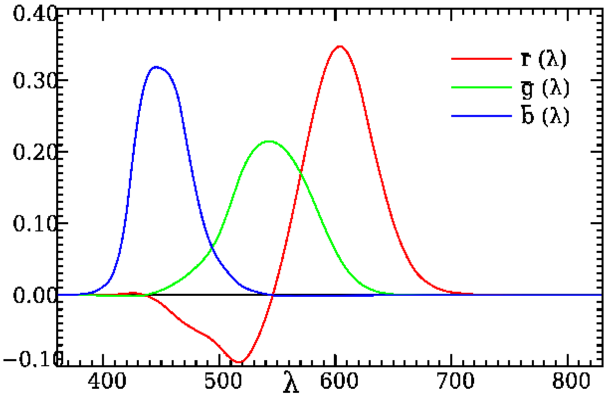


Fig. 3. Normalized curve of color matching functions

The following Fig. 4. Motor driver IC [13] shows the pinout of the l293D IC.

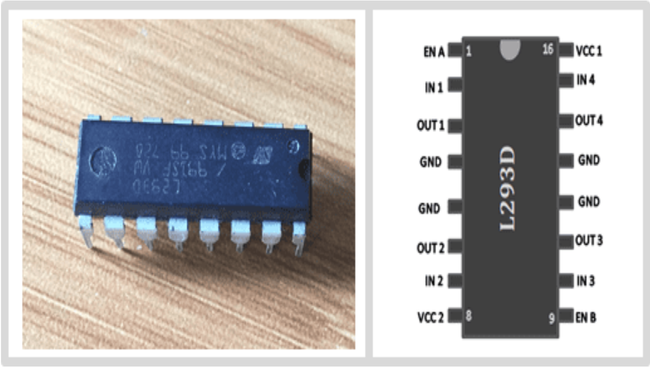


Fig. 4. Motor driver IC

There are a total of 16 pins of which two pins are power supply pins VCC1 and VCC2.

- VCC1 is the power supply for the circuitry and its voltage should be 5V, usually, VCC1 is connected to the Arduino.
- VCC2 is the power supply for the H bridge and its voltage ranges from 4.5V to 36V and is used to run the DC motor.
- There are four output terminals, two controls and two motors. One motor is connected to the output terminals OUT1 and OUT2 while another motor is connected to terminals OUT3 and OUT4.
- ENA and ENB are used to control the speed of the DC motors, when they are high the speed of the motor is high and when they get low the motors stop. Arduino sends PWM signals to these pins to control the speed of the motors.
- IN1, IN2, IN3, and IN4 pins control the direction of rotation of the motors.
- Pins 4 and 5 are the ground pins of the IC and provide ground connection as well as act as a heat sink for IC.

L293D motor drives have two H bridges. Each H bridge has four MOSFET transistor switches S1, S2, S3 and S4. At the same time, two switches must be activated to maintain the flow of current in a particular direction to rotate the motors in reverse or forward directions. The L293D motor vehicle consists of two H-Bridge circuits and controls of two DC motors simultaneously. The four pins from IN1 to IN4 control the switches of the H-Bridge circuit of the L293D motor driver module. The rotation of the DC motors depends on the input values to these four IN pins. If IN1 is low and IN2 is high the motor will rotate direction say clockwise and if IN1 is high and IN2 is low, then the motor will rotate in the opposite direction say anticlockwise. If both inputs have the same signal, the motor will stop rotating if IN1 and IN2 are both low or both high. The following table shows the corresponding direction of the motor based on the state of the four inputs (from IN1 to IN4) [1].

Unlike the first line follower, this advanced robot can handle more complex tasks and have higher efficiency in data computation. To achieve this kind of thing, Arduino Uno is not feasible because of the memory and limited processing power, so we must integrate a Raspberry Pi. Here we will use machine learning algorithms, particularly the k-NN (k-Nearest Neighbors) which empowers the autonomous vehicle with enhanced decision making. This makes the autonomous vehicle learn from the environment and improve decision making and adaptability. Example-based classification methods are based on predicting the class of a new pattern using known patterns in the training set, the k-NN method is a supervised machine learning algorithm that is simple software but effective and compatible with embedded systems [17].

Table 1. The direction of rotation of motion based on the state of IN (0: Low, 1: High)

State of IN 1	State of IN 2	Direction of Rotation of Motion
0	0	Stop
0	1	Clockwise
1	0	Anti-Clockwise
1	1	Stop

2.4 k-NN Algorithm for line followers:

The k-NN supervised learning method is a simple machine learning algorithm that is often used for classification and detection problems. In the case of a line-following robot, the k-NN algorithm can be applied to make decisions based on sensor readings, such as whether the autonomous vehicle should turn right/left or go straight with respect to the environment. A C++ code that implements the k-NN algorithm consists of the following steps:

Sensor Data: We assume that the robot has 3 sensors: left, center, and right. The sensor readings can range from 0 (line not detected) to 1 (line detected).

Training Data: We store the historical sensor readings along with the corresponding actions (Straight, Left, Right).

k-NN Algorithm: Given a new sensor reading, the algorithm will find the K-NN (by Euclidean distance) and classify the action based on most neighbors.

Action: The robot will move according to the classification: “Left”, “Right”, or “Straight”

Step-by-Step Approach:

Sensor Data: We assume the robot has 3 sensors: left, center, and right. The sensor readings can range from 0 (no line detected) to 1 (line detected).

Training Data: We store the historical sensor readings along with the corresponding actions (Straight, Left, Right).

k-NN Algorithm: Given a new sensor reading, the algorithm will find the nearest neighbors (based on Euclidean distance) and classify the action based on most neighbors.

Action: The robot will move based on the classification: either "Left", "Right", or "Straight".

Code implementation is given in the appendix.

3 Performance Results and Discussions

In this study a prototype autonomous vehicle has presented as intelligent line follower autonomous vehicle using k-NN machine learning algorithm. Although k-NN requires less iterations for both training and testing, its computational complexity is low, and it calculates fast [17]. LFR was successfully implemented using Arduino UNO. The robots are programmed using Arduino IDE software and the PID algorithm is implemented in the programming code of LFR. The algorithm makes the robot capable of following the line precisely and with high accuracy. Even if the robot goes off track it resets itself online. The final model of the line follower robot is given in Fig. 5. This model successfully implemented and followed the predefined line with high accuracy and stability.

The autonomous vehicle is made from the integration of different components; each component has its own rule in the functionality of a line follower autonomous vehicle. An array of 8 infrared sensors works as an eye of the line follower the autonomous vehicle. IR sensors observe the position of the autonomous vehicle on the line and convert these positions and physical data to voltage signals. These signals are processed by Arduino, which acts as the brain of the line follower robot and real-time take decisions. L293d motor driver receives the control signals from Arduino and controls the speed and direction of rotation of the DC motor. The line follower robot uses the PID controller algorithm for the controller autonomous vehicle’s functions. The proportional, integral, and derivative terms work together to process the sensor data and make real-time decisions to control the speed of DC motors and maintain the robot on the line. The PID algorithm minimizes the error and maintains the robot on the track with high accuracy and stability. The results show that the finalized model was able to successfully follow the black line on the white surface with high accuracy and stability.

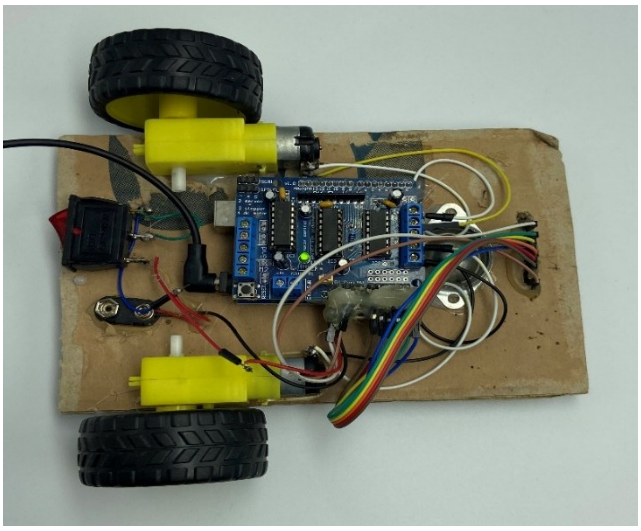


Fig. 5. Proposed intelligent line follower autonomous vehicle

The creation of a hand tracking autonomous vehicle prototype utilizing the Arduino UNO to help elderly people in supermarkets is a big step toward improving the elderly`s shopping experience and autonomy. This work not only highlights the practical application of robots in everyday life, but it also shows the possibility for combining basic, low-cost technologies to address real-world problems. The vehicle uses hand tracking technology to closely follow users without requiring physical touch, removing the need for elderly people to push heavy shopping carts, which may be physically and emotionally taxing. The final model of the line follower robot is given in Fig. 6.

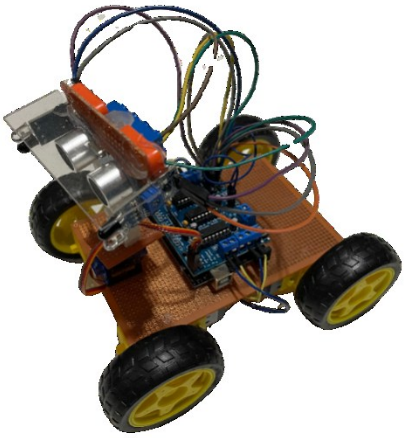


Fig. 6. Proposed intelligent hand tracking autonomous vehicle

4 Conclusions and Future Works

This study presented a Line follower autonomous vehicle which is a dynamic and simple autonomous vehicle system designed to follow a predefined black line on a white surface. It is suitable to use for electric wheelchair be used disabled people. The primary subject was to design such an autonomous vehicle capable of navigating a predefined route with high accuracy. Line follower autonomous vehicles have a lot of industrial applications. These types of vehicles are also adaptable to every daily task such as house cleaning, transportation [19], carrying goods delivery services, land use, environment, public health [20], and real time car parking [21-22]. Autonomous vehicles have facilitated parking requests due to their operational model.

The main goal of our initiative is to create and install robotic assistants specifically designed to meet the demands of the elderly population, especially in large retail stores. This initiative aims to make it easier for customers to acquire and transport items from the point of purchase to their cars or directly to their homes. A critical component of this goal is the development of a strong municipal infrastructure that supports and enhances the operational effectiveness of these robotic assistants. In addition, in military operations, the use of mod-ern sensors specifically designed to detect gas leaks represents a significant technical leap forward with the potential to trans-form security regulations and reconnaissance missions. By incorporating these sensors into military equipment, soldiers can proactively detect and inspect areas suspected to be contaminated with toxic gases, eliminating the need to physically approach these potentially dangerous areas.

The study serves as a prototype that could be refined and expanded upon, with future iterations possibly incorporating advanced features such as obstacle avoidance, voice commands, and more sophisticated user interaction capabilities. Moreover, the development of autonomous vehicle technology can also increase the use of such vehicles for the comfort of people. While the current iteration of the line follower autonomous vehicle demonstrates promising performance, there are opportunities for future enhancements. Potential areas of improvement include:

- Integration of advanced sensors for improved line detection in diverse environments.
- Implementation of machine learning algorithms for adaptive navigation and obstacle avoidance.
- Expansion of vehicles capabilities to handle dynamic, unstructured environments.
- Improvement in the design and overall shape can increase the performance like a robot having a balanced structure can easily turn in the short curves.
- These enhancements have the potential to further broaden the applicability and effectiveness of the line follower robot in real-world scenarios.

There are some challenges in the use of autonomous vehicles. We can briefly list them as follows:

- If autonomous vehicles become available very quickly, this could lead to more personal vehicles on the roads, thus causing urban traffic congestion.
- Autonomous vehicles will generally need clear lane markings. These lines may not be accurately predicted when environmental and weather conditions, etc. cause the lanes to deform.
- One of the most uncertain areas for autonomous vehicles is legal liability, the other is insurance. How will insurance companies handle minor accidents that occur when a driver is driving and is careless? Who will be the driver in these accidents legally; who will be ultimately responsible?

IR sensors may fail in high-glare environments; future work will test LiDAR alternatives. Future iterations will integrate LiDAR for robust obstacle detection in glare-prone environments and voice-command interfaces for users with limited hand mobility. Legal frameworks for liability in autonomous assistive devices will also be investigated

Countries will need to take common legal decisions to solve these difficulties that will arise from the frequent use of autonomous vehicles.

References

1. Caushi D. Intelligent line follower and hand tracking robot using Arduino uno. Master Thesis, Epoka University, February 2024.
2. Liu SC, Liu GJ. (2007). Formation control of mobile robots with active obstacle avoidance. Acta Automatica Sinica.;33(5):529-535.https://doi.org/10.1360/aas-007-0529

3. Colak I, Yildirim D. (2009). Evolving a line following robot to use in shopping centers for entertainment. 35th Annual Conference of IEEE Industrial Electronics, Porto, Portugal, pp. 3803-3807, https://doi.org/10.1109/IECON.2009.5415369
4. Pakdaman M, Sanaatiyan M, Ghahroudi MR. (2010). A line follower robot from design to implementation: Technical issues and problems. The 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, vol. 1, pp. 5-9. https//doi.org/10.1109/ICCAE.2010.5451881.
5. Román Osorio C, Romero JA, Mario Peña C, López-Juárez I. (2006). An intelligent linefollower mini-robot system. International Journal of Computers, Communications & Control. 1(2):73-83. https://doi.org/10.15837/ijccc.2006.2.2287
6. Rishabh K. (2021). Design of autonomous line follower robot with obstacle avoidance. Intenational Journal of Advance Research, Ideas and Innovations in Technology. 2021; 7(3):715-20. https://doi.org/10.13140/RG.2.2.11652.86403
7. Rajesh P, Kumar AS, Manisha Sarayu TK, Prasad AD, and Sai Dhanush C. (2024). PathTraversal and Obstacle Sensing Robot using Arduino. Second IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), Kurukshetra, India, pp. 1-5. https://doi.org10.1109/ICMICA61068.2024.1073236
8. Kumar BA, Vanajakshi L. (2017). Subramanian SC. Bus travel time prediction using a time-space discretization approach. Transportation Research Part C: Emerging Technologies. Jun 1, 2017; 79:308-32. https://doi.org/10.1016/j.trc.2017.04.002
9. Bucak IO, Karlik B. (2011). Detection of drinking water quality using CMAC based artifcial neural Networks. Ekoloji. 2011 Jan 1;20(78):75-81. doi:10.5053/ekoloji.2011.7812
10. Ozyigit, H. A., Oz, H. R., & Karlik, B. (2001). Active suspension control for vehicles andnumerical calculations by using artificial neural networks. International Journal of Appied Mechanics and Engineering, 6(3), 613-626.
11. Smith A, Wang J, Kumar R. (2024). Autonomous line following robot using deep learning. Journal of Autonomous Systems. 12(3): 234-245. https://doi:10.1016/j.autsys.2024.01.007
12. Patel S, Wong T, and Sharma M. (2023). Line following robot with a hybrid sensor approach. Int. Journal of Robotics and Automation.15(2):56-70. https://doi.org/10.1109/IJRA.2023.009567
13. Johnson, L., Thomas, K., and Lee, V. (2025). Implementation of intelligent line followers using machine vision and LIDAR. Robotics and Vision Systems. 19(1):88-101. https://doi.org/10.1109/RobVis.2025.008917
14. Patel, D., Bansal, S., & Singh, R. (2024). Self-learning line following robots using reinforcement learning algorithms. Autonomous Robotics Journal. 22(4): 317-329. https://doi.org/10.1007/s10514-024-732-9
15. Zhang H, Liu G, and Desai P. (2025). Sensor fusion for autonomous line following vehicle: A multi-sensor approach. IEEE Transaction on Robotics. 31(2);441-455. https://doi.org/10.1109/TRO.2025.011234
16. Yang, R., Yuan, Y., & Zeng, G. (2012). Real-Time Hand Gesture Recognition with a Depth Camera. In Intelligent Robotics and Applications: 5th International Conference, ICIRA 2012 (pp. 83-94). Springer Berlin Heidelberg
17. Esme E, Karlik B. (2019). Design of intelligent garment with sensor fusion for rescue teams. Journal of the Faculty of Engineering and Architecture of Gazi University 34:3 1187-1200. https://doi.org/10.17341/gazimmfd.460

18. Karlik, B. Bayrak Hayta. S. (2014). Comparison of machine learning algorithms for recognition of epileptic seizures in EEG. Proceedings IWBBIO 2014, pp. 1-12.
19. Pisarov, J. & Mester, G. (2021). The Use of Autonomous Vehicles in Transportation. Tehnika. 76:171-177. https://doi.org/10.5937/tehnika2102171P.
20. Karolemeas, C., Tsigdinos, S., Moschou, E. et al. (2024). Shared autonomous vehicles and agent-based models: a review of methods and impacts. Eur. Transp. Res. Rev. pp. 16-25 https://doi.org/10.1186/s12544-024-00644-2
21. Yusuf FH, Mangoud MA. Real-Time Car Parking Detection with Deep Learning in Different Lighting Scenarios. Inter. Journal of Computing and Digital Systems. 2025:17:1 1-9.https://iiict.uob.edu.bh/IJCDS/papers/1570964473.p
22. Othman K. Exploring the implications of autonomous vehicles: a comprehensive review. Innov. Infrastruct. Solut. 2022;7(2):165. doi: 10.1007/s41062-022-00763-6. Epub 2022 Mar 1. PMCID: PMC8885781., last accessed 2016/11/21.

28. Automata and Formal Language in Higher Education: A Comprehensive Review

Ermira Idrizi[1] and Florije Ismaili[2]

[1,2] South East Europen University, Tetovo, North Macedonia
e.idrizi@seeu.edu.mk[1], f.ismaili@seeu.edu.mk[2]

Abstract. This paper presents a systematic literature review focused on pedagogical approaches and methodologies employed in teaching Automata and Formal Language courses within higher education. Recognized as fundamental components in understanding formal languages, compiler construction, and computational theory, Automata and Formal Language theory are essential in computer science curricula. However, due to their inherent abstract nature and conceptual complexity, these topics often pose significant challenges to students. To address these difficulties, various educational tools and strategies – including traditional lectures, interactive digital tools, visualization simulators, and innovative gamified environments – have been utilized to enhance student comprehension and engagement. Nevertheless, simulations and visualizations alone frequently fall short of adequately reinforcing abstract concepts. In response, this review explores an emerging methodology that involves students actively developing simulators, enabling deeper interaction with and exploration of formal language concepts. This approach encourages practical experience and hands-on engagement, significantly improving conceptual understanding. The review also identifies a crucial gap regarding the lack of standardized assessment methods, suggesting a need for further research to develop consistent evaluation practices. The outcomes of this study provide valuable insights for educators aiming to optimize teaching effectiveness and promote deeper understanding among students in higher education.

Keywords: Automata Theory; Formal Languages; Higher Education; Pedagogy; Educational Tools; Active Learning; Assessment.

1 Introduction

Automata Theory and Formal Languages (FL) form a core part of the theoretical computer science curriculum, underpinning topics like compilers, programming languages, and computational complexity. Despite their importance, these courses are notorious for being abstract and difficult, often leading to low student motivation and high failure rates [2]. Traditional teaching approaches have tended to emphasize formal definitions and mathematical proofs; while rigorous, this can alienate students who struggle to connect theory with practical applications. Over the years, educators have documented persistent learning difficulties in these subjects – from trouble grasping abstract formalisms to an inability to see the relevance of automata theory in real-world contexts [2][17]. Students commonly perceive automata/FL content as "tedious and complex," contributing to poor engagement. These challenges have spurred extensive research into how to improve the teaching and learning of Automata/FL in higher education. A wide range of innovative instructional methods have been explored to make these topics more engaging. Early efforts introduced software tools to provide visual and interactive demonstrations of automata in action. For example, the JFLAP tool (Java Formal Languages and Automata Package) enables students to design and test automata graphically, making the learning process more approachable [1]. In recent years, online learning platforms and automated tutors – such as Automata Tutor – have emerged, offering immediate feedback on exercises and freeing instructors from tedious grading [5]. Active learning strategies and course format innovations have also been tried: some instructors incorporate in-class problem solving, labs, or even flipped classroom models to engage students, while others use gamification (e.g., puzzles and games) to motivate learners. Newer constructivist approaches encourage students to create artifacts (like building their own automata simulators or games) to learn by doing, potentially yielding deeper understanding. Each approach offers potential advantages but also comes with its own set of limitations. There is increasing recognition of the need for better assessment practices to evaluate these teaching methods and compare their effectiveness across different contexts.

Prior work in computing education has reported on these various approaches, but results are scattered. To date, there have been only a few attempts to systematically review and compare teaching methods for Automata/FL. Chakraborty et al. [3] provided a historical review of automata simulation tools over fifty years, highlighting the progression of software aids but offering less insight into pedagogical outcomes. More recently, Veiga da Silva et al. [14] conducted a broad systematic review of automata theory education (including both higher education and K-12) and noted, among other findings, a lack of standardized assessment methods in current practice. Building on and complementing these works, our review focuses specifically on higher education and emphasizes pedagogical dimensions: the teaching tools in use, how student learning is assessed, the design of courses, and differences in practice across regions. In this paper, we present a comprehensive systematic review of the literature on teaching Automata and Formal Languages in higher education. We synthesize findings from numerous studies (primarily from 2010–2024) to address the following research questions:

- **Teaching Tools:** What educational tools and technologies have been used to teach Automata/FL, and how effective are they?
- **Assessment:** What assessment techniques are employed to gauge student learning, and what are their strengths and weaknesses?
- **Course Design:** How have instructors structured and delivered Automata/FL courses (traditional vs. active learning approaches), and with what outcomes?

The goal is to identify evidence-based strategies and gaps in knowledge, providing guidance for instructors and informing future educational research in this domain. The remainder of the paper is organized as follows: Section 2 discusses related work, Section 3 describes our review methodology, Section 4 presents the results by thematic areas (tools, assessment, course design), Section 5 and 6 provides discussion and recommendations.

2 Related Work

Efforts to improve Automata and Formal Language (FL) education date back several decades, with numerous studies emphasizing the integration of visualization and simulation tools in classroom settings. Since the 1990s, software tools like JFLAP have provided animations of abstract machines and grammars, enabling students to construct automata and observe their immediate behavior with specific input strings. Cavalcante et al. [1] highlighted that the use of JFLAP significantly improved the accessibility and intuitiveness of automata concepts, enhancing student engagement.

Recent years have seen the development of advanced simulation tools, intelligent tutoring systems, and AI-driven feedback extensions. Notably, intelligent tutors such as Automata Tutor have emerged, providing instant personalized feedback and iterative practice, leading to deeper understanding and active learning. Bezáková et al. [12] introduced an intelligent feedback extension to JFLAP, significantly aiding students through explicit counterexamples and real-time corrections, thus deepening their conceptual understanding. Other modern educational tools include the FLAT tool for experimenting with automata and grammars, interactive e-textbooks with embedded visualizations, and adaptive learning platforms. Mohammed et al. [11] demonstrated that integrating interactive elements with automated assessments significantly improves student performance and satisfaction. Beyond these tools, pedagogical innovations continue to evolve. Cognitive apprenticeship models, as explored by Knobelsdorf et al. [10], have shown improvements in student problem-solving skills through active classroom engagement. Gamification remains a strong trend, with studies like Korte et al. [9] demonstrating its effectiveness by enabling students to construct games related to automata, thus making abstract concepts more concrete and engaging. Chesnevar et al. [15] emphasized the importance of early didactic strategies that highlight the value of active learning, even in traditionally theoretical courses [16][17].

Overall, contemporary research stresses the importance of an integrative approach combining visualization, interactive technologies, intelligent feedback, and active pedagogical strategies [18]. This review consolidates recent advances and addresses identified gaps, particularly the need for standardized assessment tools, to inform future pedagogical practices in Automata and Formal Languages education[20][21].

3 Methods

We conducted a systematic literature review following the ***PRISMA 2020*** guidelines (see Kitchenham [19] for general review procedures). This review systematically searched ACM Digital Library, IEEE Xplore, SpringerLink, and Google Scholar using specific search terms including "automata theory," "formal languages," "education," "interactive tools," "simulation," "visualization," "AI-driven tutors," and "automata tutor." Selection criteria limited studies to those published from 2019 to 2024, explicitly focused on higher education. Backward snowballing included reviewing cited references from selected articles up to December 2023 to ensure comprehensiveness. We searched multiple scholarly databases (*ACM Digital Library, IEEE Xplore, SpringerLink, and Google Scholar*) for articles published in roughly the last 20 years (2000–2024) related to teaching automata theory and formal languages. Key search terms included combinations of “automata”, “formal language”, “education”, “teaching”, “learning”, and specific tool names (e.g., “JFLAP”, “Automata Tutor”). We also examined the reference lists of relevant papers (backward snowballing) to find additional studies. Our initial search yielded around 200 records after filtering out clearly irrelevant hits (such as pure theory papers without an education focus). We then removed duplicates (approximately 20), leaving ~180 unique records for screening. Inclusion & Exclusion Criteria: We included studies that specifically discuss pedagogical approaches, tools, or assessment methods for Automata Theory and/or Formal Languages in a higher education context (undergraduate or graduate level courses). This covered both research papers (conference or journal articles) and notable experience reports or theses if they provided evaluative insight. We also included studies focusing on closely related theoretical CS courses (like “Theory of Computation”) if they covered automata topics. We excluded papers that solely presented automata theory content (theorems, algorithms) without an educational evaluation, as well as those set in K-12 or secondary education contexts (unless they offered transferable insights for higher ed). At the title and abstract screening stage, about 50 papers were deemed relevant enough for full-text review. From these, we excluded works that did not meet our criteria upon deeper reading (e.g., a tool introduction with no usage data, or non-English papers where translation was not feasible) [22][23]. We ultimately arrived at 30 primary studies that met all criteria and form the basis of our review. Data Extraction & Analysis: For each included study, we extracted key information such as the pedagogical approach or tool investigated, the context (class setting, level, region), the evaluation method (surveys, exams, pre/post-tests, etc.), and the main outcomes (qualitative or quantitative results on student learning or engagement). To address our research questions, we grouped findings into thematic categories corresponding to our focus areas: **Teaching Tools, Assessment Methods, Course Design [24][25].**

Within each theme, we compared results across studies. For quantitative outcomes (e.g., exam score changes), we noted reported gains or effect sizes when available; for qualitative outcomes (e.g., student feedback), we compiled representative observations.

We also noted any reported drawbacks or challenges of an approach (such as increasedinstructor workload or mixed student reception). Given the heterogeneity of study designs (ranging from controlled experiments to anecdotal classroom reports), we did not perform a meta-analytic synthesis. Instead, we used a narrative synthesis approach. We triangulated evidence from multiple sources where possible – for instance, if several studies reported improved exam performance from using a particular tool, we noted it as a consistent trend. We acknowledge potential limitations such as publication bias (successful interventions are more likely to be published) and the possibility that some relevant work might have been missed despite our search (for example, internal curricular reports or non-indexed workshop papers). Nonetheless, we are confident that the included studies represent a robust sample of the state of the art in Automata/FL education research. In the following section, we present the consolidated results, organized by the thematic questions of interest.

4 Results

4.1 Teaching Tools

The literature emphasizes using software tools to enhance teaching automata theory and formal languages, highlighting significant improvements in student engagement and understanding compared to traditional lecture-only approaches. Among the most prominent tools is **JFLAP**, which allows students to visualize automata, edit grammars, and simulate algorithms interactively. Studies consistently affirm JFLAP's effectiveness, indicating that students using this tool demonstrate improved comprehension and intuition of automata operations and higher classroom engagement (Cavalcante et al. [1], Rodger et al. [4]).

Beyond visualization, automated assessment tools like **Automata Tutor** offer significant pedagogical benefits. This web-based system enables students to solve automata problems, providing immediate targeted feedback, hints, and counterexamples for incorrect solutions (D'Antoni et al. [5]). Research shows students using Automata Tutor persist longer, engage more actively with the content, and ultimately perform better academically due to instant feedback and iterative practice. Furthermore, intelligent feedback extensions (e.g., Bezáková et al.'s [12] enhancement to JFLAP) further elevate learning outcomes by providing explicit counterexamples, aiding deeper conceptual understanding and reducing instructor grading workload. Integrated platforms, such as the interactive e-textbook implemented with the OpenDSA framework, combine visualization and automated assessment, resulting in higher exam scores and increased student satisfaction due to continuous, embedded practice opportunities (Mohammed et al. [11]). While the development effort for such resources is substantial, their long-term reuse potential makes them valuable community-driven educational resources.

Overall, effective integration of these teaching tools within thoughtfully designed pedagogy—emphasizing active learning, student interaction, and continuous feedback—is crucial for maximizing educational outcomes in automata and formallanguage courses. *(Detailed comparative features and impacts of these tools are summarized clearly in Table 1.)*

Table 1. *Detailed comparative features of teaching tools*

Tool	**Increased Student Engagement**	**Better Conceptual Understanding**	**Improved Student Performance**	**Reduced Grading Load (Instructor Benefit)**
JFLAP	**Yes:** Introduces a fun, "hands-on" element. Students preferred designing automata interactively over passive paper exercises.	**Yes:** Visual, interactive exploration simplifies complex concepts. Students experiment directly, making abstract topics concrete.	**Anecdotally:** Students report easier learning; formal studies mostly focus on engagement and usability rather than quantitative grades.	**Limited:** Initial version doesn't reduce grading load significantly. Later add-ons introduced auto-grading capabilities.
Automata Tutor	**Yes:** Interactive, game-like problem-solving enhances engagement. High persistence demonstrated by multiple attempts.	**Yes:** Immediate feedback and guided retries deepen understanding, reinforcing concepts through real-time correction of mistakes.	**Yes:** Demonstrated improvements through both self-reports and objective outcomes, notably better performance on assignments.	**Yes:** Significant reduction in grading load through automatic grading of assignments, greatly freeing instructor resources.
Interactive E-Textbook (OpenFLAP)	**Implied Yes:** Combines interactive demos and exercises within text, encouraging active participation over passive reading.	**Yes:** Visualization and immediate exercises significantly enhance comprehension by reinforcing theory through practice.	**Yes:** Empirical evidence confirms higher test scores and exam performance due to continuous interactive practice and immediate feedback.	**Yes:** Built-in auto-grading reduces instructor workload substantially, allowing easy monitoring and consistent assessment for large classes.

1. 4.2 Assessment Methods and Student Evaluation

Assessment in Automata/FL courses is critical to measure student learning and the effectiveness of teaching methods. Common methods include **traditional exams**, **pre/post-tests**, **project-based assessments**, and **automated grading tools**. A significant challenge noted in research is the **lack of standardized assessments**, making it difficult to compare outcomes across different studies or courses [14]. **Traditional exams and quizzes** remain widely used to evaluate students' understanding through theoretical and practical problems. However, a significant gap in standardized assessment practices is consistently noted in existing literature. Addressing this, we propose developing a comprehensive concept inventory and standardized benchmark problems, enabling consistent, comparative evaluation across diverse contexts and educational settings. Research often shows that new teaching methods lead to improved exam scores. For example, interactive textbooks or gamified approaches commonly yield modest but clear improvements compared to traditional methods [9,11]. However, since exam content varies greatly, direct comparisons between studies can be challenging.

Pre- and post-tests measure students' knowledge before and after a course or intervention, clearly showing learning gains. These tests provide reliable, quantitative evidence of how effectively specific teaching methods impact learning. Although valuable, these assessments are not common in regular classrooms due to their administrative complexity. **Project-based assessments** require students to apply theory practically (e.g., building automata or designing compilers). Projects enhance deeper learning and engagement but are resource-intensive for instructors to grade consistently. **Automated grading tools** (like Automata Tutor or JFLAP with automated feedback) offer immediate, personalized feedback. These tools enable repeated practice, helping students understand concepts quickly and effectively. They also greatly reduce instructor grading workload. The absence of a standardized concept inventory (a common, validated assessment tool) is a significant gap identified in the literature. Without standardized tests, it is difficult to consistently measure and compare learning outcomes across different educational interventions.

Table 2: Comparison of Automata/FL assessment methods by effectiveness, feedback speed, scalability, instructor effort, and typical use.

Assessment Method	**Effectiveness in Measuring Learning**	**Feedback Speed**	**Scalability**	**Instructor Effort**	**Typical Use Cases**
Traditional Exams (Written)	Broad coverage; reliable summative assessment.	**Delayed:** Weeks after completion.	**Medium:** Delivery scales; grading manual and challenging.	**High:** Intensive effort in creating and grading, especially open-ended questions.	Midterms, finals; comprehensive assessment.
Projects & Assignments	Deep application, demonstrates problem-solving and creativity.	**Delayed:** Feedback provided post-submission.	**Low–Medium:** Grading labor-intensive; partially automatable.	**Very High:** Extensive mentoring, reviewing, and evaluation required.	Capstone projects; applied learning and design activities.

Pre/Post-Tests (Concept Tests)	Good for tracking conceptual gains and improvements over time.	**Delayed:** Typically at course start/end; minimal	**High:** Easily administered in large groups; usually auto-graded.	**Low:** Easy to reuse, minimal grading effort.	Diagnostic tests, measuring teaching impact; often for research purposes.
		direct feedback.			
Automated Tools (Auto-Graded Exercises)	Effective for reinforcing concepts through immediate correction.	**Immediate:** Real-time feedback for every attempt.	**High:** Supports large classes effectively after initial setup.	**Medium initially, then Low:** Setup requires effort; afterward, auto-grading minimizes workload.	Online homework, labs, self-tests; formative practice tools (JFLAP, Automata Tutor).

Table 3. Multiple assessment approaches

Assessment Method	**Effectiveness**	**Feedback Speed**	**Instructor Effort**	**Scalability**	**Typical Use**
Traditional Exams	High	Delayed	High	Moderate	Summative evaluation
Projects	High	Delayed	Very High	Low/Medium	Applied practical assessments
Pre/Post-Tests	High	Delayed	Low	High	Measuring conceptual gains
Automated Tools	High	Immediate	Low (after setup)	High	Formative exercises

1. 4.3 Course Design and Delivery Approaches

The design of Automata and Formal Language (FL) courses greatly impacts student learning and engagement. Various instructional models are utilized:

- **Traditional Lecture-Based**: Standard method using instructor-led lectures, focusing on theory and homework assignments. Effective for structure and broad coverage but tends to cause passive learning and limited student engagement.
- **Active Learning Enhancements**: Adds interactive elements like quizzes, in- class problem-solving, and discussions to lectures. Improves engagement, addresses misconceptions early, and enhances students' grasp of concepts.
- **Flipped Classroom**: Students first engage with basic material independently

(through videos/readings), using class time for interactive problem-solving and clarification. Encourages active participation, although it requires motivated students and quality preparatory materials.

- **Project-Based/Constructivist Learning**: Students undertake practical projects (e.g., building a parser or automata simulator) applying theoreticalconcepts. Promotes deep, active learning and high motivation, though it can be challenging without strong support and clear guidelines.
- **Gamification/Game-Based Learning**: Integrates game elements or educational games to increase motivation and make abstract concepts more accessible. Generally enhances engagement but might not dramatically improve performance on traditional assessments**.**
- **Hybrid/Blended Approaches**: Combines several instructional methods (lectures, online modules, interactive activities) for an optimal learning experience. This model leverages the strengths of each method, typically yielding the highest student satisfaction and performance.

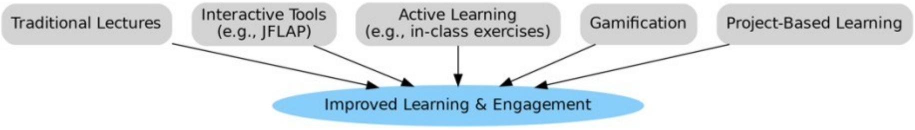

Traditional Lectures
Interactive Tools (e.g., JFLAP)
Active Learning (e.g., in-class exercises)
Gamification
Project-Based Learning
Improved Learning & Engagement

Fig. 1. A multi-faceted pedagogical model for Automata/FL courses.

In summary, a hybrid approach combining different methods (lectures for clarity, active learning for engagement, projects for depth, and gamification for motivation) is recommended as most effective in teaching Automata and Formal Languages.

5 Discussion and Conclusion

Our review highlights that no single instructional method fully addresses all challenges in teaching Automata and Formal Languages (FL). Instead, the best outcomes arise from combining multiple approaches: traditional lectures for foundational theory, interactive tools (e.g., JFLAP, Automata Tutor) for visualization and practice, gamification to enhance engagement, and projects for hands-on experience. This hybrid strategy not only boosts student performance but also makes abstract concepts more accessible and enjoyable. A significant barrier identified is the lack of standardized assessments. Establishing a shared concept inventory or a set of benchmark problems would greatly facilitate comparisons across studies, helping educators understand what teaching methods are most effective. Developing such standardized instruments—covering essential concepts like DFA/NFA equivalence, pumping lemma, and grammar-to-PDA conversions—would enable rigorous, comparative research and better-informed pedagogy. Moreover, there's a need for longitudinal studies examining long-term retention and transfer of knowledge. Currently, most studies assess immediate results rather than lasting impact. Future research should explore how different teaching strategies affect students' ability to apply automata concepts in advanced courses or professional contexts. Contextual and cultural considerations are also crucial. Methods effective in one region might require adjustments elsewhere, underscoring the importance of resource-sharing and faculty training. Creating accessible platforms to share teaching materials (e.g., exercises, projects, or automated tools) would help instructors worldwide adopt proven methods efficiently. Looking ahead, emerging technologies such as Virtual Reality (VR) and AI-driven intelligent tutors offer promising new ways to enrich learning experiences. However, these should be adopted thoughtfully, rigorously evaluated, and integrated carefully into existing curricula.

This review acknowledges several limitations. Firstly, potential publication bias may exist, as positive outcomes and successful interventions are more frequently reported in academic literature, potentially overlooking negative or neutral results. Secondly, our reliance on published literature in English may exclude relevant studies published in other languages or gray literature such as institutional reports, conference presentations, or unpublished theses, which could provide valuable insights. Furthermore, methodological constraints inherent in systematic literature reviews, such as the subjectivity in the selection of studies and interpretation of findings, are also recognized. While these limitations have been carefully considered, readers should interpret the findings within this context.

In conclusion, Automata/FL education is transitioning towards a more interactive, engaging, and evidence-based discipline. By blending pedagogical strategies, standardizing assessments, conducting long-term evaluations, facilitating resource-sharing, and cautiously embracing innovative technologies, educators can significantly improve student outcomes, engagement, and appreciation of Automata and Formal Language theory—fundamental pillars of computer science education.

References

1. Cavalcante, R., Cornell, T.F., Rodger, S.H.: A visual and interactive automata theory course with JFLAP 4.0. ACM SIGCSE Bulletin 36(1), 140–144 (2004).
2. Pillay, N.: Learning difficulties experienced by students in a course on formal languages and automata theory. ACM SIGCSE Bulletin 41(4), 48–52 (2009).
3. Chakraborty, P., Saxena, P.C., Katti, C.P.: Fifty years of automata simulation: a review. ACM Inroads 2(4), 59–70 (2011).
4. Rodger, S.H., Wiebe, E., Lee, K.M., Morgan, C., Omar, K., Su, J.: Increasing engagement

in automata theory with JFLAP. In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE ’09), pp. 403–407. ACM, New York (2009).

5. D’Antoni, L., Weaver, M., Weinert, A., Alur, R.: Automata Tutor and what we learned from building an online teaching tool. Bulletin of the EATCS 117, 144–162 (2015).
6. de Souza, G.S., Gomes, G.P.H., Correia, R.C.M., Garcia, R.E.: Teaching-learning methodology for formal languages and automata theory. In: Proceedings of the 2015 IEEE Frontiers in Education Conference (FIE), pp. 1–7. IEEE, El Paso (2015).
7. Naveed, M.S., Sarim, M.: Didactic strategy for learning theory of automata and formal languages. Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences 55(2), 55–67 (2018).
8. Neto, J., Terra, R.: LFApp: Um aplicativo móvel para o ensino de Linguagens Formais e Autômatos. In: Anais do XXIV Workshop sobre Educação em Computação, pp. 2196–2205 (2016).
9. Korte, L., Anderson, S., Pain, H., Good, J.: Learning by game-building: a novel approach to theoretical computer science education. In: Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, pp. 53–57 (2007).
10. Knobelsdorf, M., Kreitz, C., Börstler, J.: Teaching theoretical computer science using a cognitive apprenticeship approach. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE ’14), pp. 67–72. ACM, New York (2014).
11. Mohammed, M., Shaffer, C.A., Rodger, S.H.: Teaching formal languages with visualizations and auto-graded exercises. In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE ’21), pp. 569–575. ACM, New York (2021).
12. Bezáková, I., Fluet, K., Hemaspaandra, E., Miller, H., Narváez, D.E.: Effective succinct feedback for intro CS theory: A JFLAP extension. In: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education (SIGCSE ’22), pp. 976– 982. ACM, New York (2022).
13. Honda, F., Pires, F., Pessoa, M., Oliveira, E.: Automigos: Learning design para ludificação de Autômatos Finitos Determinísticos. In: Anais do XXXI Workshop sobre Educação em Computação, pp. 545–556 (2023).
14. Veiga da Silva, J., Cavalheiro, S.A., Foss, L.: Automata Theory in Computing Education: A Systematic Review. In: Proceedings of the XXXV Brazilian Symposium on Computers in Education (SBIE 2024). SBC, Brasília (2024).
15. Chesnevar, C.I., González, M.P., Maguitman, A.G.: Didactic strategies for promoting significant learning in formal languages and automata theory. ACM SIGCSE Bulletin 36(3), 7–11 (2004).
16. Castro-Schez, J.J., del Castillo, E., Hortolano, J., Rodriguez, A.: Designing and using software tools for educational purposes: FLAT, a case study. IEEE Transactions on Education 52(1), 66–74 (2009).
17. Habiballa, H., Kmet’, T.: Theoretical branches in teaching computer science. International Journal of Mathematical Education in Science and Technology 35(6), 829–841 (2004).
18. Dengel, A.: Seeking the treasures of theoretical computer science education: Towards educational virtual reality for the visualization of finite state machines. In: Proceedings of the 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), pp. 1107–1112. IEEE, Wollongong (2018).
19. Kitchenham, B.: Procedures for performing systematic reviews. Keele University Technical Report TR/SE-0401 (2004).

20. Linz, P., Rodger, S.H.: An Introduction to Formal Languages and Automata. Jones & Bartlett Learning, Burlington (2022).
21. Pettorossi, A.: Automata Theory and Formal Languages: Fundamental Notions, Theorems, and Techniques. Springer Nature, Cham (2022).
22. Morazán, M.T.: Programming-Based Formal Languages and Automata Theory. (Publisher and location/year not specified—**please add**). Mohammed, M., Shaffer, C.A., Rodger, S.H.: Teaching formal languages with visualizations and auto-graded exercises. In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), pp. 569–575. ACM, New York (2021).
23. Mohammed, M., Shaffer, C.A.: Teaching Formal Languages through Programmed Instruction. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education, vol. 1, pp. xx–xx (2024). (**Please add page numbers if available.**)
24. Mohammed, M.K.O.: Teaching Formal Languages through Visualizations, Machine Simulations, Auto-Graded Exercises, and Programmed Instruction. (Year, publisher, and location not specified—**please add**).
25. Becerra-Bonache, L., Barrachina, L., Esteller-Cucala, P., Julià, A., Martínez, D.: Formal Grammars and Languages. In: Mitkov, R. (ed.) The Oxford Handbook of Computational Linguistics, pp. 207–222. Oxford University Press, Oxford (2022).

29. Enhancing the Learning Experience Through Augmented Reality: Human Anatomy App Using AR

Elda Cina[1], Hanan Alomani[1], Rawan Almutaire[1,] Bibi Albaghli[1]

[1] University of Engineering & Technology, American University of the Middle East, Egaila, Kuwait
elda.cina@aum.edu.kw,52140@aum.edu.kw, 60066@aum.edu.kw, 54804@aum.edu.kw

Abstract. This paper presents an Augmented Reality (AR) based mobile application designed to enhance the teaching and learning of human anatomy through immersive, interactive 3D visualization. Traditional anatomy education methods, such as lectures, textbooks, and limited lab access—often fall short in addressing the spatial complexity of anatomical structures and accommodating diverse learning styles. To overcome these limitations, the proposed solution integrates Unity and Vuforia for AR model rendering, with Firebase for real-time data storage, user authentication, and quiz management. The application allows students to explore 3D anatomical models overlaid in real-world environments using mobile devices. Key features include role-based dashboards, image and plane tracking, zoom and rotation controls, integrated voice-over explanations, and personalized quizzes with instant feedback. The system architecture supports multiple user roles, students, instructors, and administrators, enabling differentiated access and functionality. Preliminary testing of the prototype demonstrated its functional reliability, interactivity, and potential to improve learner engagement. Visual outputs and usage scenarios confirm successful deployment of AR models, interactive assessments, and real-time user data handling. The study highlights how mobile AR can bridge gaps in accessibility, comprehension, and educational equity. This work contributes a scalable and pedagogically effective model for integrating AR into science and medical education.

Keywords: Augmented Reality (AR), Anatomy Education, 3D Visualization, Mobile Learning, Interactive Assessment.

1 Introduction

Across disciplines, traditional educational methodologies have long relied on lectures, textbooks, and structured laboratory sessions as primary modes of instruction. While these approaches have played a pivotal role in foundational learning, they often lack the interactivity and adaptability required to meet the diverse needs of today's learners. This is especially evident in science education, where subjects such as biology and human anatomy demand spatial reasoning, hands-on experimentation, and multi-sensory engagement. In such fields, static two-dimensional representations and textual explanations fail to fully capture the complexity and intricacies of biological systems [1]. Consequently, students are often left to memorize abstract visuals without developing a comprehensive understanding of the underlying structures and their interrelationships.

In anatomy education specifically, conventional learning materials including physical textbooks, plastic models, and infrequent access to real specimens, considerable limitations. University laboratories typically provide limited hours for student access, and the cost and ethical considerations associated with the use of real human or animal specimens further restrict opportunities for hands-on learning [2][3]. These constraints hinder repeated exposure, which is essential for mastery, and contribute to disparities in academic achievement. Students unable to afford personal models or supplemental resources may fall behind, exacerbating educational inequalities and reducing access to future academic and professional opportunities in medical and scientific domains.

Additionally, traditional methods are often ill-suited to address the needs of students with learning differences, such as dyslexia or visual processing disorders [4]. The one-size-fits-all approach of textbooks and lectures assumes a uniform learning style, neglecting the pedagogical importance of multisensory and experiential learning. As education evolves in the digital era, there is a growing recognition that technology must play a more significant role in creating personalized, engaging, and inclusive learning experiences.

This study explores the use of Augmented Reality (AR) as a pedagogically effective and technologically advanced alternative to conventional methods in anatomy education. AR allows for the projection of three-dimensional digital models into real-world environments, enabling learners to interact with anatomical structures in ways that are spatially intuitive and visually immersive. Through a mobile application developed using Unity and Vuforia, and integrated with Firebase for real-time data management, students can explore dynamic 3D models using their smartphone cameras. The application features tools for rotation, zooming, and auditory feedback, enhancing understanding through both visual and auditory channels. Role-based access control ensures differentiated experiences for students, instructors, and administrators, while interactive quizzes support formative assessment and feedback.

The aim of this study is to bridge the pedagogical gaps inherent in traditional anatomy instruction by introducing a scalable, accessible, and ethically sound learning environment that leverages AR technology. The research emphasizes the significance of promoting educational equity by enabling all students—regardless of income level or physical ability—to gain repeated, self-paced exposure to anatomical content. By fostering engagement, improving comprehension, and enhancing long-term knowledge retention, this AR-based application serves not only as a modern tool for anatomy education but also as a broader model for how emerging technologies can reshape science instruction in a more inclusive and sustainable direction.

2 Related Work

2.1 Technical Background Knowledge

The integration of Augmented Reality (AR) into anatomy education marks a significant shift in pedagogical strategy, particularly in healthcare and life sciences. Historically, anatomy has been taught using cadavers, physical models, and detailed illustrations. While effective in their time, these methods come with notable drawbacks. The use of cadavers, for example, is often limited due to high acquisition and maintenance costs, ethical concerns, and the logistical complexities of storage and preparation [5]. Moreover, physical models and textbook diagrams, though helpful, lack interactivity and fail to adequately represent the spatial complexity of anatomical systems.

AR technology addresses these limitations by superimposing digital, interactive 3D models onto the physical environment through mobile devices or headsets. This capability allows learners to engage with anatomical structures in a more intuitive and immersive way, promoting deeper spatial understanding [6]. Unlike passive 2D images, AR enables real-time manipulation of digital organs, vessels, and systems allowing users to zoom, rotate, and explore models from multiple angles [7]. This hands-on engagement is particularly valuable in medical education, where the ability to understand and visualize complex anatomical relationships is critical.

The roots of digital anatomy education can be traced back to the 1980s, with the emergence of 3D computer modeling that allowed students to view virtual representations of human anatomy [8]. By the early 2000s, advancements in AR technology began to capture the attention of educators and researchers interested in enhancing healthcare training. Since then, a growing body of literature has supported the use of AR as a tool for visualizing layered anatomical structures and fostering active learning. Research indicates that AR-enhanced learning improves students' ability to comprehend spatial relationships, retain anatomical knowledge, and apply their understanding in clinical scenarios [9].

One of the key advantages of AR is its capacity to extend learning beyond the physical confines of a laboratory. Mobile AR applications can be used on smartphones and tablets, making anatomical exploration accessible anytime and anywhere. This flexibility supports independent study, repeat practice, and greater inclusivity for learners with limited access to traditional lab resources [11]. Furthermore, AR aligns well with current trends in digital and remote education, offering a robust platform for e-learning in both formal and informal settings.

However, the implementation of AR in educational settings is not without challenges. High-end AR devices such as the Microsoft HoloLens are often cost-prohibitive for many institutions, especially those in under-resourced regions [12]. In addition, technical limitations including limited resolution, lower visual fidelity, and a lack of tactile feedback can diminish the realism and accuracy required for detailed anatomical study. The steep learning curve associated with AR platforms also poses a barrier for both students and instructors unfamiliar with the technology [13].

Despite these limitations, the rapid development of AR continues to enhance its viability as an instructional tool. As hardware becomes more affordable and software more sophisticated, AR is poised to become an integral component of medical and anatomy education. Continued research and long-term studies are essential to assess the effectiveness of AR in promoting knowledge retention, improving learner outcomes, and meeting curricular standards in anatomy education.

2.2 Literature Review and Related Projects

Over the past decade, the incorporation of Augmented Reality (AR) into anatomy education has gained substantial momentum, with numerous studies and applications demonstrating its potential to improve comprehension, engagement, and retention among learners. Several AR tools have been developed to enhance the visualization of anatomical structures and provide interactive learning environments that surpass the limitations of traditional methods.

Applications such as Visible Body, Complete Anatomy, and Human Anatomy Atlas are widely recognized in academic contexts for offering high-fidelity 3D anatomical models [14]. These platforms allow students to manipulate virtual organs, zoom in on specific components, and view systems from multiple perspectives. In doing so, they provide a more intuitive and spatially accurate understanding of the human body, which is difficult to achieve through textbooks or static models alone. The integration of dynamic features, including layered anatomical structures, labeling, and motion simulations supports active learning and bridges the gap between theoretical knowledge and practical understanding.

Devices like the Microsoft HoloLens have further expanded AR’s role in education by introducing mixed-reality environments. These headsets enable hands-free interaction with virtual anatomical elements overlaid on real-world surroundings, facilitating immersive, collaborative learning experiences [15]. Students can perform virtual dissections, observe organs in situ, and explore physiological functions in real time. This immersive quality not only boosts engagement but also supports the development of situational awareness, a critical skill in clinical training.

Studies have consistently shown that AR-enhanced learning environments improve students' motivation, attention span, and confidence. The ability to revisit complex topics at one's own pace and explore concepts through multiple modalities supports both comprehension and long-term retention. AR's portability via smartphones and tablets also democratizes access to quality anatomy instruction, making it feasible for learners to study outside of institutional settings, a particularly valuable advantage for distance education or under-resourced schools.

Despite its many benefits, literature also acknowledges several limitations and ongoing challenges associated with AR in anatomy education. One significant concern is the current lack of visual resolution in many AR platforms, which can hinder the accurate depiction of fine anatomical details such as nerves, capillaries, or microstructures. Without high-resolution rendering, the effectiveness of AR for advanced anatomical study may be compromised. Moreover, the absence of tactile feedback in most AR systems prevents learners from developing the manual skills and haptic memory that physical dissections offer [16].

There is also a need for more comprehensive, longitudinal studies to evaluate AR's impact on long-term knowledge retention and skill acquisition. While many pilot studies report positive short-term outcomes, data on sustained learning and practical application remain limited. Additionally, the financial cost and technical complexity of implementing AR tools, particularly those requiring specialized hardware, pose challenges for widespread adoption.

Nonetheless, as AR technology continues to evolve, these limitations are gradually being addressed. The convergence of AR with artificial intelligence, machine learning, and haptic feedback holds promise for future educational innovations. Current research continues to support AR's potential as a transformative force in anatomy education, positioning it not only as a supplement to traditional methods but also as a foundation for the next generation of immersive and inclusive medical instructions.

3 Proposed Solution

The application of Augmented Reality (AR) in anatomy education presents an innovative response to the limitations of conventional learning methods. By merging interactive technologies with pedagogical principles, AR offers a dynamic and accessible framework for visualizing complex anatomical structures. This study proposes a mobile-based AR solution designed to provide students with immersive, personalized, and repeatable learning experiences.

The solution incorporates multiple advanced features to ensure both usability and educational value:

3D Visualization and Manipulation

Central to the application is the ability to visualize anatomical structures in three-dimensional space. Users can interact with models by rotating, zooming in and out, and exploring structures from multiple angles. This spatial interaction enhances understanding of anatomical relationships that are often lost in static 2D representations.

Embedded Knowledge Database:

The AR models are supplemented with an integrated database containing detailed textual explanations and contextual information. This allows students to access layered educational content while exploring the models, supporting multimodal learning strategies.

Image Recognition Technology:

The app utilizes advanced image tracking capabilities to detect specific markers or surfaces in the physical environment. Once recognized, these cues activate the corresponding 3D anatomical content, creating a seamless blend between physical space and digital overlay.

Voice-Over Integration:

To support auditory learners and improve comprehension, the application includes voice-over functionality. Each anatomical model is linked to audio explanations that describe functions and features, allowing students to engage with the content in a more inclusive manner.

Plane Detection for Real-World Interaction:

Plane detection allows the application to recognize flat surfaces (e.g., a table or floor), enabling the projection of anatomical models into real-world contexts. This functionality increases realism and encourages interactive exploration.

Interactive Assessment Tools:

The application features customizable quizzes that can be managed by instructors. After completing a quiz, students receive immediate feedback, including scores and correct answers. This formative assessment mechanism supports self-evaluation and reinforces learning through active recall.

By combining these elements, the proposed solution addresses critical gaps in traditional anatomy education—namely, limited accessibility, lack of interactivity, and insufficient accommodation of varied learning needs. It provides a scalable, user-centered educational platform that is well-suited to the evolving landscape of digital and remote learning. Furthermore, the design supports inclusivity, sustainability, and pedagogical rigor, aligning with the broader objectives of improving engagement, retention, and equity in scientific education.

1.1 Conceptual Design

The conceptual design of the Human Anatomy AR application focuses on user-centered interactivity, system modularity, and real-time educational feedback. The application architecture integrates front-end augmented reality components with a robust cloud-based back end to support visualization, assessment, and user management functionalities.

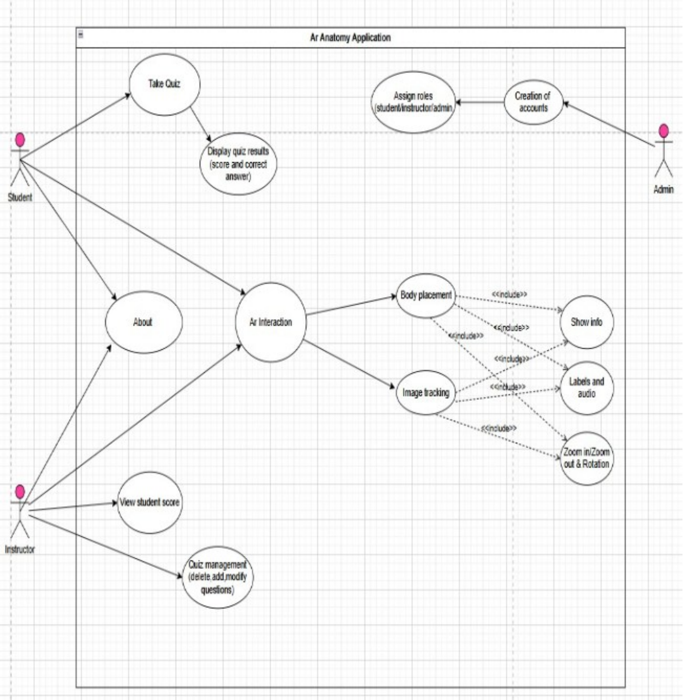


Fig. 14: Functional Modeling

A key design feature is the direct model selection interface, which allows users to access anatomical models through clearly labeled buttons rather than sequential navigation. This improves usability and minimizes cognitive load, enabling learners to efficiently explore specific anatomical structures as needed. Each model is accompanied by embedded data labels and audio explanations, allowing students to click on a label and hear an auditory description of the corresponding anatomical part. This dual-modality approach supports both visual and auditory learning preferences.

Within the AR view, users can manipulate models through intuitive slides for rotation and zooming. These functions allow learners to observe structures from multiple perspectives and at various levels of detail, fostering spatial awareness and enhancing comprehension of anatomical relationships.

To facilitate access control and tailored learning experiences, the application employs Firebase Authentication and Firestore. Users are assigned roles, student, instructor, or administrator, each of which grants access to different functionalities. Instructors can upload, modify, and review quiz questions, while students interact with learning materials and complete assessments. Administrators are responsible for managing user access and assigning roles.

Two distinct dashboards are integrated into the system. The Instructor Dashboard allows faculty members to monitor student progress, manage quiz content, and analyze learning outcomes. The Admin Dashboard provides tools for user management and role assignment, streamlining the oversight of the educational environment. Both dashboards are built with modular flexibility, enabling future expansion or integration of additional features.

This design emphasizes real-time interactivity, personalized learning, and administrative scalability. By combining AR-based exploration with auditory support, dynamic quizzes, and role-based access control, the application offers a comprehensive platform that aligns with pedagogical best practices and contemporary digital learning needs.

4 Implementation

4.1 Development Environment

The implementation of the Human Anatomy AR application was carried out using a combination of industry-standard development tools and platforms. The application was built in Unity, a cross-platform game engine widely used for interactive 3D content, with scripting handled in C#. For augmented reality capabilities, the Vuforia Engine was integrated into the Unity environment to enable both image target tracking and ground plane detection. Backend services were managed using Firebase, which provided cloud-based authentication, real-time data storage, and role-based access management.

Development was performed on a MacBook Pro equipped with the Apple M4 chip, providing sufficient processing power and graphics performance for building and testing complex AR models. Application testing and deployment were conducted on an Android-based mobile device (Samsung Galaxy S10 Lite), selected for its compatibility with ARCore and robust performance for rendering real-time 3D environments.

This configuration allowed for efficient testing of both the AR experience and user interface across various roles (students, instructors, administrators). The modular setup and cloud-based architecture ensured smooth synchronization of user data, quizzes, and anatomical content during the development process.

4.2 System Architecture

The system architecture follows a modular and layered design to ensure maintainability, scalability, and high performance. The architecture consists of three primary layers: the front-end AR interface, the application logic layer, and the backend database.

- **Front-End Layer**: Built using Unity and Vuforia, the front-end layer handles rendering of 3D anatomical models and user interaction. Image target tracking enables the display of relevant anatomical models based on scanned markers, while ground plane detection allows placement of models into physical space without requiring markers.
- **Application Logic Layer**: C# scripts developed in Unity handle all functional logic, including model interaction (e.g., rotation, zooming), role-based navigation, quiz functionality, and communication with Firebase. This logic ensures that users see only the interface and data appropriate to their role—student, instructor, or administrator.
- **Backend Layer**: Firebase Authentication and Firestore are used for real-time data handling. Authentication secures user access and assigns roles, while Firestore stores quiz data, user scores, and role-based permissions. Changes made to quizzes or user records are immediately synchronized across all connected devices without requiring app updates, thanks to the real-time capabilities of Firebase.

This architecture enables dynamic, role-specific experiences while supporting secure data handling and easy extensibility. Future features such as advanced analytics, quiz categorization, or expanded AR interactions can be integrated without disrupting the existing framework.

4.3 Algorithm and Logic Implementation

The core logic of the application is implemented through a series of modular C# scripts within the Unity environment. These scripts manage user authentication, AR content activation, quiz delivery, and system feedback mechanisms.

Upon login, Firebase Authentication verifies user credentials and retrieves the corresponding user role. Based on this role, the system dynamically loads the appropriate dashboard interface—whether for students, instructors, or administrators. User interactions, such as selecting a model or submitting a quiz, are monitored and recorded in Firestore for real-time synchronization.

For AR interaction, model activation is triggered through Vuforia's image recognition or plane detection. Users can manipulate models through touch gestures linked to control sliders, allowing zooming and rotation. Each model is linked to interactive labels, which provide both text and audio explanations to enrich the learning experience.

Quiz logic supports customization by instructors, who can add, edit, or remove questions directly from the dashboard. Student performance is recorded and analyzed through the system's feedback mechanism, which generates scores and displays correct answers upon quiz completion.

The implementation prioritizes responsiveness, interactivity, and data integrity, ensuring a seamless and engaging learning environment for all users.

5 Experimental Results and Evaluation

To assess the functionality and usability of the Human Anatomy AR application, we conducted a series of validation tests focusing on key system features, including AR model projection, user interaction, and role-based access performance. The evaluation aimed to determine whether the system delivered on its intended outcomes: improved anatomical visualization, accessible learning, and interactive self-assessment.

The application was tested on multiple Android devices with ARCore support, and its functionality was verified across different user roles. Figure 2 shows the main menu interface, while Figures 3 4 illustrate successful AR model deployment using both image tracking and ground plane detection Figure 5 shows a sample form the quiz interface. Users were able to manipulate 3D anatomical models with responsive zoom and rotation features, and auditory explanations were triggered reliably through labeled icons.

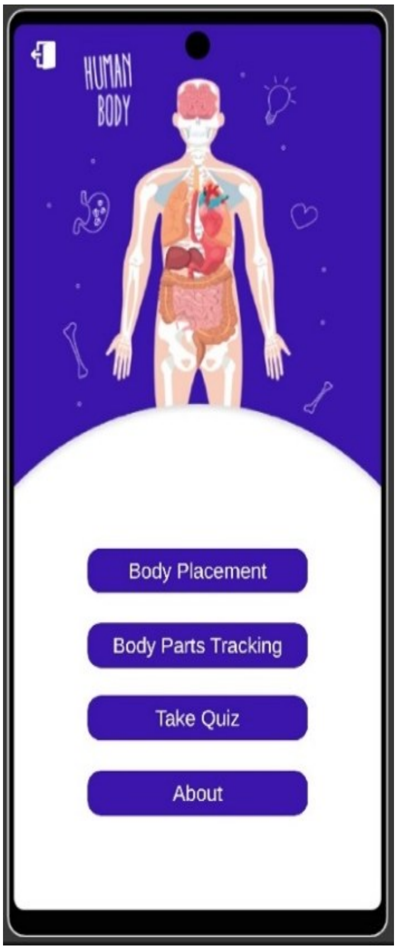


Fig. 2 Main Menu

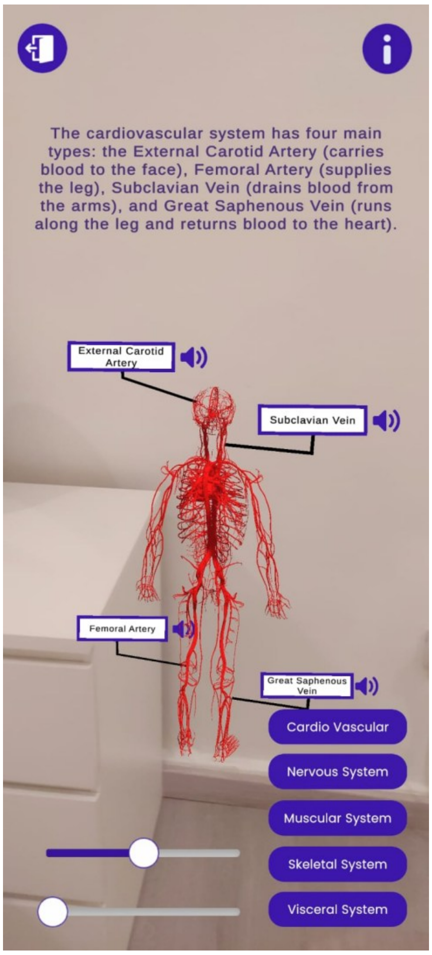
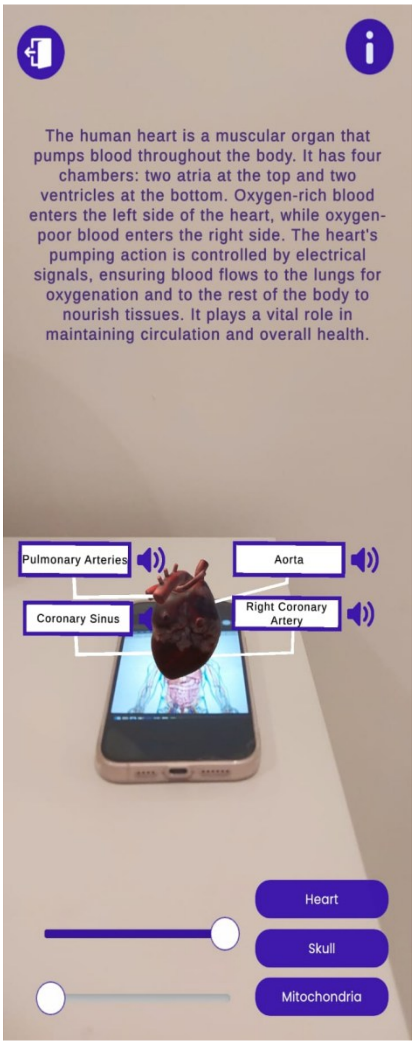
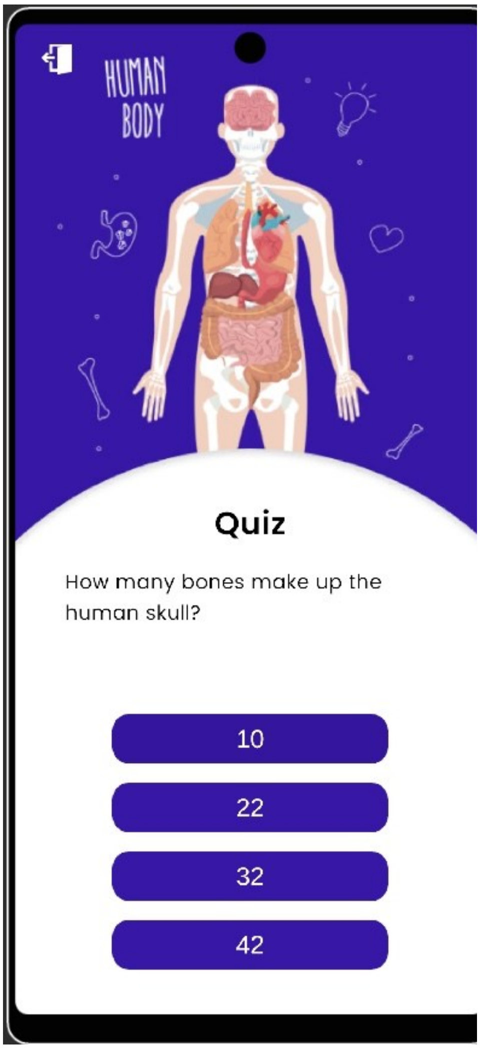

The cardiovascular system has four main types: the External Carotid Artery (carries blood to the face), Femoral Artery (supplies the leg), Subclavian Vein (drains blood from the arms), and Great Saphenous Vein (runs along the leg and returns blood to the heart).

Fig 3. AR camera ground detection.



The human heart is a muscular organ that pumps blood throughout the body. It has four chambers: two atria at the top and two ventricles at the bottom. Oxygen-rich blood enters the left side of the heart, while oxygen-poor blood enters the right side. The heart's pumping action is controlled by electrical signals, ensuring blood flows to the lungs for oxygenation and to the rest of the body to nourish tissues. It plays a vital role in maintaining circulation and overall health.

Fig. 4AR camera image tracking



How many bones make up the human skull?

Fig. 5 Student quiz sample

To simulate the user experience, a test group of students interacted with the application under instructor guidance. The role-based login system functioned as intended, granting each user appropriate access, students completed quizzes, instructors viewed performance summaries, and administrators managed user accounts. The quiz system provided immediate feedback, correctly scoring responses and displaying the correct answers at the end of each session.

A summary of feature validation is shown in table 1:

Table 8. Table captions should be placed above the tables.

Feature Tested	Outcome	Notes
AR model projection (image)	Successful	Accurate alignment and stable overlay
AR model projection (plane)	Successful	Models anchored to detected surfaces accurately
Voice-over descriptions	Functional	Clear audio and appropriate label binding
Role-based login	Functional	Role restrictions and dashboard access enforced
Quiz system (input/output)	Successful	Score tracking and result feedback worked as expected
Real-time data sync (Firebase)	Real-time synchronizatio n	Quiz updates and user records appeared instantly

To contextualize the capabilities and pedagogical contributions of the Human Anatomy AR application, a comparative analysis was conducted against two prominent AR-based anatomy platforms currently available in the market: *Complete Anatomy* by 3D4Medical and *Human Anatomy Atlas* by Visible Body. The comparison focuses on key functional, technical, and instructional features such as AR integration, user interactivity, assessment tools, accessibility, and cost. This evaluation highlights the strengths and limitations of each application while underscoring the unique contributions of the proposed system, particularly in terms of educational flexibility, role-based access, integrated quizzes, and cost-effectiveness for academic institutions. The results of this comparison are summarized in Table 2.

Table 2. Comparative Analysis of AR-Based Anatomy Applications

Feature / Criterion	Human Anatomy AR (This Study)	Complete Anatomy (3D4Medical)	Human Anatomy Atlas (Visible Body)
Platform	Android (tested), Unity + Vuforia	iOS, Android, Windows, Mac	iOS, Android, Windows, Mac
AR Technology	Image and plane tracking via Vuforia	Limited AR (markerless; requires newer iPads/iPhones)	AR via marker-based and markerless (iOS/Android)
3D Model Interactivity	Zoom, rotate, explore individual components	Highly detailed manipulation, slicing, labeling	Interactive 3D models with zoom and rotation

Voice-Over Support	Yes (custom audio explanations per label)	No integrated voice-over	No integrated voice-over
Quizzes / Assessment Tools	Built-in quizzes with instant feedback (Firebase)	Limited; quiz packs available as paid content	Includes multiple-choice quizzes and flashcards
User Roles and Access Control	Multi-role (student, instructor, admin)	Single-user model; limited admin controls	Primarily individual user access
Content Personalization	Instructor-customizable content and quizzes	Static content; some user-created features	Predefined learning paths; limited customization
Offline Functionality	Partially (AR model requires local marker; Firebase online)	Yes (after downloading content)	Yes (after downloading content)
Cost	Free prototype (open-source/academic use)	Paid subscription (~$35–$50/year)	Paid app (~$25–$35); institutional licenses available
Language / Accessibility Options	Planned multilingual voice/text support	English only (as of latest version)	English with limited multilingual UI support
Real-time data sync (Firebase)	Real-time synchronization	Quiz updates and user records appeared instantly	

As shown in the comparative analysis, the Human Anatomy AR application demonstrates unique strengths that position it as a more flexible, accessible, and educator-oriented tool compared to existing commercial alternatives. While platforms like *Complete Anatomy* and *Human Anatomy Atlas* offer advanced anatomical models and broad platform support, they are primarily designed for individual users and often require paid subscriptions, limiting institutional scalability and customization. In contrast, the proposed application integrates role-based dashboards, customizable quizzes, and voice-over support, all features that enhance inclusivity, active learning, and instructor control. Additionally, its use of Firebase for real-time data management allows for adaptive feedback and collaborative oversight, which are not fully supported in competing systems. These distinctions underline the application's value as an academic tool tailored for educational institutions seeking affordable and pedagogically effective AR-based solutions, particularly in resource-constrained environments.

6 Conclusions and Future Work

This paper introduced a mobile Augmented Reality (AR) application designed to enhance anatomy education by addressing the limitations of traditional pedagogical approaches. Through the integration of Unity, Vuforia, and Firebase, the system enables students to visualize and interact with 3D anatomical models in real-world environments using their smartphones. The application also incorporates voice-over explanations, customizable quizzes, and role-based access dashboards to support diverse learning needs and instructional workflows.

Experimental validation confirmed the functional reliability of the system across its key features, including image and plane tracking, user role separation, model manipulation, and real-time quiz interaction. The application's ability to deliver a flexible, immersive, and accessible learning experience makes it a promising tool for academic environments where physical resources are limited or where inclusive digital learning strategies are prioritized. In particular, the system offers advantages over existing commercial solutions by focusing on educational equity, instructor control, and cost-effectiveness.

However, several limitations remain. The current prototype lacks haptic feedback, comprehensive usability studies, and high-resolution models necessary for advanced medical training. Future iterations will aim to enhance model fidelity, expand the range of anatomical systems covered, and introduce adaptive learning features powered by AI. Additional work will include conducting formal user studies, integrating multilingual support, and enabling collaborative AR sessions.

By demonstrating a viable, scalable approach to AR-based anatomy instruction, this study contributes to the growing body of research advocating for immersive technologies in education. The findings support the potential of AR not only to supplement but to transform science and medical curricula through engaging, personalized, and accessible digital learning experiences.

References

1. S. Narayanan and R. Ramakrishnan, "Strategies to Effectively Utilize Images in Anatomical Teaching and Assessment," Medical Science Educator, vol. 34, no. 3, pp. 671–678, Mar. 2024, doi: https://doi.org/10.1007/s40670-024-02030-y.
2. Y. M. Baptiste and S. Abramovich, "Community college student perceptions of digital anatomy models as a curricular resource," Anatomical Sciences Education, vol. 17, no. 9, pp. 1731–1748, Oct. 2024, doi: https://doi.org/10.1002/ase.2523.

3. B. Kramer and J. T. Soley, “Medical student perception of problem topics in anatomy,” *East African Medical Journal*, vol. 79, no. 8, Aug. 2002, doi: https://doi.org/10.4314/eamj.v79i8.8826.
4. Zharko Bliznakov, “Inclusive Education: Adapting Assessments for Dyslexic Learners in the 21st Century Classroom,” ResearchGate, Feb. 2024, doi: https://doi.org/10.13140/RG.2.2.31420.45449.
5. A. M. Șișu et al., “Blending Tradition and Innovation: Student Opinions on Modern Anatomy Education,” Education Sciences, vol. 14, no. 11, p. 1150, Oct. 2024, doi: https://doi.org/10.3390/educsci14111150.
6. A. M. Al-Ansi, M. Jaboob, Askar Garad, and A. Al-Ansi, “Analyzing augmented reality (AR) and virtual reality (VR) recent development in education,” Social Sciences & Humanities Open, vol. 8, no. 1, pp. 100532–100532, Jan. 2023, doi: https://doi.org/10.1016/j.ssaho.2023.100532.
7. D. Deka, “Applications of virtual reality in the field of anatomy: a review article,” International Journal of Research in Medical Sciences, vol. 12, no. 8, pp. 3124–3128, Jul. 2024, doi: https://doi.org/10.18203/2320-6012.ijrms20242255.
8. K. Hou et al., “The role of 3D printing in anatomy education and surgical training: A narrative review,” MedEdPublish, vol. 6, pp. 92–92, Jun. 2017, doi: https://doi.org/10.15694/mep.2017.000092.
9. P. Dhar, T. Rocks, R. M. Samarasinghe, G. Stephenson, and C. Smith, “Augmented reality in medical education: students’ experiences and learning outcomes,” Medical Education Online, vol. 26, no. 1, Jan. 2021, doi: https://doi.org/10.1080/10872981.2021.1953953.
10. F. Bork, A. Lehner, U. Eck, Nassir Navab, Jens Waschke, and D. Kugelmann, “The Effectiveness of Collaborative Augmented Reality in Gross Anatomy Teaching: A Quantitative and Qualitative Pilot Study,” Anatomical Sciences Education, vol. 14, no. 5, pp. 590–604, Sep. 2020, doi: https://doi.org/10.1002/ase.2016.
11. N. Sinou, N. Sinou, and D. Filippou, “Virtual Reality and Augmented Reality in Anatomy Education During COVID-19 Pandemic,” Cureus, Feb. 2023, doi: https://doi.org/10.7759/cureus.35170.
12. Dimitrios Chytas et al., “The role of augmented reality in Anatomical education: An overview,” Annals of Anatomy - Anatomischer Anzeiger, vol. 229, pp. 151463–151463, Jan. 2020, doi: https://doi.org/10.1016/j.aanat.2020.151463.
13. K. Lee, “Augmented Reality in Education and Training,” TechTrends, vol. 56, no. 2, pp. 13–21, Feb. 2012, doi: https://doi.org/10.1007/s11528-012-0559-3.
14. M. H. Kurniawan, Suharjito, Diana, and G. Witjaksono, “Human Anatomy Learning Systems Using Augmented Reality on Mobile Application,” Procedia Computer Science, vol. 135, pp. 80–88, 2018, doi: https://doi.org/10.1016/j.procs.2018.08.152.
15. B. B. Boyanovsky, M. Belghasem, B. A. White, and S. Kadavakollu, “Incorporating Augmented Reality Into Anatomy Education in a Contemporary Medical School Curriculum,” Cureus, Apr. 2024, doi: https://doi.org/10.7759/cureus.57443.
16. Sreenivasulu Reddy Mogali et al., “Evaluation by medical students of the educational value of multi-material and multi-colored three-dimensional printed models of the upper limb for anatomical education,” Anatomical Sciences Education, vol. 11, no. 1, pp. 54–64, May 2017, doi: https://doi.org/10.1002/ase.1703.

30. Self-Adaptive Security Level Correction for Dynamic Access Control Using Fuzzy Time Series

Anita Xhemali [1], Elma Zanaj [2], Gledis Basha[3] and Lorena Balliu[4]

[1] Polytechnical University of Tirana, Faculty of Electrical Engineering, Boulevard "Dëshmorët e Kombit", Square "Mother Teresa", 4
[2,3,4] Polytechnical University of Tirana, Faculty of Information Technology, Boulevard "Dëshmorët e Kombit", Square "Mother Teresa", 4
anita.xhemali@fti.edu.al , ezanaj@fti.edu.al, gledis.basha@fti.edu.al,
lorena.balliu@fti.edu.al

Abstract. Ensuring secure and adaptive access to medical data is essential in intelligent telemedicine monitoring systems. This study introduces a self-adaptive security level correction framework for dynamic access control using Fuzzy Time Series (FTS). The proposed approach integrates two interconnected Fuzzy Inference System (FIS) controllers to enhance security decision-making based on user roles, access request patterns, and historical security trends. Real and generative IoT data simulations validate the system's ability to adaptively adjust security levels in response to evolving risk factors. By leveraging fuzzy time series for trend analysis, the proposed model enhances proactive security decision-making, striking a balance between accessibility and risk mitigation in assisted living environments. This adaptive approach strengthens access control mechanisms in telemedicine, ensuring security policies remain responsive to dynamic user behaviors and emerging threats.

Keywords: IoT, Fuzzy Time Series, Security Level Access Control, Feed Back correction mechanism.

1 Introduction

Access control plays a crucial role in protecting data within IoT systems by limiting access to authorized users only. Although IoT facilitates communication between devices without the need for human input, it also introduces serious security concerns, particularly in sensitive environments like healthcare. To address these issues, researchers have explored the use of fuzzy logic and blockchain technologies to strengthen privacy measures and improve system responsiveness in healthcare-focused IoT applications [1], [2]. However, risk factors are often treated with equal weight, and expert judgments can introduce subjectivity. To address this, some IoT access control systems use fuzzy sets to evaluate risk more adaptively, incorporating indicators such as vulnerabilities and anomalies [3] , while learning user roles and daily activities for security models. Our literature review highlights adaptive risk-based access control models that use real-time data to assess security risks, considering user attributes, action severity and risk history. The optimized FIS parameters allow the security system to dynamically adjust based on real-time user access time and historical patterns [4]. A notable gap is the lack of trend analysis to better understand risk patterns in system already implemented [5] , [6]. At our knowledge's, some studies emphasize incorporating user roles to improve risk analysis, training, and compliance with GDPR (General Data Regulation) regulations [7], [8], [9]. Network risk assessment often struggles with the absence of unified evaluation criteria. Fuzzy logic, most of the time, offers a flexible way to address this challenge by handling uncertainty and combining diverse risk factors [10]. Cybersecurity involves diverse threats, and fuzzy optimization helps determine input parameters for better management of congestion, warnings, and actions [11], [12]. There are also studies that are fuzzy-based risk assessment method only for IoT providing a foundation for expert decision support systems [13], [14].

This study introduces a self-adaptive security framework for dynamic access control, powered by Fuzzy Time Series (FTS). The system integrates two Fuzzy Inference Systems (FIS) to make smarter security decisions by analyzing user roles, access behavior and historical trends. By leveraging actual access history and past security levels, it continuously self-corrects and adjusts security levels in real time ensuring the system responds effectively to evolving risks and usage patterns. Simulations with real data (Kaggle Dataset), [15], and generative data demonstrate its ability to balance accessibility with risk mitigation and enhance proactive security decision-making in assisted living environments. This ability is surely tested with real time test code and for more accuracy of the predicted and corrected security levels, we used and adapt for our parameters the Access Level Validation and Error Analysis Module (ALEAM). This module compares predicted values against ground truth security levels generated using contextual rules (role, time, location), computes absolute errors, and calculates improvement rates based on correction strategies**.**

The paper is organized as follows: Section 2 presents the overall framework, highlighting the dual-model approach we adopted for access prediction and security level assignment. Section 3 details the simulation environment, including both real and synthetically generated access data, along with visualizations created in MATLAB. It also includes real-time testing and a comprehensive error analysis comparing predicted and corrected security levels against real security levels from real dataset. Section 4 concludes the study and outlines key directions for future work.

2 Methodology

Our framework uses a two-part approach that combines FIS (Fuzzy Inference System) and FTS (Fuzzy Time Series) to adjust security levels in real time. The first part that is FIS system considers things like the user's role, when and where they're accessing the system, and past access history patterns.

Meanwhile, FTS uses historical data to predict and adjust security levels. It considers past security trends, previous access predictions, and the current access time to ensure the correct security level is applied, Figure 1.

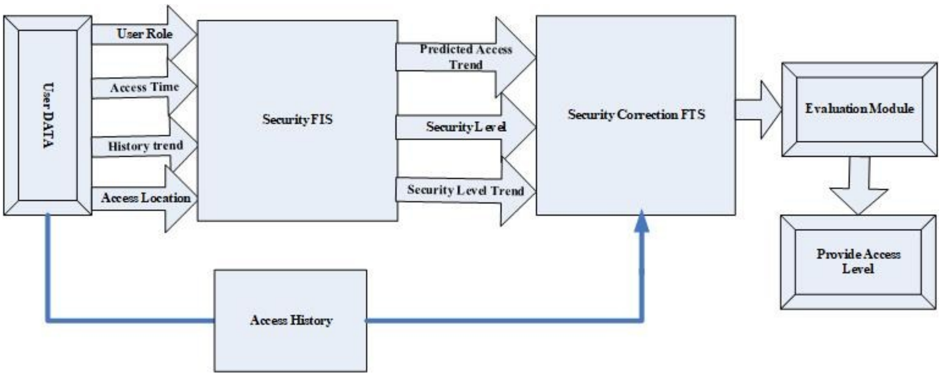


Fig. 9. Workflow of corrected security levels.

This framework is validated on two datasets. The first is a Generative Dataset with simulated roles and time patterns, while the second is a Real-World Dataset from Kaggle [15], containing actual healthcare access logs, roles, and security levels. Our module consists of three core components:

2. Security FIS for initial access level prediction based on user role, access time, history trend of requested accessing time and access location, defined as outside area or inside area of service.
3. The Security Correction FTS model is developed to correct the accuracy of access security predictions initially produced by the Security FIS. It has four inputs, and two of them require preprocessing: The Predicted Access Trend is generated using a sliding window over the last five access predictions. The Security Level Trend is calculated by analyzing variations in previously predicted security levels. The Predicted Security Level is the direct output of the Security FIS, which assigns a security level based on current access conditions. The User Access History is derived from past access logs and is processed to identify recurring behavioral patterns. While the Security FIS processes raw user data (user role, access time, location, etc.) to forecast initial

security levels and access trends, these outputs undergo an additional preprocessing phase to extract trend patterns. This dual structure enables the system to adapt to real time changes and maintain high reliability in user access control, especially in sensitive domains such as healthcare. The Security Correction FTS model is implemented using a simplified fuzzy time series approach, where time variant access trends are fuzzified, and a dominant pattern is used instead of traditional FLRGs (Fuzzy Logical Relationship Groups). This ensures computational efficiency while retaining reasoning over fuzzy inputs, prioritizing simplicity and interpretability.

4. Evaluation Tester Module for testing the effectiveness of the correction mechanism tried over real time security level from downloaded database. This module is based on calculating PSL (Prior Security Level) and CSL (Corrected Security Level) errors in the following formulas:

$$PSL\ Error = |security\ levels - true\ security\ levels| \qquad (1)$$

$$CSL\ error = |corrected\ security\ levels - true\ security\ levels| \qquad (2)$$

These formulas are combined with long-term Access History from user dataset, allowing the FTS to detect mismatches in access patterns. When it does find anomalies based on requested security levels it adjusts the security levels dynamically. The improvement rate is calculated as follows:

$$Improvement\ rate = 100x\left(\frac{corrected\ levels - prior\ levels}{total\ levels}\right)^2 \qquad (3)$$

Both datasets are preprocessed to extract fuzzy variables and labeled security decisions.

3 Simulation and Visualization

The simulations, set on two distinct datasets, evaluate the effectiveness of our fuzzy correction model. The first database is a generative dataset designed to mimic access control scenarios in assisted living environments, while the second is a real-world dataset containing access log records from an actual IoT system. Table1 reflects an example of data structure. The negative numbers in the Access History Trend column represent a decline in access frequency or activity over time.

Table 1. Examples of Data Inputs for Security FIS.

User Id	User Role	Access Time	Access History	Access History Trend	Access Location
3484	'Nurse'	**10**	**43**	**0**	'Inside Residential Area'
7117	'Lab Analyst'	**2**	**31**	**12**	'Inside Residential Area'
6895	'Pharmacist'	**5**	**17**	**-14**	'Outside Residential Area'
2070	'Technician'	**3**	**20**	**-6**	'Inside Residential Area'
5485	'Doctor'	**19**	**3**	**-17**	'Inside Residential Area'
3295	'Admin'	**23**	**2**	**-19**	'Inside Residential Area'
3586	'Nurse'	**10**	**40**	**0**	'Inside Residential Area'
7120	'Lab Analyst'	**6**	**31**	**12**	'Inside Residential Area'
6875	'Pharmacist'	**5**	**17**	**-2**	'Outside Residential Area'

To ensure consistency across different input features and make the models comparable, all data was normalized prior to simulation because the real dataset had only 5 levels of users. For both datasets, we set a maximum number of 50 requests per day, which reflects a realistic number typically observed in assisted living healthcare on more loaded days, though the system is capable of scaling to accommodate higher access volumes when necessary. Figure 2 illustrates the output of the Security that predicts restricted security levels and access based on fuzzified access history, access location, and historical access trend patterns.

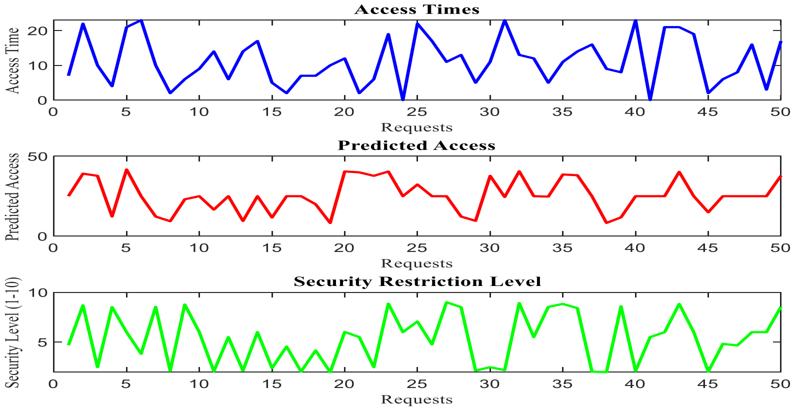


Fig. 2. Security levels and Access prediction of security FIS.

After using the Correction FTS model, the system adjusted some of the security levels to improve prediction accuracy. The Correction FTS used a five-point sliding window for utilizing 5 prior predicted data and trend-based preprocessing patterns. Figure 3 presents the adjusted predictions, which are different compared with security levels predicted from first FIS processing. As we can see, the new predictions are now much more aligned with the Predicted Access History trend. This means that the system understands better how a user behaves over time, whether they access the system at certain times, from specific locations, or in particular ways and adjusts the security levels according to these changes.

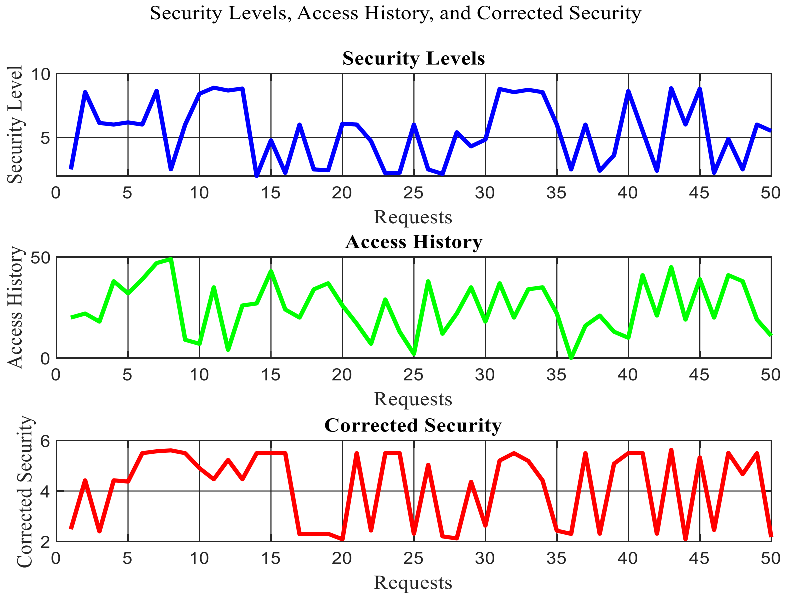


Fig. 3. Corrected Security levels based on Access History and FIS output.

Instead of relying on the initial predictions, the system predicts real patterns from past behavior, making the security levels feel more accurate for each user as shown in Figure 4

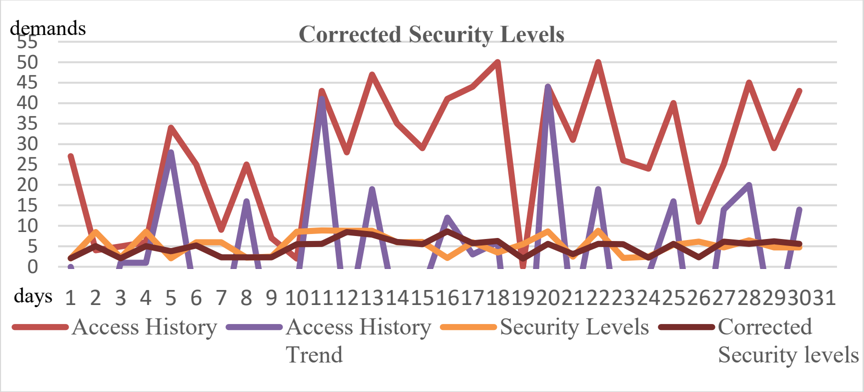


Fig. 4. Dynamic security correction with Trend Analysis of Security levels.

4 Conclusions

The proposed security correction system, using dual FIS and FTS, provides a dynamic solution for adjusting security levels based on evolving user behaviors, access patterns, and historical trends. This adaptive approach does successfully control user access in complex environments like assisted living setting by offering real-time security adjustments such as Figure 5.

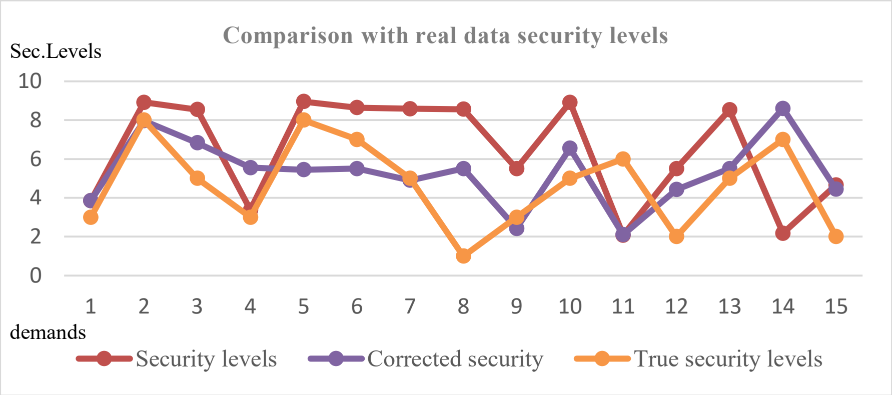


Fig. 5. Corrected Security Levels and True security levels based on FTS.

As we can see, the proposed correction framework significantly outperforms Security FIS across datasets in terms of prediction nearer the true Security Levels. The corrected model performed on real data set (true security levels) showed a 24% reduction in prediction errors PSL and CSL calculated from Equations (1) and (2). The model adjusts security based on real-time data, anticipating risks using historical security trends and access patterns. Furthermore, immediate risk assessment with long term trend analysis and improve decision making with insights into evolving security patterns like comparison of Table 2 with just 10 data inputs.

Table 2. Examples of comparing Corrected Vs True Levels.

User Role	Access Time	Access History	Access Location	Access Prediction	Sec. Levels	Corrected Sec.	True Sec. Level
'Doctor'	**2**	**1**	'Inside Residential Area'	**25**	**4**	**4**	**3**
'Technician'	**6**	**45**	'Outside Residential Area'	**24**	**9**	**8**	**8**
'Nurse'	**19**	**46**	'Outside Residential Area'	**38**	**9**	**7**	**5**
'Doctor'	**0**	**40**	'Inside Residential Area'	**25**	**3**	**6**	**3**
'Technician'	**22**	**5**	'Outside Residential Area'	**41**	**9**	**5**	**8**
'Technician'	**17**	**13**	'Outside Residential Area'	**26**	**9**	**6**	**7**
'Nurse'	**11**	**17**	'Outside Residential Area'	**32**	**9**	**5**	**5**
'Admin'	**13**	**34**	'Outside Residential Area'	**12**	**9**	**6**	**1**
'Doctor'	**5**	**6**	'Outside Residential Area'	**25**	**6**	**2**	**3**
'Nurse'	**11**	**36**	'Outside Residential Area'	**39**	**9**	**7**	**5**

The fuzzy correction model seems to change the security level prediction by reducing the variance, which in turn makes the output more stable and consistent. By incorporating Trends, the model adjusts the predictions more closely with the true security levels, offering a more reliable assessment of security conditions. This correction process helps smooth out fluctuations that might arise from the raw input data. It still underperforms slightly at certain decision points (point 4 & 8), but referring to the instruction given to the system regarding restrictions on access times out of the working hours (As seen in case 4, the access occurs at 0:00 hour, indicates midnight activity) it can be said it does perform well. But this also may suggest that rule weights could be fine-tuned in user desired needs of security.

A general overview of using FTS in correction of simple fuzzy systems is given in pivot analysis referring to Figure 6. It does emphasize the usefulness of using FTS over traditional FIS because of the ability to account for trends, which is crucial for making more accurate predictions over extended periods of time.

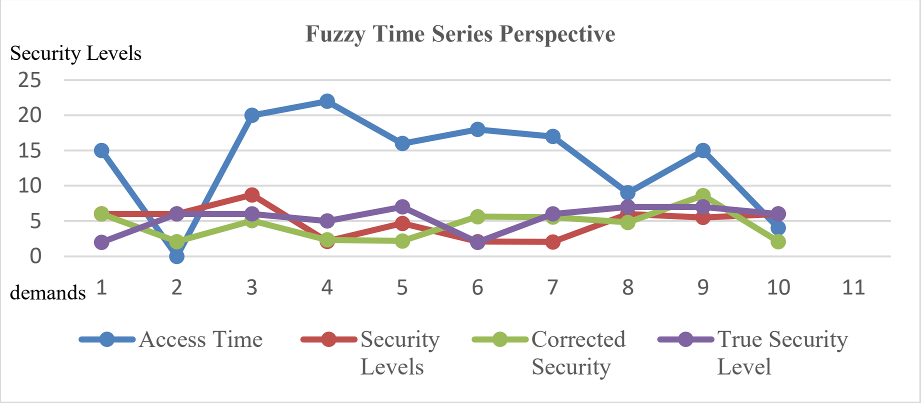


Fig. 6. Evaluation of Fuzzy-Based Correction of Security Level.

We could consider refining membership functions or adding more influence from Access Time spikes to better reflect urgency. Over time, more data can be collected, so the system can evolve. With continuous learning models, the fuzzy inference system can adapt to new patterns and risks, becoming more precise in its predictions.

References

1. Z. Zulkifl et al., “FBASHI: Fuzzy and Blockchain-Based Adaptive Security for Healthcare IoTs,” IEEE Access, vol. 10, pp. 15644–15656, 2022, doi: 10.1109/ACCESS.2022.3149046.

2. S. H. Gökler, D. Yılmaz, Z. F. Ürük, and S. Boran, "A new hybrid risk assessment method based on Fine-Kinney and ANFIS methods for evaluation spatial risks in nursing homes," Heliyon, vol. 8, no. 10, p. e11028, Oct. 2022, doi: 10.1016/J.HELIYON. 2022.E11028.
3. H. Medjahed, D. Istrate, J. Boudy, and B. Dorizzi, "Human activities of daily living recognition using fuzzy logic for elderly home monitoring," in IEEE International Conference on Fuzzy Systems, 2009, pp. 2001–2006. doi: 10.1109/FUZZY.2009.5277257.
4. A. Xhemali, E. Zanaj G. Basha, and L. Balliu, "Enhancing Energy Efficiency Prediction in Assisted Living Through GA-FIS, PSO-FIS, and NGSA-II-FIS: A Comparative Evaluation.," Proceedings of the 5th Winter IFSA Conference on Automation, Robotics& Communications for Industry 4.0/5.0 (ARCI' 2025) IFSA Publishing, S. ISSN: 2938-4796 ISBN: 978-84-09-69171-5. Doi: 10.13140/ RG.2.2.13085.63208,
5. H. F. Atlam, R. J. Walters, G. B. Wills, and J. Daniel, "Fuzzy Logic with Expert Judgment to Implement an Adaptive Risk-Based Access Control Model for IoT," Mobile Networks and Applications, vol. 26, no. 6, pp. 2545–2557, Dec. 2021, doi: 10.1007/s11036-019-01214-w.
6. C. Joshi, R. K. Ranjan, and V. Bharti, "A Fuzzy Logic based feature engineering approach for Botnet detection using ANN," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, pp. 6872–6882, Oct. 2022, doi: 10.1016/j.jksuci.2021.06.018.
7. D. Suleiman, M. Al-Zewairi, and A. Shaout, "Enhanced Multilevel Fuzzy Inference System for Risk Adaptive Hybrid RFID Access Control System," International journal of online and biomedical engineering, vol. 18, no. 4, pp. 31–51, 2022, doi: 10.3991/ijoe.v18i04.27485.
8. Z. Jian, Z. Qun, and T. Jianping, "A Network Security Risk Fuzzy Clustering Assessment Model Based on Weighted Complex Network," 2011.
9. E. K. Szczepaniuk, H. Szczepaniuk, T. Rokicki, and B. Klepacki, "Information security assessment in public administration," Comput Secur, vol. 90, Mar. 2020, doi: 10.1016/j.cose.2019.101709.
10. A. Agrawal, M. Alenezi, S. A. Khan, R. Kumar, and R. A. Khan, "Multi-level Fuzzy system for usable-security assessment," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 3, pp. 657–665, Mar. 2022, doi: 10.1016/j.jksuci.2019.04.007.
11. J. Alam, M. Kumar Pandey, M. K. Pandey, and A. Professor, "Advance Cyber Security System using fuzzy logic," ACME: Journal of Management &IT, vol. 10, no. 17, 2014, [Online]. Available: https://www.researchgate.net/publication/279917296
12. E. C. G. Gabriel, A. O. A. Manuel, and M. Saba, "Fuzzy System for Perception Level Estimation in E-Commerce Web Sites," TEM Journal, vol. 12, no. 4, pp. 1939–1947, Nov. 2023, doi: 10.18421/TEM124-03.
13. P.-C. Cheng, P. Rohatgi, and C. Keser, "Fuzzy MLS: An Experiment on Quantified Risk-Adaptive Access Control," 2007.
14. M. Kchaou, C. Castro, R. Abbassi, V. Leiva, and H. Jerbi, "Security Control for a Fuzzy System under Dynamic Protocols and Cyber-Attacks with Engineering Applications," Mathematics, vol. 12, no. 13, Jul. 2024, doi: 10.3390/math12132112.

Cloud Access Control Parameter Management (https://www.kaggle.com/datasets/brijlaldhankour/cloud-access-control-parameter-management/data

1st International Scientific Conference
on Computer Science and Information Technology - CSIT® 2025

31. More Concise Representation of Regular Languages by Constant Memory Automata

Dominik Leśniewski[1]

[1]University of Lodz, Faculty of Mathematics and Computer Science, Lodz, Poland
dominik.lesniewski@wmii.uni.lodz.pl

Abstract. We study constant memory automata as a formalism for representing regular languages. We utilize the notion of connected components from graph theory to construct more concise constant length queue automata and constant height pushdown automata simulating finite automata. We show that even a single memory cell allows them to outperform traditional finite state automata.

Keywords: Regular Languages, Constant Memory Automata, Queue Automata of Constant Length, Pushdown Automata of Constant Height, Weakly Connected Components, Descriptional Complexity, Graph Reduction.

1 Introduction

Among the well-established computational models in automata theory there are queue automata and pushdown automata, with the latter being more popular. Both models are obtained by equipping finite state automata with auxiliary memory. For queue automata that memory is used according to the first-in, first-out principle, whereas for pushdown automata the memory is used according to the last-in, first-out principle instead. It is known that queue automata are computationally universal, that is, equivalent to Turing machines, while pushdown automata recognize only context-free languages. When we impose restrictions upon these models, their power drops significantly. Constant memory automata (pushdown or queue) were introduced and studied as a formalism for representing regular languages in [1, 2, 3]. Due to fixing a constant bound on the amount of available memory, not depending on the input length, the computational power of such automata is equivalent to that of finite state automata. Because of that, the authors considered them as a potentially more succinct model for representing regular languages and investigated their descriptional power. The notion of descriptional complexity is well described in [4]. The oldest and most famous result in this field is the exponential size cost of simulating nondeterministic finite automata by deterministic finite automata [5]. This cost is proven to be optimal [6, 7] in general case. Descriptional complexity can be measured in two different ways: relative and absolute [8]. In this paper, we focus on the former, that is, we compare the descriptional complexity of different models. Size costs of simulations between constant height pushdown automata and finite automata are well known and optimal. We can simulate constant height nondeterministic (resp. deterministic) pushdown automata by exponentially larger nondeterministic (resp. deterministic) finite automata [3]. It is easy to see that opposite simulations cause linear size increase. Cost of removing nondeterminism in constant height pushdown automata is proven to be double-exponential and optimal [9]. Respective simulation costs between constant length queue automata and finite automata are the same with the exception of removing nondeterminism, which results in an optimal, exponential size blow-up [1, 2]. These results can be found summarized in [2].

Although the size costs of transformations from finite automata to constant memory automata are generally regarded as trivial and turn out to be linear, we show that, in the best-case scenario, it is possible to transform an n-state finite automaton into a constant memory automaton with $\sqrt{n}$ states, $\sqrt{n}$ memory symbols and a single memory cell. It is achieved by new constructions utilizing connected components of the graph of finite automata, with the aim of creating a more succinct constant memory automata accepting the same language as the original finite automata. The underlying concept of those constructions is a kind of a graph reduction considered in [10], where it found application in graph neural networks.

Since our constructions use only a single memory cell, the results are the same for both queue and pushdown automata. Because of that, in the article we describe only the transformation of finite automata into constant length queue automata. We investigate the descriptional complexity of the proposed constructions.

2 Definitions and Preliminaries

We assume that the reader is familiar with automata theory, and we refer to [11] for an extensive introduction to the topic. Descriptional complexity of models has a well-known meaning in the field of theoretical computer science and has been a subject of intensive research (for more details, see [4]).

2.1 Constant Memory Automata

Nondeterministic finite automaton is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q, Σ, q_0 and F are as usual and δ is the next-state function, corresponding to the set of edges in the graph representation of a given automaton. The deterministic version is obtained by imposing the standard restrictions on the function δ.

Nondeterministic queue automaton is a septuple $Z = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$, where Q, Σ, q_0 and F are defined as for nondeterministic finite automata, Γ is the queue alphabet, transition function δ maps $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma$ to finite subsets of $Q \times \{D, K\} \times \Gamma^*$, and $\perp \in \Gamma$ is the initial symbol in the queue. Let $\delta(q, \sigma, \gamma) \ni (p, \chi, \omega)$. Then Z, being in the state q, reading σ from the input and γ as the head of the queue, can reach the state p, delete (resp. keep) γ if $\chi = D$ (resp. $\chi = K$), enqueue ω and finally, if $\sigma \neq \lambda$, advance the input head by one symbol. An input string is accepted if there exists a computation beginning in the initial state q_0 with $\perp$ in the queue and ending in some final state $q \in F$ after reading the entire input. The set of all inputs accepted by Z is denoted by $L(Z)$. The deterministic version is obtained by imposing that for any $q \in Q, \sigma \in (\Sigma \cup \{\lambda\})$ and $\gamma \in \Gamma$, we have $|\delta(q, \sigma, \gamma)| \leq 1$, and if $\delta(q, \lambda, \gamma)$ is defined then $|\delta(q, a, \gamma)| = 0$ for any $a \in \Sigma$.

A *constant length nondeterministic queue automaton* is a nondeterministic queue automaton in which the queue is restricted to contain at most h symbols, for a given constant $h \geq 1$ not depending on the input length. Any attempt to store more than h symbols in the queue results in rejecting the input. Such a machine will be denoted by an octuple $Z = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F, h)$, where $h \geq 1$ is the queue length and all other elements are defined as for nondeterministic finite automata.

Nondeterministic pushdown automaton is a septuple $Z = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$, where Q, Σ, q_0, and F are defined as for nondeterministic finite automata, Γ is the pushdown alphabet, transition function δ maps $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma$ to finite subsets of $Q \times \Gamma^*$, and $\perp \in \Gamma$ is the initial symbol on the pushdown stack. Let $\delta(q, \sigma, \gamma) \ni (p, \omega)$. Then Z, being in the state q, reading σ from the input and the stack symbol γ on the top of the stack, can reach the state p, replace γ by ω, and finally advance input scanning to the next input symbol only if $\sigma \neq \lambda$. An input string is accepted if there exists a computation beginning in the initial state q_0 with $\perp$ on the stack and ending in some final state $q \in F$ after reading the entire input. The set of all inputs accepted by Z is denoted by $L(Z)$. The deterministic version is obtained by imposing that for any $q \in Q, \sigma \in (\Sigma \cup \{\lambda\})$ and $\gamma \in \Gamma$, we have $|\delta(q, \sigma, \gamma)| \leq 1$, and if $\delta(q, \lambda, \gamma)$ is defined then $|\delta(q, a, \gamma)| = 0$ for any $a \in \Sigma$.

A *constant height nondeterministic pushdown automaton* is a nondeterministic pushdown automaton in which the pushdown is restricted to contain at most h symbols, for a given constant $h \geq 1$ not depending on the input length. Any attempt to store more than h symbols in the pushdown results in rejecting the input. Such a machine will be denoted by an octuple $Z = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F, h)$, where $h \geq 1$ is the pushdown height and all other elements are defined as for traditional nondeterministic pushdown automata.

These constant memory models are equivalent in terms of computational power because they recognize the same class of languages - the regular languages [2]. In particular, when the memory limit is equal to 1, that is, $h = 1$, constant length queue automata and constant height pushdown automata operate identically, as the method of memory access no longer affects their performance.

We denote *descriptional complexity* of constant memory automata by a triple $(|Q|, |\Gamma|, \mathrm{h})$, where $|Q|$ is the number of states, $|\Gamma|$ is the size of memory alphabet, and h is the memory limit. In the above, by $|X|$ we denote the size of a set X. To compare descriptional complexity of various models, we also define the *total descriptional complexity* as the sum $|Q| + |\Gamma| + h$.

In the following, a *constant memory automaton* is either a constant length nondeterministic queue automaton or a constant height nondeterministic pushdown automaton.

2.2 Connected Components

We utilize connected components from the graph theory ([12], Ch. 9) in construction of constant length queue automata equivalent to finite automata.

Definition 1 (Weakly Connected Components). *Given a directed graph A of a finite automaton, the weakly connected components are the maximal subgraphs of the graph A in which all vertices are of the same type and are connected to each other by some path, ignoring the direction of edges.*

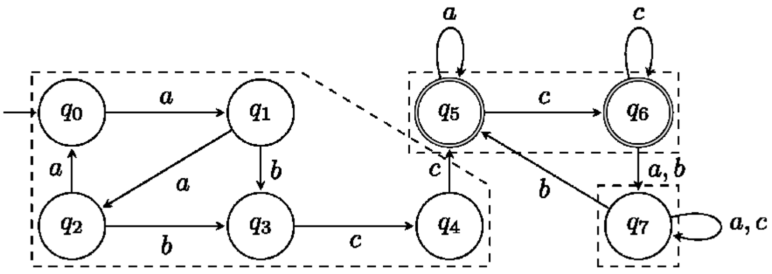


Fig. 15. Weakly connected components of the graph *A* of an automaton.

A simple example of a graph and its weakly connected components is given in Fig. **15**. Note that the symbols on the edges are irrelevant when determining weakly connected components. We focus solely on the existence of edges between vertices. It is easy to observe that the collection of weakly connected components forms a partition of the set of vertices of *A* into disjoint sets.

Definition 2 (Strongly Connected Components). *Given a directed graph A of a finite automaton, the strongly connected components are the maximal subgraphs of the graph A in which for every two vertices v and w, there exists a directed path from v to w and a directed path from w to v.*

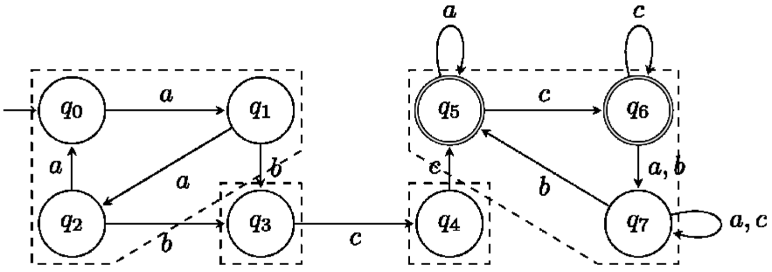


Fig. 16. Strongly connected components of the graph *A* of an automaton.

A simple example of a graph and its strongly connected components is given in Fig. 16. Those components may consist of final and non-final states. As in the case of weakly connected components, symbols on edges are irrelevant when determining strongly connected components and the collection of strongly connected components forms a partition of the set of vertices of A into disjoint sets.

3 Construction by Weakly Connected Components

We will use weakly connected components, consisting of states of the same type, to transform a finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ into an equivalent constant length queue automaton $A' = (Q', \Sigma, \Gamma, \delta', q_0', \perp, F', 1)$ with $h = 1$. The goal of this transformation is to reduce the size of the automaton without losing any information.

The construction starts with finding all weakly connected components of our finite automaton A. Denote these components by $q_0', q_1', \dots, q_k'$ where q_0' is the component containing the initial state of the automaton A. In each component we index the vertices (states) in any way, starting from 1. In A', every weakly connected component from A is represented by a single state, that is, $Q' = \{q_0', \dots, q_k'\}$. We put q_r' is final (resp. non-final) if the weakly connected component consists of final (resp. non-final) states. In such a state of A', the original vertices in a weakly connected component will be distinguished by the queue symbols. We put $\Gamma = \{1, 2, \dots, n\}$, where n is the maximal number of states in any weakly connected component.

Example 1. The automaton A' constructed from A shown in
Fig. **15** would consist of three states: q_0', q_1', q_2', where q_0' is the initial, non-final state, q_1' is final and q_2' is non-final.

In A', we will represent a particular state q_i of A by the pair: state q_r' representing the weakly connected component containing q_i and the queue symbol γ representing the index of q_i in the component q_r'. The transition function δ' reflects the transitions of automaton A. Namely, for each transition $\delta(q_i, \sigma) = q_j$ of the automaton A we define:

1. If this transition from q_i to q_j does not change weakly connected component (q_i and q_j are of the same type) we put

$$\delta'(q_r', \sigma, \gamma) = (q_r', D, \gamma')$$

   where $q_r' \in Q'$ denotes the weakly connected component in automaton A which contains q_i and q_j, $\sigma \in \Sigma$ is an input symbol, $\gamma \in \Gamma$ is the index of q_i in q_r' and $\gamma' \in \Gamma$ is the index of q_j in q_r'. In such case, automaton A' stays at the same state while the queue symbol γ gets replaced by γ'.

2. If this transition from q_i to q_j does change weakly connected component (q_i and q_j are not of the same type) we put

$$\delta'(q'_r, \sigma, \gamma) = (q'_s, D, \gamma')$$

where $q'_r, q'_s \in Q'$ denote weakly connected components in automaton A which contain q_i and q_j respectively, $\sigma \in \Sigma$ is an input symbol, $\gamma \in \Gamma$ is the index of q_i in q'_r and $\gamma' \in \Gamma$ is the index of q_j in q'_s. In such case, automaton A' goes to another state (meaning another weakly connected component) while the queue symbol γ gets replaced by γ'.

In A', the initial state q'_0 denotes the weakly connected component of automaton A that contains the initial state of A. As $\perp$, we take the index of the initial state of A in q'_0. The set of final states F' in A' consists of states representing the weakly connected components of automaton A made of final states. At last, we take $h = 1$, which means that the queue can never be longer than 1 symbol.

Moreover, it is easy to see that this construction preserves determinism.

Example 2. The automaton A' equivalent to A from Fig. **15** is shown in Fig. 17.

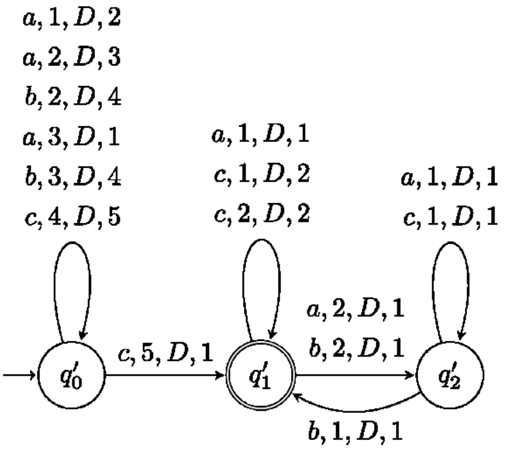


Fig. 17. The automaton A'.

The above considerations lead to the following theorem in which we also examine the descriptional complexity of the constant length queue automata obtained by the proposed construction.

Theorem 1. *Let A be a finite automaton. Then there exists a constant length queue automaton A', equivalent to A (accepting the same language), such that the descriptional complexity of A' is* $(|Q'|, |\Gamma|, 1)$*, where:*

- $|Q'|$ *is the number of weakly connected components of A;*
- $|\Gamma|$ *is the size of the largest weakly connected component of A;*
- $h = 1$ *is the constant memory limit.*

Proof. To prove the equivalence of A and A' we need to show that for each accepting computation in A, there is an equivalent accepting computation in A', and vice versa.

The above construction maps each state q_i of A to a unique pair (q'_r, γ) in A', where q'_r is the weakly connected component containing q_i and γ is the index of q_i in q'_r. Moreover this correspondence is an injection. Similarly, the transition function δ' exactly reflects the transitions of automaton A defined by δ, using the aforementioned mapping. So, the above construction guarantees that every word accepted by A is also accepted by A', and every word accepted by A' is also accepted by A. Therefore, A and A' accept the same languages.

The descriptional complexity given in the theorem above comes from the construction. □

In the above construction we can easily distinguish the best case scenario in which case we get more concise representation of finite automata.

Corollary 1. *Let A be a finite automaton with the set of states* Q*. If the number of weakly connected components in A is equal to their sizes then constant length queue automaton A' equivalent to A has the descriptional complexity* $\left(\sqrt{|Q|},\ \sqrt{|Q|}, 1\right)$*. The total descriptional complexity is* $2\sqrt{|Q|} + 1$*, that is* $O\left(\sqrt{|Q|}\right)$.

Example 3. We show the following example for the above corollary.

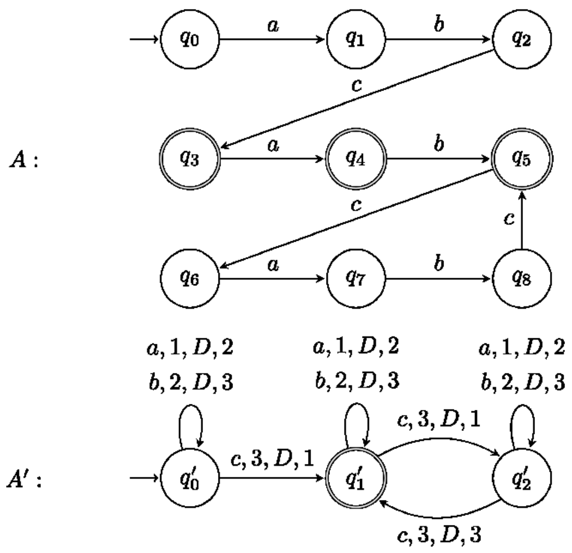


Fig. 18. A best-case example.

On the other hand, we can easily show the total descriptional complexity in the worst case scenario for this construction. It is sufficient to take an automaton A consisting of one weakly connected component containing only non-final states and one weakly connected component consisting of a single final state. In this case, A' has 2 states, $|Q| - 1$ symbols in the queue alphabet and $h = 1$. The total descriptional complexity is $|Q| + 2$, that is, $O(|Q|)$. See *Example 4.*

Example 4. Finite automaton A shown in Fig. 19 consists of four states divided into two weakly connected components. One of them consists of three non-final states and the other one consists of a single final state. The constant length queue automaton A', equivalent to A, has descriptional complexity $(2, 3, 1)$ and its total descriptional complexity is 6.

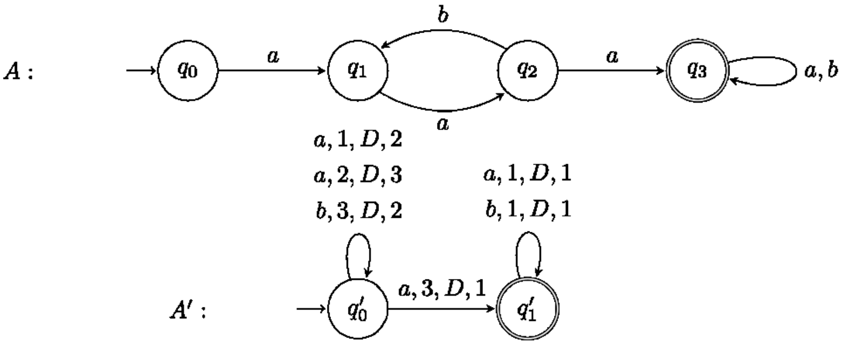


Fig. 19. A worst-case example.

Hence, the case presented above coincides with the linear size cost of the transformation from finite automata to constant length queue automata considered in [1, 2].

Remark 1. The same results hold true if we replace constant length queue automata with $h = 1$ by constant height pushdown automata with $h = 1$, as the memory with $h = 1$ makes them operate in the same way.

4 Construction by Strongly Connected Components

In this section we will use strongly connected components to transform a finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ into an equivalent constant length queue automaton $A' = (Q', \Sigma, \Gamma, \delta', q_0^*, \perp, F', 1)$ with $h = 1$.

The construction starts with finding all strongly connected components of our finite automaton A. Denote these components by $S_0, S_1, \dots, S_k$ where S_0 is the component containing the initial state of the automaton A. Each strongly connected component S_r can contain either only one type of states or both non-final and final states. In the first scenario, S_r will be represented in A' by a single state q_r' if it consists of non-final states (resp. q_r'' if it consists of final states). In such S_r we index its states in any way, starting from 1. In the second scenario, that is, S_r consists of both non-final and final states, we will represent it by two states: q_r' and q_r''. In such S_r we index non-final and final states independently. So, in A', every strongly connected component from A is represented by at most two states, that is, $Q' \subseteq \{q_0', q_0'', \dots, q_k', q_k''\}$. We put q_r' to be non-final and q_r'' to be final in A'. The initial state q_0^* is q_0' or q_0'' depending on the type of q_0 in A. We put $\Gamma = \{1, 2, \dots, n\}$, where n is the maximal number of states of the same type in any strongly connected component. The original states of A in a strongly connected component will be distinguished by the queue symbols from Γ.

Example 5. The automaton A' constructed from A shown in Fig. 16 would consist of five states: $q_0' = \{q_0, q_1, q_2\}, q_1' = \{q_3\}, q_2' = \{q_4\}, q_3' = \{q_7\}, q_3'' = \{q_5, q_6\}$, where q_0' is the initial, non-final state, q_1', q_2', q_3' are non-final and q_3'' is final.

In A', we will represent a particular state q_i of A by the pair: state q_r' (resp. q_r'') representing strongly connected component containing q_i and the queue symbol γ representing the index of q_i in q_r' (resp. q_r''). The transition function δ' reflects the transitions of the automaton A. Namely, for each transition $\delta(q_i, \sigma) = q_j$ of the automaton A we define:

1. If this transition from q_i to q_j does not change strongly connected component and both q_i, q_j are non-final (resp. final), we put one of the following:

$$\delta'(q_r', \sigma, \gamma) = (q_r', D, \gamma')$$

$$\delta'(q_r'', \sigma, \gamma) = (q_r'', D, \gamma')$$

   where $q_r' \in Q'$ (resp. $q_r'' \in Q'$) denotes the non-final (resp. final) state of S_r in automaton A which contain q_i and q_j, $\sigma \in \Sigma$ is an input symbol, $\gamma \in \Gamma$ is the index of q_i in S_r and $\gamma' \in \Gamma$ is the index of q_j in S_r. In such case, automaton A' stays at the same state while the queue symbol γ gets replaced by γ'.

2. If this transition from q_i to q_j does not change strongly connected component but q_i, q_j are of different types, we put one of the following:

$$\delta'(q_r', \sigma, \gamma) = (q_r'', D, \gamma')$$

$$\delta'(q_r'', \sigma, \gamma) = (q_r', D, \gamma')$$

   where $q_r' \in Q'$ (resp. $q_r'' \in Q'$) denotes the non-final (resp. final) states of S_r in automaton A which contain q_i, and $q_r'' \in Q'$ (resp. $q_r' \in Q'$) denotes the final (resp. non-final) states of S_r in automaton A which contain q_j, $\sigma \in \Sigma$ is an input symbol, $\gamma \in \Gamma$ is the index of q_i in S_r and $\gamma' \in \Gamma$ is the index of q_j in S_r. In such case, automaton A' goes to another state (either from q_r' to q_r'' or from q_r'' to q_r') representing the corresponding state type of the same strongly connected component, while the queue symbol γ gets replaced by γ'.

3. If this transition from q_i to q_j changes strongly connected component we put one of the following:

$$\delta'(q_r', \sigma, \gamma) = (q_s', D, \gamma')$$

$$\delta'(q_r'', \sigma, \gamma) = (q_s'', D, \gamma')$$

$$\delta'(q_r', \sigma, \gamma) = (q_s'', D, \gamma')$$

$$\delta'(q''_r, \sigma, \gamma) = (q'_s, D, \gamma')$$

where $q'_r, q'_s \in Q'$ (resp. $q''_r, q''_s \in Q'$) denote the non-final (resp. final) states of strongly connected components in automaton A which contain q_i and q_j respectively, $\sigma \in \Sigma$ is an input symbol, $\gamma \in \Gamma$ is the index of q_i in S_r and $\gamma' \in \Gamma$ is the index of q_j in S_s. In such case, automaton A' goes to another state (meaning another strongly connected component) while the queue symbol γ gets replaced by γ'.

As $\bot$, we take the index of the initial state of A in S_0. The final states F' of A' are those states of A' which consist of final states of A, that is, q''_r. At last, we take $h = 1$, which means that the queue can never be longer than 1 symbol.

Moreover, it is easy to see that this construction preserves determinism.

Example 6. The automaton A' equivalent to A from Fig. 16 is shown in Fig. 20.

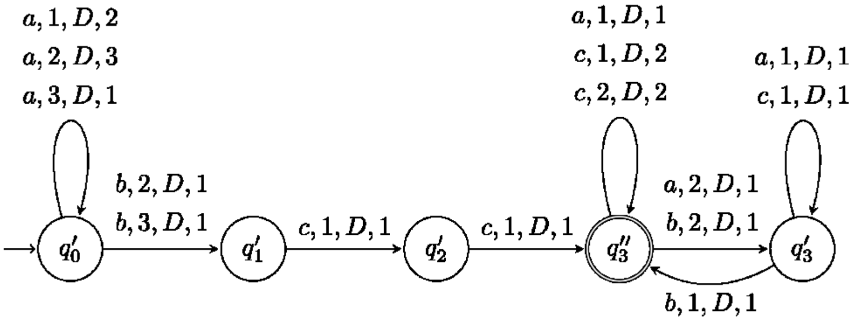


Fig. 20. The automaton A'.

The above considerations lead to the following theorem:

Theorem 2. *Let A be a finite automaton. Then there exists a constant length queue automaton A', equivalent to A (accepting the same language), such that the descriptional complexity of A' is $(|Q'|, |\Gamma|, 1)$, where:*

- $|Q'|$ *is the number of states representing strongly connected components of A;*
- $|\Gamma|$ *is the maximal number of states of the same type in any strongly connected component of A;*
- $h = 1$ *is the constant memory limit.*

Proof. Analogous to the proof of Theorem 1. □

In the above construction we can easily distinguish the best case scenario in which case we get more concise representation of finite automata.

Corollary 2. *Let A be a finite automaton with the set of states Q. If the number of strongly connected components in A is equal to their sizes and each strongly connected component consists of only one type of states, then constant length queue automaton A' equivalent to A has the descriptional complexity* $\left(\sqrt{|Q|}, \sqrt{|Q|}, 1\right)$ *. The total descriptional complexity is* $2\sqrt{|Q|} + 1$*, that is* $O\left(\sqrt{|Q|}\right)$.

In the worst case scenario, for instance for an automaton A whose graph is a tree, the total descriptional complexity is $|Q| + 2$, that is, $O(|Q|)$. It is easy to show that $|Q| + 2$ is the maximal descriptional complexity of constant length queue automata A' equivalent to A.

Remark 2. As in Section 3, the same results hold true if we replace constant length queue automata with $\mathrm{h} = 1$ *by constant height pushdown automata with* $\mathrm{h} = 1$.

5 Conclusions

In the article we study constant memory automata equivalent to finite automata. We utilize the notion of connected components in the graphs of finite automata to achieve more concise representation of regular languages by constant memory automata. Both presented constructions have the same total descriptional complexity $2\sqrt{|Q|} + 1$ in the best case and $|Q| + 2$ in the worst case scenarios, where Q is the set of states of a finite automaton.

In the best case, the results come from the fact that we map each state of a finite automaton to a unique pair (*state*, *queue symbol*) in a constant memory automaton. Because of this, we can say that we are trying to find two numbers $|Q'| \in \mathbb{N}$ and $|\Gamma| \in \mathbb{N}$ such that:

- $|Q'| * |\Gamma| \geq |Q|$, and
- $|Q'| + |\Gamma|$ is as small as possible.

From mathematical considerations we get that the above optimization problem is realized by real numbers $|Q'| = \sqrt{|Q|}$ and $|\Gamma| = \sqrt{|Q|}$. So, we should take integers $|Q'|$ and $|\Gamma|$ as close as possible to $\sqrt{|Q|}$. Our constructions precisely implement this complexity in the best case scenarios.
In the worst case, the results match the well-known linear size cost of the transformation from finite automata to constant length queue automata considered in [1, 2].

References

1. Jakobi, S., Meckel, K., Mereghetti, C., Palano, B.: Queue Automata of Constant Length. In: Jurgensen, H., Reis, R. (eds.) Descriptional Complexity of Formal Systems. DCFS 2013. Lecture Notes in Computer Science, vol. 8031, pp. 124–135. Springer, Heidelberg (2013).
2. Jakobi, S., Meckel, K., Mereghetti, C., Palano, B.: The Descriptional Power of Queue Automata of Constant Length. Acta Informatica, vol. 58, pp. 335–356 (2021).
3. Geffert, V., Mereghetti, C., Palano, B.: More Concise Representation of Regular Languages by Automata and Regular Expressions. In: Ito, M., Toyama, M. (eds.) Developments in Language Theory. DLT 2008. Lecture Notes in Computer Science, vol. 5257, pp. 359–370. Springer, Heidelberg (2008).
4. Holzer, M., Kutrib, M.: Descriptional Complexity - An Introductory Survey. Scientific Applications of Language Methods, vol. 2, pp. 1–58. Imperial College Press (2010).
5. Rabin, M. O., Scott, D.: Finite Automata and Their Decision Problems. IBM Journal of Research and Development, vol. 3(2), pp. 114-125 (1959).
6. Meyer, A. R., Fischer, M. J.: Economy of Description by Automata, Grammars, and Formal Systems. In: 12th Annual Symposium on Switching and Automata Theory, pp. 188-191. East Lansing, MI, USA (1971).
7. Moore, F. R.: On the Bounds for State-Set Size in the Proofs of Equivalence Between Deterministic, Nondeterministic, and Two-Way Finite Automata. IEEE Transactions on Computers, vol. C-20(10), pp. 1211-1214 (1971).
8. Schmidt, E. M.: Succinctness of Descriptions of Context-Free, Regular, and Finite Languages. DAIMI Report Series, vol. 7(84) (1978).
9. Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: Removing Nondeterminism in Constant Height Pushdown Automata. In: Kutrib, M., Moreira, N., Reis, R. (eds.) Descriptional Complexity of Formal Systems. DCFS 2012. Lecture Notes in Computer Science, vol. 7386, pp. 76–88. Springer, Heidelberg (2012).
10. Hashemi, M., Gong, S., Ni, J., Fan, W., Prakash, B. A., Jin, W.: A Comprehensive Survey on Graph Reduction: Sparsification, Coarsening, and Condensation. ArXiv:2402.03358v4 (2024).
11. Hopcroft, J. E., Motwani, R., Ullman, J. D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley (2001).
12. Deo, N.: Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall (1974).

CIP Katalogimi në botim BK Tiranë

Universiteti Aleksandër Moisiu
Advances in computer science and information technology :
international conference on computer science and
information technology CSIT® 2025 : book of proceedings /
University "Aleksandër Moisiu" Durrës, South East European
University. - Durrës : Luis Print, 2025.
... f. ; ... cm.
ISBN 9789928835147
1.Shkenca e kompjuterit 2.Teknologjia e
informacionit 3.Konferenca

004 (062)

ISBN 978-9-92-883514-7

9 789928 835147

LUIS PRINT

SHTËPI BOTUESE | PRINTIME